Title
Mean-Field Cooperative Multi-agent Reinforcement Learning: Modelling, Theory, and Algorithms

Author
Gu, Haotian

Publication Date
2023

Peer reviewed|Thesis/dissertation

Mean-Field Cooperative Multi-agent Reinforcement Learning: Modeling, Theory, and Algorithms

By

Haotian Gu

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Mathematics

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Xin Guo, Co-chair
Professor Fraydoun Rezakhanlou, Co-chair
Professor Daniel Tataru

Spring 2023

Mean-Field Cooperative Multi-agent Reinforcement Learning:
Modeling, Theory, and Algorithms

Abstract

Mean-Field Cooperative Multi-agent Reinforcement Learning:
Modeling, Theory, and Algorithms

by

Haotian Gu

Doctor of Philosophy in Mathematics

University of California, Berkeley

Professor Xin Guo, Co-chair

Professor Fraydoun Rezakhanlou, Co-chair

In numerous stochastic systems involving a large number of agents, the model parameters and dynamics are typically not known beforehand. As a result, learning algorithms are crucial for these agents to enhance their decision-making abilities while engaging with the partially unknown system and interacting with other agents. In this case, multi-agent reinforcement learning (MARL) has enjoyed substantial successes for analyzing the otherwise challenging games arising from numerous fields including autonomous driving, supply chain, manufacturing, e-commerce and finance. Despite its empirical success, MARL suffers from the curse of dimensionality: its sample complexity by existing algorithms for stochastic dynamics grows exponentially with respect to the total number of agents N in the system. This PhD thesis focuses on advancing the theoretical understandings and developing novel efficient algorithms with provable performance guarantees to solve large-population cooperative games using MARL and mean-field approximation.

The mean-field approximation of cooperative games in the regime with a large number of homogeneous agents is also known as mean-field control (MFC). It is therefore natural meanwhile important to consider the learning problem in MFCs. The first part of this dissertation focuses on investigating the learning framework of MFCs and establishing the corresponding dynamic programming principle (DPP). Dynamic programming principle is fundamental for control and optimization, including Markov decision problems (MDPs) and reinforcement learning (RL). However, in the learning framework of MFCs, DPP has not been rigorously established, despite its critical importance for algorithm designs. We first present a simple example in MFCs with learning where DPP fails with a mis-specified Q-function; and then propose the correct form of Q-function in an appropriate space for MFCs with learning. This particular form of Q-function is different from the classical one and is called

the IQ-function. Compared to the classical Q-function in the single-agent RL literature, MFCs with learning can be viewed as lifting the classical RLs by replacing the state-action space with its probability distribution space. This identification of the IQ-function enables us to establish precisely the DPP in the learning framework of MFCs. The time consistency of this IQ-function is further illustrated through numerical experiments.

The second part of this dissertation focuses on addressing the curse of dimensionality in MARL with MFC approximations, and developing sample efficient learning algorithms. The mathematical framework to approximate cooperative MARL by MFC is rigorously established, with the approximation error of $\mathcal{O}(\frac{1}{\sqrt{N}})$. Furthermore, based on the DPP for both the value function and the Q-function of learning MFC, it introduces a model-free kernel-based Q-learning algorithm (MFC-K-Q) with a linear convergence rate, which is the first of its kind in MARL literature. Empirical studies confirm the effectiveness of MFC-K-Q, particularly for large-scale problems.

The other approach to reduce the sample complexity for cooperative MARL and learning MFC is to design efficient decentralized learning algorithms, in which each individual agent only requires local information of the entire system. In particular, little is known theoretically for decentralized MARL with network of states. The third study proposes a framework of localized training and decentralized execution for cooperative MARL with network of states and mean-field approximation, to study MARL systems such as self-driving vehicles, ride-sharing, and data and traffic routing. Localized training is to collect local information in agents' neighboring states for training; decentralized execution means to execute the learned decentralized policies that depend only on agents' current states. The theoretical analysis consists of three key components: the first is to establish the mean-field reformulation of the original MARL system as a networked MDP with teams of agents, enabling updating locally the associated team Q-function; the second is to develop the DPP for the mean-field type of Q-function for each team on the probability measure space; and the third is to analyze the exponential decay property of the Q-function, facilitating its approximation with sample efficiency and with controllable error. The analysis leads to a neural-network-based algorithm LTDE-Neural-AC, where the actor-critic approach is coupled with over-parameterized neural networks. Convergence and sample complexity of the algorithm are established and shown to be scalable with respect to the size of agents and states.

To my parents for their unconditional love and support.

Contents

List of Figures

List of Tables

Acknowledgments

I am grateful to numerous individuals who have supported me throughout my PhD journey.

First and foremost, I would like to express my deepest appreciation to my supervisor, Professor Xin Guo, for her unwavering support, guidance, and expertise. Since joining her research group at the end of my first year, she has been instrumental in helping me navigate the challenges of graduate school. Xin has consistently been inspiring, encouraging, enthusiastic, open-minded, and curious. Every discussion with Xin has resulted in insightful comments, novel ideas, precise feedback, valuable advice, and constructive criticism. Her feedback has enabled me to refine and improve my research, and her dedication to my success has been a source of inspiration. I was constantly inspired by her to explore further, not only in our research projects but also in all the fascinating topics this beautiful world has to offer.

I would also like to thank the members of my dissertation and qualification exam committees. Special thanks go to Professor Fraydoun Rezakhanlou for serving as my co-chair in the math department and for providing tremendous support during the past five years. I am also grateful to Professor Daniel Tataru for his role in my dissertation committee and for testing me on PDEs during my qualification exam. MATH 222A, taught by Professor Daniel Tataru in Fall 2018, was the first graduate course I took at Berkeley and marked the beginning of my journey.

In addition to my supervisor and committee members, I am grateful to other faculty members in Berkeley's math, IEOR, statistics, and EECS departments for offering a variety of excellent courses that sparked my research interests and provided useful tools. I would also like to extend my gratitude to graduate student advisors, Vicky and Jon, for their daily support and ongoing assistance.

I wish to thank the faculty members who supported me during my undergraduate studies and PhD application at the University of Hong Kong, UCLA, and Cornell University: Professors Ngaiming Mok, Zhiwen Zhang, Waiki Ching, Wenan Zang, Robert Strichartz, and Wilfrid Gangbo. My undergraduate studies at HKU, along with my exchange experiences at UCLA and Cornell, introduced me to the beauty of mathematics and cultivated my initial research interests in applied mathematics.

I am also grateful for the internship opportunities at Amazon in the summer of 2021 and Citadel Securities in the summer of 2022. I would like to thank my manager Mauricio Resendo at Amazon Middle Mile Research for his kindness and assistance, and my manager Yanfeng Chen at Citadel Securities for introducing me to the exciting world of quantitative trading.

My heartfelt appreciation goes to my parents, whose love, encouragement, and unwavering support have been the foundation of every success in my life. Their faith in me has sustained me through the stress and uncertainty of graduate school and has motivated me to persevere. I would also like to sincerely thank my girlfriend, Can, in front of whom I can always be my true self.

My PhD journey would have been far less enjoyable without the company of friends and colleagues. I am grateful to my brilliant academic peers: Renyuan, Xiaoli, Haoyang, Anran, Junzi, Yusuke, Mahan, Jiacheng, and Xinyu for their guidance and unconditional support. I also wish to thank the talented friends I made during my PhD journey: Jiahao, Yiling, Yunbo, Fan, Tianyu, Anlu, Wenjun, Lin, Yixuan, Xiaohan, Jiaming, Zirui, Jiasu, Hongyi, Yuhao, Tianyi, Sizhu, Mogen, Qihang, Brian, Sijie and Yichen. We shared laughter, songs, meals, and games together to create unforgettable memories. I also want to express my gratitude to my dear old friends: Zhaozhen, Danqing, Zichong, Xiangying, Yuming, Xiaojun, Hengjia, Yuqi, Hongpeng, Fangjun, Yang, and Xiaotian, who have accompanied me through my youth and provided many years of friendship.

Thank you to all the individuals who have accompanied and supported me during this journey in ways that cannot be quantified. I hope that our paths will cross again in the future.

Chapter 1

Introduction

1.1 Single-agent Reinforcement Learning

Reinforcement learning (RL) is about agents interacting with the environment, learning an optimal policy, by trial and error, for sequential decision making problems in a wide range of fields in both natural and social sciences, and engineering [22, 167, 140, 20, 163].

Recent years have witnessed sensational advances of reinforcement learning in many prominent sequential decision-making problems, such as playing the game of Go [155, 157], playing real-time strategy games [176], robotic control [90, 103], playing card games [25, 26], and autonomous driving [154, 150], especially accompanied with the development of deep neural networks (DNNs) for function approximation [126].

In this section, the necessary background on reinforcement learning in the single-agent setting will be provided.

1.1.1 Markov Decision Process

A reinforcement learning agent is modeled to perform sequential decision-making by interacting with the environment. The environment is usually formulated as an infinite horizon discounted Markov decision process (MDP), henceforth referred to as Markov decision process, which is formally defined as follows.

Definition 1.1.1 *A Markov decision process is defined by a tuple* $(\mathcal{S}, \mathcal{A}, P, r, \gamma)$*, where* $\mathcal{S}$ *and* $\mathcal{A}$ *denote the state and action spaces, respectively;* $P : \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathcal{S})$ *denotes the transition probability from any state* $s \in \mathcal{S}$ *to any state* $s' \in \mathcal{S}$ *for any given action* $a \in \mathcal{A}$*;* $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ *is the reward function that determines the immediate reward received by the agent for a transition from* (s, a)*;* $\gamma \in [0, 1)$ *is the discount factor that trades off the instantaneous and future rewards.*

As a standard model, MDP has been widely adopted to characterize the decision making of an agent with full observability of the system state $s \in \mathcal{S}$. At each time t, the agent chooses

to execute an action a_t in face of the system state s_t, which causes the system to transition to $s_{t+1} \sim P(s_t, a_t)$. Moreover, the agent receives an instantaneous reward $r(s_t, a_t)$. The goal of solving the MDP is thus to find a policy $\pi : \mathcal{S} \to \mathcal{P}(\mathcal{A})$, a mapping from the state space $\mathcal{S}$ to the distribution over the action space $\mathcal{A}$, so that $a_t \sim \pi(s_t)$ and the discounted accumulated reward

$$\mathbb{E}\left[\sum_{t\geq 0} \gamma^t r(s_t, a_t) \,\middle|\, s_0, a_t \sim \pi(s_t), s_{t+1} \sim P(s_t, a_t)\right]$$

is maximized. Here the policy π can be either deterministic such that $\pi_t : \mathcal{S} \to \mathcal{A}$, or randomized such that $\pi_t : \mathcal{S} \to \mathcal{P}(\mathcal{A})$.

Note that there are several other standard formulations of MDPs, e.g., time-average-reward setting [123, 186, 178] and finite-horizon episodic setting [43, 44, 132]. Here, we only present the classical infinite-horizon discounted setting for ease of exposition. In this infinite-time horizon, we assume the reward r and the transition dynamics P are time homogeneous, which is a standard assumption in the MDP literature. Meanwhile, there is another important model class called partially observed MDP (POMDP), which is usually advocated when the agent has no access to the exact system state but only an observation of the state. See [127, 108] for more details on the POMDP model.

One can define the action-value function (Q-function) and state-value function (V-function) under policy π as

$$V^\pi(s) := \mathbb{E}\left[\sum_{t\geq 0} \gamma^t r(s_t, a_t) \,\middle|\, s_0 = s, a_t \sim \pi(s_t), s_{t+1} \sim P(s_t, a_t)\right],$$

$$Q^\pi(s, a) := \mathbb{E}\left[\sum_{t\geq 0} \gamma^t r(s_t, a_t) \,\middle|\, s_0 = s, a_0 = a, a_t \sim \pi(s_t), s_{t+1} \sim P(s_t, a_t)\right].$$

Here, the Q-function, one of the basic quantities used for RL, is defined to be the expected reward from taking action a at state s and then following the policy π thereafter.

Meanwhile, the optimal value function and optimal Q-function is defined as

$$V^*(s) = \sup_\pi V^\pi(s),$$

$$Q^*(s, a) = \sup_\pi Q^\pi(s, a).$$

The well-known dynamic programming principle (DPP) [16, 21, 57] that the optimal policy can be obtained by maximizing the reward from one step and then proceeding optimally from the new state, can be used to derive the following Bellman equation for the value function.

$$V^*(s) = \sup_{a\in\mathcal{A}} \left\{\mathbb{E}[r(s, a)] + \gamma \mathbb{E}_{s'\sim P(s,a)}\left[V^*(s')\right]\right\}.$$

In addition, the optimal value function and the optimal Q-function are shown to satisfy the following condition:

$$Q^*(s,a) = \mathbb{E}[r(s,a)] + \gamma \mathbb{E}_{s' \sim P(s,a)} \left[V^*(s') \right],$$
$$V^*(s) = \sup_{a \in \mathcal{A}} Q^*(s,a).$$

There is also a Bellman equation for the optimal Q-function derived from the above relations and given by

$$Q^*(s,a) = \mathbb{E}[r(s,a)] + \gamma \mathbb{E}_{s' \sim P(s,a)} \sup_{a' \in \mathcal{A}} Q^*(s',a').$$

One can also retrieve the optimal (deterministic) policy $\pi^*(s,a)$ (if it exists) from $Q^*(s,a)$ once it is learned, in that $\pi^*(s,a) \in \arg\max_{a \in \mathcal{A}} Q(s,a)$.

The optimal value function and the optimal policy can be obtained by dynamic programming approaches, e.g., value iteration and policy iteration algorithms [19], which require the knowledge of the model, i.e., the transition probability P and the form of reward function r. Reinforcement learning, on the other hand, is to find such an optimal policy without knowing the model. The RL agent learns the policy from experiences collected by interacting with the environment. By and large, RL algorithms can be categorized into two mainstream types, value-based and policy-based methods.

1.1.2 Value-Based Methods

Value-based RL methods are devised to find a good estimate of the state-action value function, namely, the optimal Q-function Q^*. The (approximate) optimal policy can then be extracted by taking the greedy action of the Q-function estimate. One of the most popular value-based algorithms is Q-learning [184], where the agent maintains an estimate of the Q-function $\widehat{Q}(s,a)$. When transitioning from state-action pair (s,a) to next state s', the agent receives a payoff r and updates the Q-function according to:

$$\widehat{Q}(s,a) \leftarrow (1-\alpha) \underbrace{\widehat{Q}(s,a)}_{\text{current estimate}} + \alpha \underbrace{\left[r + \gamma \max_{a'} \widehat{Q}(s',a') \right]}_{\text{new estimate}},$$

where $\alpha > 0$ is the step-size or learning rate. Under certain conditions on α, Q-learning can be proved to converge to the optimal Q-value function almost surely [184, 168], with finite state and action spaces. Moreover, when combined with neural networks for function approximation, deep Q-learning has achieved great empirical breakthroughs in human-level control applications [126].

Another popular value-based method is SARSA (State-Action-Reward-State-Action). In contrast to the Q-learning algorithm, which takes samples from any policy π as the input where these samples could be collected in advance before performing the Q-learning algorithm, SARSA adopts a policy which is based on the agent's current estimate of the Q-function. The different source of samples is indeed the key difference between *on-policy*

and *off-policy* learning. More specifically, an off-policy agent learns the value of the optimal policy independently of the agent's actions. For example, Q-learning is an off-policy agent as the samples (s, a, r, s') used in updating the Q-function may be collected from any policy and may be independent of the agent's current Q-function estimate. In contrast, an on-policy agent, such as SARSA select its next action based on its own estimation of the Q-function, and receive a real-time sample in each iteration. The convergence of SARSA convergence was established in [158] for finite-space settings.

An alternative while popular value-based method is Monte-Carlo tree search (MCTS) [39, 91, 42], which estimates the optimal value function by constructing a search tree via Monte-Carlo simulations. Tree polices that judiciously select actions to balance exploration-exploitation are used to build and update the search tree. The most common tree policy is to apply the UCB1 (UCB stands for upper confidence bound) algorithm, which was originally devised for stochastic multi-arm bandit problems [3, 10], to each node of the tree. This yields the popular UCT algorithm [91]. Recent research endeavors on the non-asymptotic convergence of MCTS include [85, 117].

Besides, another significant task regarding value functions in RL is to estimate the value function associated with a given policy (not only the optimal one). This task, usually referred to as policy evaluation, has been tackled by algorithms that follow a similar update as Q-learning, named temporal difference (TD) learning [171, 172, 163]:

$$\widehat{V}^{\pi}(s) \leftarrow (1-\alpha) \underbrace{\widehat{V}^{\pi}(s)}_{\text{current estimate}} + \alpha \underbrace{\left[r + \gamma \widehat{V}^{\pi}(s')\right]}_{\text{new estimate}}.$$

Some other common policy evaluation algorithms with convergence guarantees include gradient TD methods with linear [164, 165, 109], and nonlinear function approximations [115]. See [45] for a more detailed review on policy evaluation.

1.1.3 Policy-Based Methods

Another type of RL algorithms directly searches over the policy space, which is usually estimated by parameterized function approximators such as neural networks, namely, approximating $\pi(s) \approx \pi_\theta(s)$. As a consequence, the most straightforward idea, which is to update the parameter along the gradient direction of the long-term reward, has been instantiated by the policy gradient (PG) method. As a key premise for the idea, the closed-form of PG is given by [166]:

$$\nabla J(\theta) = \mathbb{E}_{a\sim\pi_\theta(s), s\sim\eta_{\pi_\theta}} \left[Q^{\pi_\theta}(s,a) \nabla_\theta \log \pi_\theta(s)(a)\right],$$

where $J(\theta)$ and Q^{π_θ} are the expected return and Q-function under policy π_θ, respectively, $\nabla_\theta \log \pi_\theta(s)(a)$ is the score function of the policy, and η_{π_θ} is the state occupancy measure, either discounted or ergodic, under policy π_θ. Then, various policy gradient methods, including REINFORCE [188], G(PO)MDP [15], and actor-critic algorithms [92, 24, 125], have

been proposed by estimating the gradient in different ways. A similar idea also applies to deterministic policies in continuous-action settings, whose PG form has been derived by [156]. Besides gradient-based ones, several other policy optimization methods have achieved state-of-the-art performance in many applications, including TRPO [151], PPO [152], soft actor-critic [73].

Compared with value-based RL methods, policy-based approaches enjoy better convergence guarantees [92, 198, 202, 1], especially with neural networks for function approximation [110, 181], which can readily handle massive or even continuous state-action spaces.

1.2 Multi-agent Reinforcement Learning

In a similar vein, multi-agent reinforcement learning (MARL) also addresses sequential decision-making problems, but with more than one agent involved. In particular, both the evolution of the system state and the reward received by each agent are influenced by the joint actions of all agents. More intriguingly, each agent has its own long-term reward to optimize, which now becomes a function of the policies of all other agents. Such a general model finds broad applications in practice, including two-agent or two-team computer games [155, 176], self-driving vehicles [154], real-time bidding games [87], ride-sharing [100], and traffic routing [54].

The environment of MARL is usually formulated as an infinite horizon discounted Markov Game (MG), henceforth referred to as Markov game, which is formally defined as follows.

Definition 1.2.1 *A Markov game is defined by a tuple*

$$\left(N, \{\mathcal{S}^i\}_{i=1}^N, \{\mathcal{A}^i\}_{i=1}^N, \{P^i\}_{i=1}^N, \{r^i\}_{i=1}^N, \gamma\right).$$

Here N denotes the number of all agents in the system; $\mathcal{S}^i$ and $\mathcal{A}^i$ denote the state and action spaces of agent i, respectively; $\mathcal{S} := \mathcal{S}^1 \times \cdots \times \mathcal{S}^N$ and $\mathcal{A} := \mathcal{A}^1 \times \cdots \times \mathcal{A}^N$ denote the joint state and action spaces of all the agents, respectively; $P^i : \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathcal{S}^i)$ denotes the transition probability of agent i from any joint state $\boldsymbol{s} \in \mathcal{S}$ to any state $\boldsymbol{s}' \in \mathcal{S}$ for any given joint action $\boldsymbol{a} \in \mathcal{A}$; $r^i : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function that determines the immediate reward received by agent i for a transition from $(\boldsymbol{s}, \boldsymbol{a})$; $\gamma \in [0, 1)$ is the discount factor that trades off the instantaneous and future rewards.

At each step $t = 0, 1, \cdots$, the state of agent i $(= 1, 2, \cdots, N)$ is $s_t^i \in \mathcal{S}^i$ and it takes an action $a_t^i \in \mathcal{A}^i$. Given the current joint state profile $\boldsymbol{s}_t = (s_t^1, \cdots, s_t^N) \in \mathcal{S}$ and the current action profile $\boldsymbol{a}_t = (a_t^1, \cdots, a_t^N) \in \mathcal{A}$ of N agents, agent i will receive a reward $r^i(\boldsymbol{s}_t, \boldsymbol{a}_t)$ and its state will change to s_{t+1}^i according to a transition probability function $P^i(\boldsymbol{s}_t, \boldsymbol{a}_t)$. A Markovian game further restricts the admissible policy for agent i to be of the form $a_t^i \sim \pi^i(\boldsymbol{s}_t)$. That is, $\pi^i : \mathcal{S} \to \mathcal{P}(\mathcal{A})$ maps each state profile $\boldsymbol{s} \in \mathcal{S}$ to a randomized action, with $\mathcal{P}(\mathcal{A}^i)$ the space of all probability measures on space $\mathcal{A}^i$. In particular, for any joint

policy $\boldsymbol{\pi} = \{\pi^i\}_{i=1}^N$ and joint state $\boldsymbol{s} \in \mathcal{S}$, the value function of agent i given by

$$V^i(\boldsymbol{s}, \boldsymbol{\pi}) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r^i(\boldsymbol{s}_t, \boldsymbol{a}_t) \middle| \boldsymbol{s}_0 = \boldsymbol{s}, a_t^j \sim \pi^j(s_t^j), s_{t+1}^j \sim P^j(\boldsymbol{s}_t, \boldsymbol{a}_t), j = 1, \cdots, N\right]$$

is the discounted accumulated reward for agent i, given the initial state profile $\boldsymbol{s}_0 = \boldsymbol{s}$ and policy $\boldsymbol{\pi} = (\pi^1, \ldots, \pi^N)$.

Since the optimal performance of each agent is controlled not only by its own policy, but also the choices of all other players of the game, the solution concept of an MG deviates from that of an MDP. Two most commonly used solution concepts are the Nash equilibrium (NE) for competitive MARL and the Pareto optimality (PO) for cooperative MARL.

1.2.1 Competitive MARL

Under the competitive setting, a Nash equilibrium (NE) is defined as follows [14, 56].

Definition 1.2.2 *A Nash equilibrium of the Markov game*

$$\left(N, \{\mathcal{S}^i\}_{i=1}^N, \{\mathcal{A}^i\}_{i=1}^N, \{P^i\}_{i=1}^N, \{r^i\}_{i=1}^N, \gamma\right)$$

is a joint policy $\boldsymbol{\pi}^* = \left(\pi^{1,*}, \cdots, \pi^{N,*}\right)$*, such that for any* $\boldsymbol{s} \in \mathcal{S}$ *and* $i = 1, \cdots, N$*,*

$$V^i(\boldsymbol{s}, \pi^{i,*}, \pi^{-i,*}) \geq V^i(\boldsymbol{s}, \pi^i, \pi^{-i,*}), \text{ for any } \pi^i,$$

where $-i$ *represents the indices of all agents except agent* i*.*

Nash equilibrium characterizes an equilibrium point $\boldsymbol{\pi}^*$, from which none of the agents has any incentive to deviate. In other words, for any agent $i = 1, \cdots, N$, the policy $\pi^{i,*}$ is the best response of $\pi^{-i,*}$. As a standard learning goal for MARL, NE always exists for finite-space infinite-horizon discounted MGs [56], but may not be unique in general. Most of the MARL algorithms are contrived to converge to such an equilibrium point, if it exists [107, 77, 114, 11].

Many of the existing works in competitive MARL focus on the fully competitive setting, where the Markov game is modeled as a zero-sum Markov games, namely, $\sum_{i \in \mathcal{N}} r^i(\boldsymbol{s}, \boldsymbol{a}) = 0$ for any $(\boldsymbol{s}, \boldsymbol{a})$. For ease of algorithm analysis and computational tractability, most literature focused on two agents that compete against each other [106, 210]. In addition to direct applications to game-playing [106, 155, 176], zero-sum games also serve as a model for robust learning, since the uncertainty that impedes the learning process of the agent can be accounted for as a fictitious opponent in the game that is always against the agent [13, 201]. Therefore, the Nash equilibrium yields a robust policy that optimizes the worst-case long-term reward.

1.2.2 Cooperative MARL

In the cooperative MARL setting [136], N agents are coordinated by a central controller to maximize the expected discounted accumulated reward averaged over all agents. That is to find

$$V^*(\boldsymbol{s}) = \sup_{\boldsymbol{\pi}} \frac{1}{N} \sum_{i=1}^{N} V^i(\boldsymbol{s}, \boldsymbol{\pi}).$$

The setting is closely related to the concept of Pareto optimality (PO) in the cooperative game theory [137], which is formally defined as the following.

Definition 1.2.3 *A Pareto optimality of the Markov game*

$$\left(N, \{\mathcal{S}^i\}_{i=1}^N, \{\mathcal{A}^i\}_{i=1}^N, \{P^i\}_{i=1}^N, \{r^i\}_{i=1}^N, \gamma\right)$$

is a joint policy $\boldsymbol{\pi}^* = \left(\pi^{1,*}, \cdots, \pi^{N,*}\right)$*, if and only if there does not exist another joint policy* $\boldsymbol{\pi}$*, such that for all* $\boldsymbol{s} \in \mathcal{S}$*,*

$$\forall i \in \{1, \cdots, N\}, V^i(\boldsymbol{s}, \boldsymbol{\pi}) \geq V^i(\boldsymbol{s}, \boldsymbol{\pi}^*); \text{ and } \exists j \in \{1, \cdots, N\}, V^j(\boldsymbol{s}, \boldsymbol{\pi}) > V^j(\boldsymbol{s}, \boldsymbol{\pi}^*).$$

It can be easily verify that the optimal policy maximizing the expected discounted accumulated reward averaged over all agent is a Pareto optimal policy.

Meanwhile, the optimal Q-function is defined as

$$Q^*(\boldsymbol{s}, \boldsymbol{a}) = \mathbb{E}\left[\frac{1}{N} \sum_{i=1}^{N} r^i(\boldsymbol{s}, \boldsymbol{a})\right] + \gamma \mathbb{E}_{\boldsymbol{s}' \sim \boldsymbol{P}(\boldsymbol{s}, \boldsymbol{a})}[V^*(\boldsymbol{s}')],$$

consisting of the expected reward from taking action $\boldsymbol{a}$ at state $\boldsymbol{s}$ and then following the optimal policy thereafter.

The corresponding Bellman equation for the optimal value function is

$$V^*(\boldsymbol{s}) = \max_{\boldsymbol{a} \in \mathcal{A}} \left\{ \mathbb{E}\left[\frac{1}{N} \sum_{i=1}^{N} r^i(\boldsymbol{s}, \boldsymbol{a})\right] + \gamma \mathbb{E}_{\boldsymbol{s}' \sim \boldsymbol{P}(\boldsymbol{s}, \boldsymbol{a})}\left[V^*(\boldsymbol{s}')\right] \right\},$$

with the population transition kernel $\boldsymbol{P} = (P^1, \cdots, P^N)$. Correspondingly, the Bellman equation for the optimal Q-function, defined from $\mathcal{S}^N \times \mathcal{A}^N$ to $\mathbb{R}$, is given by

$$Q(\boldsymbol{s}, \boldsymbol{a}) = \mathbb{E}\left[\frac{1}{N} \sum_{i=1}^{N} r^i(\boldsymbol{s}, \boldsymbol{a})\right] + \gamma \mathbb{E}_{\boldsymbol{s}' \sim \boldsymbol{P}(\boldsymbol{s}, \boldsymbol{a})}\left[\max_{\boldsymbol{a}' \in \mathcal{A}^N} Q(\boldsymbol{s}', \boldsymbol{a}')\right].$$

The optimal value function and the optimal Q-function satisfy the following relation:

$$V^*(\boldsymbol{s}) = \max_{\boldsymbol{a} \in \mathcal{A}} Q^*(\boldsymbol{s}, \boldsymbol{a}).$$

One can thus retrieve the optimal control $\pi^*(\boldsymbol{s})$ (if it exists) from $Q^*(\boldsymbol{s}, \boldsymbol{a})$, with $\pi^*(\boldsymbol{s}) \in \arg\max_{\boldsymbol{a} \in \mathcal{A}} Q^*(\boldsymbol{s}, \boldsymbol{a})$.

1.3 Challenges in Multi-agent Reinforcement Learning

MARL has enjoyed substantial successes for analyzing the otherwise challenging games. Despite its empirical success, MARL suffers from the curse of dimensionality known also as the *combinatorial nature* of MARL: its sample complexity by existing algorithms for stochastic dynamics grows exponentially with respect to the number of agents N. (See Proposition 3.2.1 in Section 3.2). In practice, this N can be on the scale of thousands or more, for instance, in rider match-up for Uber-pool and network routing for Zoom. Here we introduce two common approaches in the literature aiming to reduce sample complexity and to develop scalable learning algorithms: mean-field approximation and decentralized structure.

1.3.1 Mean-field Approximation to Multi-agent Reinforcement Learning

Mean-field approximation in MARL is to consider MARL in the regime with a large number of homogeneous agents. In this paradigm, by functional strong law of large numbers (a.k.a. propagation of chaos) [88, 119, 169, 64], non-cooperative MARLs can be approximated under Nash equilibrium by mean-field games with learning, and cooperative MARLs can be studied under Pareto optimality by analyzing mean-field controls (MFC) with learning. This approach is appealing not only because the dimension of MFC or MFG is independent of the number of agents N, but also because solutions of MFC/MFG (without learning) have been shown to provide good approximations to the corresponding N-agent game in terms of both game values and optimal strategies [79, 96, 129, 147, 149].

MFG with learning has gained popularity in the reinforcement learning (RL) community [59, 72, 82, 195, 199], with its sample complexity shown to be similar to that of single-agent RL ([59, 72]). Yet MFC with learning is by and large an uncharted field despite its potentially wide range of applications [100, 104, 180, 187]. The primary objective of this thesis is to enhance the theoretical understandings of learning MFCs. Building upon this foundation, the thesis will further focus on the development of innovative and efficient algorithms for large-population cooperative MARL. A literature review on MFCs with learning is provided in the next few paragraphs.

MKV controls/MFCs. McKean-Vlasov (MKV) processes, first introduced and studied in [118], are stochastic processes governed by stochastic differential equations whose coefficients depend on distributions of the solutions. MKV controls concern optimal controls of such systems where interchangeable agents interact through the distribution of their states and actions. As such, MKV controls are often called mean-field controls (MFCs). From the game theory perspective, MFCs are closely related to mean-field games (MFGs). Both are stochastic games with infinite number of agents, with MFGs the limiting regime of games under Nash equilibrium and MFCs that of games by Pareto optimality. Theories of MFGs

and MFCs have progressed rapidly and have been adopted in a number of fields such as physics, economics, and data science. (See [80, 97, 18, 33]). MFCs, in particular, have been broadly applied to model collective behaviors of stochastic systems with a large number of mutually interacting agents, including [63] for systemic risk assessment, [133] for a large benevolent planner such as the government or the central bank to control taxes or interest rates, and [4] for consumers to choose between new energy resources and traditional ones.

DPP for learning MFCs. The main challenge for building the MFC learning framework is to deal with probability measure space over the state-action space, and find the appropriate form of dynamic programming principle (DPP).

Widely regarded as one of the fundamental principles for control and optimization, dynamic programming principle (DPP) was first established for value functions of Markov decision problems (MDPs) in [16], and later for more general frameworks in [21, 57]. DPP was also established for the Q-functions in a learning framework of MDP in [183] (see also [22] and [163]). The DPP implies the time consistency property of the optimal control in that a current optimal policy remains so for the future. This time consistency is critical for reinforcement learning (RL): for model-free learning, time consistency of the Q-function is the key apparatus for Q-learning algorithms [184, 126] and for the actor-critic approach [92, 102]; for model-based learning, time consistency of the value function lays the foundation for value iteration and policy based algorithms [52, 53].

Most of the existing works (for example, [35, 36, 182]) focus mainly on designing MFC learning algorithms while assuming heuristically some forms of DPP. It is tempting to assume DPP given the similarity between MFCs and MDPs. Yet, MFCs are fundamentally different from MDPs: MKV systems depend on marginal distributions of both the state and the control. Consequently, MFCs are inherently time inconsistent. For instance, it has been well recognized that DPP in general does not hold for the controlled MKV system due to its non-Markovian nature [8, 27, 32]. Only recently, this time inconsistency issue for MFCs was resolved by appropriately enlarging state spaces, for example, in [99] and [138] for a finite time horizon and in [50] for a more general framework. When MFC is coupled with learning, it is unclear if, when, and how DPP will hold. This is the focus of Chapter 2.

Algorithms for learning MFCs. Another open problem for MFC with learning is, as pointed out in [129], to design efficient RL algorithms on probability measure space.

To circumvent designing algorithms on probability measure space, [36] proposed to add common noises to the underlying dynamics. This approach enables them to apply the standard RL theory for stochastic dynamics. Their model-free algorithm, however, suffers from high sample complexity as illustrated in Table 3.1 of Chapter 3, and with weak performance as demonstrated in Section 3.7. For special classes of linear-quadratic MFCs with stochastic dynamics, [35] explored the policy gradient method and [113] developed an actor-critic type algorithm. In Chapter 3, a model-free kernel-based Q-learning algorithm will be proposed, with state-of-art convergence guarantee.

1.3.2 Decentralized Structure in Multi-agent Reinforcement Learning

Another approach to tackle the curse of dimensionality is to focus on exploiting localized structures of MARL problems and designing decentralized learning algorithms to reduce the complexity. This approach is also inspired by a large class of practical MARL problems in which each individual agent has only limited or partial information of the entire system. In such a system, it is necessary to design algorithms to learn policies of the decentralized type, i.e., policies that depend only on the *local* information of each agent.

In a simulated or laboratory setting, decentralized policies may be learned in a centralized fashion. It is to train a central controller to dictate the actions of all agents. Such paradigm of *centralized training with decentralized execution* has achieved significant empirical successes, especially with the computational power of deep neural networks [112, 58, 40, 145, 197, 173]. However, such a training approach still suffers from the curse of dimensionality since the global information is needed throughout the training [205]; it also requires extensive and costly communications between the central controller and all agents [143]. Moreover, policies derived from the centralized training stage may not be robust in the execution phase [203]. Most importantly, this approach has not been supported or analyzed theoretically. A literature review on such paradigm is given in the ext paragraph.

Centralized training with decentralized execution. Most of the existing works in this paradigm can be summarized into two categories: value-based method [162, 145, 197, 160] and actor-critic method [112, 58]. For the first category, Value Decomposition Network (VDN) [162] proposes to directly factorize the joint value function into a summation of individual value functions; QMIX [145] augments the summation to be non-linear aggregations, while maintaining a monotonic relationship between centralized and individual value functions; QTRAN [160] introduces a refined learning objective on top of QMIX along with specific network designs; Determinantal Q-Learning [197] utilizes the idea of determinantal point process and promotes agents t acquire diverse behavioral models to allow natural factorization of the joint Q-function without no prior structure constraints. For the second category, COMA [58] proposes a centralized critic to estimate the Q-function and decentralized actors to optimize the agents' policies; Multi-agent DDPG (MADDPG) [112] uses separate actors and critics for each agent and train the critic in a centralised way and use the actor in execution. However, none of the above mentioned methods has provable convergence and sample complexity guarantee.

Network structure in MARL. An alternative and promising paradigm is to take into consideration the network structure of the system to train decentralized policies. Compared with the centralized training approach, exploiting network structures makes the training procedure more efficient as it allows the algorithm to be updated with parallel computing and reduces communication cost.

There are two distinct types of network structures. The first is the *network of agents*, often found in social networks such as Facebook and Twitter, as well as team video games including StarCraft II. This network describes *interactions and relations among heterogeneous agents*. For MARL systems with such network of agents, [206] establishes the asymptotic convergence of decentralized-actor-critic algorithms which are scalable in agent actions. Similar ideas are extended to the continuous space where deterministic policy gradient method (DPG) is used [204], with finite-sample analysis for such framework established in the batch setting [207]. [142] studies a network of agents where state and action interact in a local manner; by exploiting the network structure and the exponential decay property of the Q-function, it proposes an actor-critic framework scalable in both actions and states. Similar framework is considered for the linear quadratic case with local policy gradients conducted with zero order optimization and parallel updating [101].

The second type of network, *the network of states*, has been frequently used for modeling self-driving vehicles, ride-sharing, and data and traffic routing. It focuses on the *state of agents*. Compared with the network of agents which is *static* from agent's perspective [162], the network of states is *stochastic*: neighboring agents of any given agent may change dynamically. This type of network has been empirically studied in various applications, including packet routing [200], traffic routing [30, 71], resource allocations [31] and social economic systems [208]. However, there is no existing theoretical analysis for this type of decentralized MARL. Moreover, the dynamic nature of agents' relationship makes it difficult to adopt existing methodology from the static network of agents. Chapter 4 aims to propose a framework of *localized training and decentralized execution* for cooperative MARL with network of states and mean-field approximation.

1.4 Contribution

The main contributions of this thesis are summarized as follows.

In Chapter 2, we first present a simple example in MFCs with learning where DPP fails with a mis-specified Q-function; and then propose the correct form of Q-function in an appropriate space for MFCs with learning. This particular form of Q-function is different from the classical one and is called the IQ-function. In the special case when the transition probability and the reward are independent of the mean-field information, it *integrates* the classical Q-function for single-agent RL over the state-action distribution. In other words, MFCs with learning can be viewed as lifting the classical RLs by replacing the state-action space with its probability distribution space. This identification of the IQ-function enables us to establish precisely the DPP in the learning framework of MFCs. Finally, we illustrate through numerical experiments the time consistency of this IQ-function.

Chapter 3 builds the mathematical framework to approximate cooperative MARL by MFCs with learning. The approximation error is shown to be of $\mathcal{O}(\frac{1}{\sqrt{N}})$ (N the number of agents). It then proposes an efficient kernel-based algorithm (MFC-K-Q) for MFC with learning. This model-free Q-learning-based algorithm combines the technique of kernel

regression with approximated Bellman operator. The convergence rate and the sample complexity of this algorithm are shown to be independent of the number of agents N, and rely only on the size of the state-action space of the underlying single-agent dynamics (Table 3.1). As far as we are aware of, there is no prior algorithm with linear convergence rate for cooperative MARL. Our experiment in Section 3.7 demonstrates that MFC-K-Q avoids the curse of dimensionality and outperforms both existing MARL algorithms and MFC algorithms, especially in the large-population regime (when $N > 50$).

Chapter 4 proposes a framework of localized training and decentralized execution for cooperative MARL with *network of states* and mean-field approximation, to study MARL systems such as self-driving vehicles, ride-sharing, and data and traffic routing. In this network, agents can move from one state to any connecting state, and observe only partial information of the entire system in an aggregated fashion. Localized training is to collect local information in agents' neighboring states for training; decentralized execution means to execute the learned decentralized policies that depend only on agents' current states. The theoretical analysis consists of three key components: the first is to establish the mean-field reformulation of the original MARL system as a networked MDP with teams of agents, enabling updating locally the associated team Q-function; the second is to develop the DPP for the mean-field type of Q-function for each team on the probability measure space; and the third is to analyze the exponential decay property of the Q-function, facilitating its approximation with sample efficiency and with controllable error. The analysis leads to a neural-network-based algorithm LTDE-Neural-AC, where the actor-critic approach is coupled with over-parameterized neural networks. Convergence and sample complexity of the algorithm are established and shown to be scalable with respect to the size of agents and states.

Chapter 2

Dynamic Programming Principles for Learning Mean-Field Controls

Dynamic programming principle (DPP) is fundamental for control and optimization, including Markov decision problems (MDPs), reinforcement learning (RL), and more recently mean-field controls (MFCs). However, in the learning framework of MFCs, DPP has not been rigorously established, despite its critical importance for algorithm designs. In this chapter, we first present a simple example in MFCs with learning where DPP fails with a mis-specified Q-function; and then propose the correct form of Q-function in an appropriate space for MFCs with learning. This particular form of Q-function is different from the classical one and is called the IQ-function. In the special case when the transition probability and the reward are independent of the mean-field information, it *integrates* the classical Q-function for single-agent RL over the state-action distribution. In other words, MFCs with learning can be viewed as lifting the classical RLs by replacing the state-action space with its probability distribution space. This identification of the IQ-function enables us to establish precisely the DPP in the learning framework of MFCs. Finally, we illustrate through numerical experiments the time consistency of this IQ-function.

2.1 Motivation and Related Works

DPP. Widely regarded as one of the fundamental principles for control and optimization, dynamic programming principle (DPP) was first established for value functions of Markov decision problems (MDPs) in [16], and later for more general frameworks in [21, 57]. DPP was also established for the Q-functions in a learning framework of MDP in [183] (see also [22] and [163]). The DPP implies the time consistency property of the optimal control in that a current optimal policy remains so for the future. This time consistency is critical for reinforcement learning (RL): for model-free learning, time consistency of the Q-function is the key apparatus for Q-learning algorithms [184, 126] and for the actor-critic approach [92, 102]; for model-based learning, time consistency of the value function lays the foundation

for value iteration and policy based algorithms [52, 53]. More recently, the time consistency property has been analyzed in a series of papers for mean-field controls (MFCs) also known as McKean-Vlasov (MKV) controls [99, 138, 50], without the context of learning.

MKV controls/MFCs. McKean-Vlasov (MKV) processes, first introduced and studied in [118], are stochastic processes governed by stochastic differential equations whose coefficients depend on distributions of the solutions. MKV controls concern optimal controls of such systems where interchangeable agents interact through the distribution of their states and actions. As such, MKV controls are often called mean-field controls (MFCs).

From the game theory perspective, MFCs are closely related to mean-field games (MFGs). Both are stochastic games with infinite number of agents, with MFGs the limiting regime of games under Nash equilibrium and MFCs that of games by Pareto optimality. Theories of MFGs and MFCs have progressed rapidly and have been adopted in a number of fields such as physics, economics, and data science. (See [80, 97, 18, 33]). MFCs, in particular, have been broadly applied to model collective behaviors of stochastic systems with a large number of mutually interacting agents, including [63] for systemic risk assessment, [133] for a large benevolent planner such as the government or the central bank to control taxes or interest rates, and [4] for consumers to choose between new energy resources and traditional ones.

MFCs with learning and DPP. For many of the stochastic systems with a large population of agents, model parameters and dynamics are in general unknown *a priori* and learning algorithms are essential for the agents to improve their decisions while interacting with the (partially) unknown system and other agents. In this case, multi-agent reinforcement learning (MARL) has enjoyed substantial successes for analyzing the otherwise challenging games, including two-agent or two-team computer games [155, 176], self-driving vehicles [154], real-time bidding games [87], ride-sharing [100], and traffic routing [54]. Despite its empirical success, MARL suffers from the curse of dimensionality known also as the *combinatorial nature* of MARL: its sample complexity by existing algorithms for stochastic dynamics grows exponentially with respect to the number of agents N. In practice, this N can be on the scale of thousands or more, for instance, in rider match-up for Uber-pool and network routing for Zoom. MFCs, on the other hand, provide good approximations to the multi-agent system and address the curse of dimensionality suffered in most of the existing MARL algorithms. It is therefore natural meanwhile important to consider the learning problem in MFCs.

Despite its potential for improving existing MARL algorithms, theory of MFCs with learning remains by and large undeveloped. Instead, almost all works (for example, [35, 36, 182]) focus mainly on learning algorithms while assuming heuristically some forms of DPP.

It is tempting to assume DPP given the similarity between MFCs and MDPs. Yet, MFCs are fundamentally different from MDPs: MKV systems depend on marginal distributions of both the state and the control. Consequently, MFCs are inherently time inconsistent. For instance, it has been well recognized that DPP in general does not hold for the controlled

MKV system due to its non-Markovian nature [8, 27, 32]. Only recently, this time inconsistency issue for MFCs was resolved by appropriately enlarging state spaces, for example, in [99] and [138] for a finite time horizon and in [50] for a more general framework. When MFC is coupled with learning, it is unclear if, when, and how DPP will hold. This is the focus of this chapter.

Time consistency in MFCs with learning. In this chapter, we will first present a simple example (Example 2.3.1 in Section 2.3.1) to demonstrate the time inconsistency issue for MFCs with learning. This example shows that when the Q-function is mis-specified, Q table will converge to different values with different initial population distributions.

We will then establish precisely the DPP by identifying a correct form of the Q-function in an appropriate space. This particular form of the Q-function reflects the essence of MFCs: MFC is equivalent to an auxiliary control problem in which the objective function depends on the cost functional of every agent for the purpose of social optimality. This control perspective enables us to specify the Q-function as an integral form of the classical Q-function over the state-action distribution of each agent. To distinguish such Q-function from the classical one, we called it integrated Q-function (IQ). (See also Section 2.3.5).

Next, we derive the suitable form of DPP for this IQ-function. This DPP generalizes the classical DPP for Q-learning of MDP to that of MKV system, and extends the DPP for MFCs to the learning framework. To accommodate model-based learning for MFCs, we also obtain the corresponding DPP for the value function.

Finally, we illustrate through numerical experiments the time consistency of the IQ-function.

Relation to existing works. Our analysis and framework for establishing DPP for MFCs with learning differ fundamentally from those in [99, 138, 50] on DPP for value functions of MFCs without learning.

The first is the adoption of relaxed controls instead of strict controls used in these earlier works. As illustrated in Example 2.3.1 in Section 2.3.1, optimal controls for MFCs with learning are intrinsically relaxed types, whereas classic control problems with concave reward functions are inevitably strict even for MFGs [93]. Relaxed controls are essential for learning, and in particular for RL which is characterized with exploration and exploitation. Exploration relies on randomized strategies with actions sampled from a distribution of actions. Relaxed controls are known also as mixed strategies in game theory [47, 190, 116], also for MFC without learning in [94]. Moreover, incorporation of entropy regularization in many machine learning problems would destroy the convexity or the concavity structure of the value function, and optimal controls are necessarily relaxed ones.

The second is the aforementioned IQ-function, identified and analyzed for the first time in the learning framework on the lifted probability measure space with relaxed controls.

To the best of our knowledge, this is the first time that DPP is rigorously established for MFCs with learning. This form of DPP provides one critical insight: learning problems

with MFCs can be recast as general forms of RLs where the state variable is replaced by the probability distribution. This reformulation paves the way for developing efficient value-based and policy-based algorithms for MFCs with learning. It is also the first step towards future theoretical development of learning problem with MFCs. For instance, [129] has further established the DPP for learning in a discrete-time model with the incorporation of common noise and with open-loop controls.

Outline of the chapter. The rest of the chapter is organized as follows. Section 2 presents the mathematical framework of MFCs with learning. Section 3 introduces the IQ-function and establishes DPPs for both the IQ-function and the value function. Section 4 concludes by revisiting Example 2.3.1 with the performance of the IQ-function. Section 5 demonstrates an example on equilibrium pricing with IQ-function.

Notations.

- Let (X, d_X) be a metric space and X is equipped with the Borel σ-field $\mathcal{B}(X)$, meaning the σ-field generated by the open sets of X. Denote $\mathcal{P}(X)$ for the set of (Borel) probability measures on X. When (X, d_X) is a *compact* metric space, any probability measure $\mu \in \mathcal{P}(X)$ has a first moment. $\mathcal{W}_1$ denotes the Wasserstein distance of order 1 such that

$$\mathcal{W}_1(\mu, \mu') = \inf\left\{\left(\int_{X\times X} d_X(x, x')\nu(dx, dx')\right): \nu \in \mathcal{P}(X \times X) \text{ with marginals } \mu, \mu' \in \mathcal{P}(X)\right\}.$$

  $\mathcal{P}(X)$ is always equipped with $\mathcal{W}_1(\mu, \mu')$. Note that the Borel σ-field $\mathcal{B}(\mathcal{P}(X))$ generated by $\mathcal{W}_1$ is equivalent to the weak topology induced by the evaluation $\mathcal{P}(X) \ni \mu \mapsto \mu(C)$ for any Borel set $C \in \mathcal{B}(X)$. (See e.g. [175] and [93]).

- When X is finite, $\mathcal{P}(X) = \left\{(p_i)_{i=1}^{|X|} \in \mathbb{R}^{|X|} : \sum_{i=1}^{|X|} p_i = 1, p_i \geq 0\right\}$ is the probability simplex in $\mathbb{R}^{|X|}$, where $|X|$ denotes the size of X; Moreover, X is always equipped with discrete metric, i.e., $d(x, x') = \mathbf{1}_{\{x\neq x'\}}$. In this case, $\mathcal{W}_1$ is equivalent to the L^1-norm. (See e.g. [67]).

- For a metric space X, $\mathcal{M}(X)$ denotes the set of all real-valued measurable functions on X. For each bounded $f \in \mathcal{M}(X)$, the sup norm of f is defined as $\|f\|_\infty = \sup_{x\in X} |f(x)|$.

- Denote $(\Omega, \mathcal{F} = \{\mathcal{F}_t\}_{t=0}^\infty, \mathbb{P})$ as a probability space with Ω being Polish space, $\mathcal{F}$ its Borel σ field and $\mathbb{P}$ an atomless probability measure, and denote $L(\Omega, \mathcal{F}, \mathbb{P}; X)$ as the space of all X-valued random variables; $(\Omega, \mathcal{F} = \{\mathcal{F}_t\}_{t=0}^\infty, \mathbb{P})$ is "rich" in the sense that for any $\mu \in \mathcal{P}(X)$, there exists $\xi \in L(\Omega, \mathcal{F}, \mathbb{P}; X)$ satisfying $\xi \sim \mu$.

- Given two measurable spaces $(Y, \mathcal{B}(Y))$ and $(X, \mathcal{B}(X))$, we say a measure-valued function $f : Y \to \mathcal{P}(X)$ is measurable if $\Lambda_C \circ f : Y \to [0,1]$ is measurable for any $C \in \mathcal{B}(X)$, where $\Lambda_C : \mathcal{P}(X) \ni \mu \mapsto \mu(C) \in [0,1]$ (or equivalently, if $f^{-1}(C) \in \mathcal{B}(Y)$ for every $C \in \mathcal{B}(\mathcal{P}(X))$).
- Given two measurable spaces $(X, \mathcal{B}(X))$ and $(Y, \mathcal{B}(Y))$, for a measurable function $f : X \to Y$ and a measure $\mu \in \mathcal{P}(X)$, the pushforward measure $f \star \mu$ is defined to be a probability measure on $\mathcal{B}(Y)$: $f \star \mu(C) = \mu(f^{-1}(C))$ for any $C \in \mathcal{B}(Y)$.
- Given a metric space X, δ_x denotes the Dirac measure on some fixed point $x \in X$. $\mathbb{N}$ denotes the set of all positive integers.

2.2 The Mathematical Framework of Learning Mean-Field Controls

2.2.1 Review: Single-Agent Reinforcement Learning

Before introducing the mathematical framework of MFCs with learning, let us review relevant terminologies for single-agent RL.

Let us start with a discrete time MDP in an infinite time horizon of the following form

$$v(s) = \sup_{\pi} v^{\pi}(s) := \sup_{\pi} \mathbb{E}^{\pi}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \middle| s_0 = s\right], \tag{2.2.1}$$

subject to

$$s_{t+1} \sim P(s_t, a_t), \quad a_t \sim \pi_t(s_t), \ \ t \in \mathbb{N} \cup \{0\}. \tag{2.2.2}$$

Here and throughout the chapter, $\mathbb{E}^{\pi}$ denotes the expectation under control π; the state space $(\mathcal{S}, d_{\mathcal{S}})$ and the action space $(\mathcal{A}, d_{\mathcal{A}})$ are two compact separable metric space , including the case of $\mathcal{S}$ and $\mathcal{A}$ being finite, as often seen in RL literature; $\gamma \in (0,1)$ is a discount factor; $s_t \in \mathcal{S}$ and $a_t \in \mathcal{A}$ are the state and the action at time t; $P : \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathcal{S})$ is the transition matrix of the underlying Markov system; the reward $r(s,a)$ is random valued in $\mathbb{R}$ for each $(s,a) \in \mathcal{S} \times \mathcal{A}$; and the control $\pi = \{\pi_t\}_{t=0}^{\infty}$ can be either deterministic such that $\pi_t : \mathcal{S} \to \mathcal{A}$, or randomized such that $\pi_t : \mathcal{S} \to \mathcal{P}(\mathcal{A})$. Note that our results can be easily extended to the situation where $(\mathcal{S}, d_{\mathcal{S}})$ and $(\mathcal{A}, d_{\mathcal{A}})$ are not compact but the measures under consideration have a first moment.

When the transition dynamics P and the reward function r are unknown, this MDP becomes an RL problem, which is to find an optimal control π (if it exists) while simultaneously learning the unknown P and r. The learning of P and r can be either explicit or implicit, which leads to model-based and model-free RL, respectively.

One basic model-free algorithm for RL is the Q-learning algorithm, in which a Q-function is defined as

$$Q(s,a) = \mathbb{E}[r(s,a)] + \gamma \mathbb{E}_{s' \sim P(s,a)}[v(s')]. \tag{2.2.3}$$

The well-known DPP for such Q-function is expressed in the form of the following Bellman equation

$$Q(s,a) = \mathbb{E}[r(s,a)] + \gamma \mathbb{E}_{s' \sim P(s,a)} \sup_{a' \in \mathcal{A}} Q(s',a'). \tag{2.2.4}$$

Meanwhile, the Bellman equation for the value function is

$$v(s) = \sup_{a \in \mathcal{A}} \big\{ \mathbb{E}[r(s,a)] + \gamma \mathbb{E}_{s' \sim P(s,a)}[v(s')] \big\}. \tag{2.2.5}$$

By the definition of Q-function and (2.2.5), the value function and the Q-function are closely connected by the following relation

$$v(s) = \sup_{a \in \mathcal{A}} Q(s,a).$$

Thus, one can retrieve the optimal (stationary) control $\pi^*(s)$ (if it exists) from $Q(s,a)$, i.e., $\pi^*(s) \in \arg\max_{a \in \mathcal{A}} Q(s,a)$.

2.2.2 Mathematical Framework of MFCs with Learning

Our MFC framework is motivated from cooperative N-agent games. To see this, assume that there are N homogeneous agents. At each time step $t \in \mathbb{N} \cup \{0\}$, the state and the action of each agent i $(= 1, \cdots, N)$ is denoted as $s_t^i \in \mathcal{S}$ and $a_t^i \in \mathcal{A}$. Each agent i moves to the next state s_{t+1}^i according to the transition probability $P(s_t^i, \mu_t^N, a_t^i, \nu_t^N)(\cdot)$ and receives a reward $r_t^i \sim R(s_t^i, \mu_t^N, a_t^i, \nu_t^N)(\cdot)$, where $\mu_t^N = \frac{1}{N}\sum_{i=1}^N \delta_{s_t^i}$ and $\nu_t^N = \frac{1}{N}\sum_{i=1}^N \delta_{a_t^i}$ are the empirical distributions of s_t^i and a_t^i, $i = 1, \ldots, N$; the probability transition P is a measurable function from $\mathcal{S} \times \mathcal{P}(\mathcal{S}) \times \mathcal{A} \times \mathcal{P}(\mathcal{A})$ to $\mathcal{P}(\mathcal{S})$ and is unknown; and the distribution of the reward function $R : \mathcal{S} \times \mathcal{P}(\mathcal{S}) \times \mathcal{A} \times \mathcal{P}(\mathcal{A}) \to \mathcal{P}(\mathbb{R})$ is measurable and unknown.

Now, taking $N \to \infty$, by law of large number, we can consider MFCs, which are stochastic games under Pareto optimality with infinitely many identical, indistinguishable, and interchangeable agents. We can define analogously the learning framework for MFCs over the infinite horizon with the same terminology $\mathcal{S}$, $\mathcal{P}(\mathcal{S})$, $\mathcal{A}$, $\mathcal{P}(\mathcal{A})$, R, and γ used in the RL framework. Due to the indistinguishability of agents, one can focus on a single representative agent and consider an auxiliary control problem in which the objective function depends on the average cost/reward of every agent.

At each time $t \in \mathbb{N} \cup \{0\}$, the state of the representative agent is $s_t \in \mathcal{S}$. Given the population state distribution, i.e., the probability distribution $\mu_t \in \mathcal{P}(\mathcal{S})$ of state s_t, the representative agent takes an action $a_t \in \mathcal{A}$ according to some control π_t. She will receive

an instantaneous stochastic reward $r_t = r(s_t, \mu_t, a_t, \nu_t) \sim R(s_t, \mu_t, a_t, \nu_t)(\cdot)$ and her state will move to the next state s_{t+1} according to a probability transition function of mean-field type $P(s_t, \mu_t, a_t, \nu_t)(\cdot)$. Here $\nu_t \in \mathcal{P}(\mathcal{A})$ denotes the action distribution at time t.

The (accumulated) reward of the auxiliary control problem, given the initial state $s_0 = \xi \in L(\Omega, \mathcal{F}, \mathbb{P}; \mathcal{S})$, and given the control $\pi = \{\pi_t\}_{t=0}^{\infty}$, is defined as

$$V^{\pi}(\xi) \;=\; \mathbb{E}^{\pi}\Big[\sum_{t=0}^{\infty} \gamma^t r(s_t, \mu_t, a_t, \nu_t)\Big| s_0 = \xi\Big], \tag{2.2.6}$$

subject to

$$s_{t+1} \sim P(s_t, \mu_t, a_t, \nu_t)(\cdot), \quad a_t \sim \pi_t(s_t, \mu_t)(\cdot), \quad r(s_t, \mu_t, a_t, \nu_t) \sim R(s_t, \mu_t, a_t, \nu_t)(\cdot). \tag{2.2.7}$$

The admissible controls are of feedback forms and relaxed types. That is, at each time t, $\pi_t = \pi_t(s_t, \mu_t)$ and $\pi_t : \mathcal{S} \times \mathcal{P}(\mathcal{S}) \to \mathcal{P}(\mathcal{A})$ is measurable and maps the current state and the current state distribution to a distribution over the action space. We denote by Π_t such set of admissible controls starting from time $t \in \mathbb{N} \cup \{0\}$, and set $\Pi = \Pi_0$. Note that a relaxed control differs from a strict control, which is a measurable function defined from $\mathcal{S} \times \mathcal{P}(\mathcal{S})$ to $\mathcal{A}$. Clearly a strict control α_t is a relaxed control with a special form of $\pi_t = \delta_{\alpha_t}$, the point mass at some measurable function $\alpha_t : \mathcal{S} \times \mathcal{P}(\mathcal{S}) \to \mathcal{A}$. Note that under a feedback relaxed control π_t, we have $\nu_t(\cdot) = \int_{s \in \mathcal{S}} \pi_t(s, \mu_t)(\cdot)\mu_t(ds) \in \mathcal{P}(\mathcal{A})$.

The objective of the auxiliary controller is to find

$$V(\xi) \;=\; \sup_{\pi \in \Pi} V^{\pi}(\xi), \quad \text{for any } \xi \in L(\Omega, \mathcal{F}, \mathbb{P}; \mathcal{S}), \tag{2.2.8}$$

and to search for an optimal control (if it exists).

Note that the nature of MFCs is different from the single-agent RL (2.2.1)-(2.2.2) in that it reflects the nature of MFC that the representative agent interacts with all agents via probability distributions of states μ_t and actions ν_t.

To ensure the well-definedness of this learning problem for MFC (2.2.6)-(2.2.8), throughout the chapter we assume:

Outstanding Assumption (**A**). For any initial state $s_0 = \xi \sim \mu$,

$$\sup_{\pi \in \Pi} \mathbb{E}^{\pi}\Big[\sum_{t=0}^{\infty} \gamma^t \big|r(s_t, \mu_t, a_t, \nu_t)\big|\Big] \;<\; \infty.$$

It is clear that when $\|r\|_{\infty} \le r_{\max}$, a.s. for some $r_{\max} > 0$, condition in Outstanding assumption (A) is satisfied. In general, the following conditions (A1)-(A3) will ensure Outstanding Assumption (A).

(A1) For fixed arbitrary $(s^o, \delta_{s^o}, a^o, \delta_{a^o}) \in \mathcal{S} \times \mathcal{P}(\mathcal{S}) \times \mathcal{A} \times \mathcal{P}(\mathcal{A})$, there exists some positive constant L_P such that for every $(s, \mu, a, \nu) \in \mathcal{S} \times \mathcal{P}(\mathcal{S}) \times \mathcal{A} \times \mathcal{P}(\mathcal{A})$,

$$\int_{s' \in \mathcal{S}} d_{\mathcal{S}}(s', s^o)\Big(P(s, \mu, a, \nu)(ds') - P(s^o, \delta_{s^o}, a^o, \delta_{a^o})(ds')\Big) \\ \le L_P\Big(d_{\mathcal{S}}(s, s^o) + d_{\mathcal{A}}(a, a^o) + \mathcal{W}_1(\mu, \delta_{s^o}) + \mathcal{W}_1(\nu, \delta_{a^o})\Big).$$

(A2) For fixed arbitrary $(s^o, \delta_{s^o}, a^o, \delta_{a^o}) \in \mathcal{S} \times \mathcal{P}(\mathcal{S}) \times \mathcal{A} \times \mathcal{P}(\mathcal{A})$, there exists some positive constant L_R such that for every $(s, \mu, a, \nu) \in \mathcal{S} \times \mathcal{P}(\mathcal{S}) \times \mathcal{A} \times \mathcal{P}(\mathcal{A})$,

$$\int_{\mathbb{R}} |r| \Big(R(s, \mu, a, \nu)(dr) - R(s^o, \delta_{s^o}, a^o, \delta_{a^o})(dr) \Big) \\ \leq L_R \Big(d_{\mathcal{S}}(s, s^o) + d_{\mathcal{A}}(a, a^o) + \mathcal{W}_1(\mu, \delta_{s^o}) + \mathcal{W}_1(\nu, \delta_{a^o}) \Big).$$

(A3) For fixed arbitrary $(s^o, \delta_{s^o}, a^o) \in \mathcal{S} \times \mathcal{P}(\mathcal{S}) \times \mathcal{A}$, there exists some positive constant L_π such that for every $(s, \mu) \in \mathcal{S} \times \mathcal{P}(\mathcal{S})$

$$\int_{a \in \mathcal{A}} d_{\mathcal{A}}(a, a^o) \Big(\pi(s, \mu)(da) - \pi(s^o, \delta_{s^o})(da) \Big) \leq L_\pi \Big(d_{\mathcal{S}}(s, s^o) + \mathcal{W}_1(\mu, \delta_{s^o}) \Big), \\ \int_{a \in \mathcal{A}} d_{\mathcal{A}}(a, a^o) \pi(s^o, \delta_{s^o})(da) < +\infty.$$

In the MFC formulation, it is important to view alternatively the control π_t as a measurable mapping from $\mathcal{P}(\mathcal{S})$ to $\mathcal{H}$. For notational simplicity, set $h_t := \pi_t(\mu_t) \in \mathcal{H}$, where

$$\mathcal{H} = \{h \mid h : \mathcal{S} \to \mathcal{P}(\mathcal{A}) \text{ is measurable such that Outstanding Assumption (A) holds}\} \quad (2.2.9)$$

Here $\mathcal{H}$ contains all "local" policies that depend only on the state variable.

We first show that the probability distribution of the dynamics $\{\mu_t\}_{t=0}^{\infty}$ in (2.2.7) satisfies the following flow property.

Lemma 2.2.1 *(Flow property of μ_t) Under Outstanding Assumption (A), for any given admissible policy $\pi \in \Pi$ and the initial state distribution μ, the evolution of the state distribution $\{\mu_t\}_{t=0}^{\infty}$ in (2.2.7) follows*

$$\mu_{t+1} \quad = \quad \Phi(\mu_t, \pi_t(\mu_t)), \ \mu_0 = \mu, \ \ t \in \mathbb{N} \cup \{0\}. \quad (2.2.10)$$

Here $\Phi : \mathcal{P}(\mathcal{S}) \times \mathcal{H} \to \mathcal{P}(\mathcal{S})$ is measurable and defined by

$$\Phi(\mu, h)(ds') \quad := \quad \int_{s \in \mathcal{S}} \mu(ds) \int_{a \in \mathcal{A}} h(s)(da) P(s, \mu, a, \nu(\mu, h))(ds'), \quad (2.2.11)$$

for any $(\mu, h) \in \mathcal{P}(\mathcal{S}) \times \mathcal{H}$ and $\nu(\mu, h)(\cdot) := \int_{s \in \mathcal{S}} h(s)(\cdot)\mu(s) \in \mathcal{P}(\mathcal{A})$. In particular, when $h_t = \delta_{\alpha_t}$ for some measurable function $\alpha_t : \mathcal{S} \to \mathcal{A}$ (i.e., h_t is a strict control),

$$\mu_{t+1} = \Phi(\mu_t, \delta_{\alpha_t}) \quad = \quad \int_{s \in \mathcal{S}} \mu_t(ds) P(s, \mu_t, \alpha_t(s), \alpha_t \star \mu_t)), \ \ t \in \mathbb{N} \cup \{0\},$$

where $\alpha_t \star \mu_t$ is the pushforward of measure μ_t.

Proof of Lemma 2.2.1 Fix $\pi = \{\pi_t\}_{t=0}^{\infty} \in \Pi$. For any bounded measurable function φ on $\mathcal{S}$, by the law of iterated conditional expectation,

$$\begin{aligned}
\mathbb{E}^{\pi}[\varphi(s_{t+1})] &= \mathbb{E}^{\pi}\Big[\mathbb{E}^{\pi}\big[\varphi(s_{t+1})\big|s_1 \cdots, s_t\big]\Big] \\
&= \mathbb{E}^{\pi}\Big[\int_{s' \in \mathcal{S}} \varphi(s') P(s_t, \mu_t, a_t, \nu_t)(ds')\Big] \\
&= \int_{s' \in \mathcal{S}} \varphi(s') \mathbb{E}^{\pi}\Big[P(s_t, \mu_t, a_t, \nu_t)(ds')\Big] \\
&= \int_{s' \in \mathcal{S}} \varphi(s') \int_{s \in \mathcal{S}} \mu_t(ds) \int_{a \in \mathcal{A}} \pi_t(s, \mu_t)(da) P(s, \mu_t, a, \nu(\mu_t, \pi_t(\mu_t)))(ds').
\end{aligned}$$

□

Now, given Outstanding Assumption (A), adopting the technique from [138] for strict controls, we can show that the value function $V^{\pi}(\xi)$ for relaxed controls can still be rewritten in terms of the state distribution flow $\{\mu_t\}_{t=0}^{\infty}$ and that it depends on the initial random variable ξ only through the probability distribution μ. In other words, $V^{\pi}(\xi)$ can be written as $v^{\pi}(\mu)$ for some function $v^{\pi} : \mathcal{P}(\mathcal{S}) \to \mathbb{R}$. More precisely,

Lemma 2.2.2 *(Law-invariant property) Under Outstanding Assumption (A), given any $\pi \in \Pi$, $V^{\pi}(\xi)$ in (2.2.6) can be written as*

$$v^{\pi}(\mu) = \sum_{t=0}^{\infty} \gamma^t \widehat{r}(\mu_t, \pi_t(\mu_t)), \tag{2.2.12}$$

where the integrated averaged reward function $\widehat{r}$ is the measurable function from $\mathcal{P}(\mathcal{S}) \times \mathcal{H}$ to $\mathbb{R}$ such that

$$\widehat{r}(\mu, h) := \int_{s \in \mathcal{S}} \mu(ds) \int_{a \in \mathcal{A}} h(s)(da) \int_{r \in \mathbb{R}} r R(s, \mu, a, \nu(\mu, h))(dr). \tag{2.2.13}$$

In particular, if $h_t = \delta_{\alpha_t}$ for some $\alpha_t : \mathcal{S} \to \mathcal{A}$, $t \in \mathbb{N} \cup \{0\}$ (i.e., π_t is a strict control), and $R(s, \mu, a, \nu)(\cdot) = \delta_{r(s,\mu,a,\nu)}(\cdot)$ for some $r : \mathcal{S} \times \mathcal{P}(\mathcal{S}) \times \mathcal{A} \times \mathcal{P}(\mathcal{A}) \to \mathbb{R}$, then with a slight abuse of notation, we can write $v^{\pi}(\mu) = v^{\alpha}(\mu)$ and

$$v^{\alpha}(\mu) = \sum_{t=0}^{\infty} \gamma^t \widehat{r}(\mu_t, \delta_{\alpha_t}) = \sum_{t=0}^{\infty} \gamma^t \int_{s \in \mathcal{S}} r(s, \mu_t, \alpha_t(s), \alpha_t \star \mu_t) \mu_t(ds).$$

The flow property of $\{\mu_t\}_{t=0}^{\infty}$ and the law-invariant property of v^{π} in the above lemmas suggest that MFCs with learning may be viewed as an RL problem with the state variable s_t replaced by the probability distribution μ_t. This view is useful for subsequent analysis, and in particular critical for establishing the DPP, the main result of the chapter.

2.3 DPP for Learning Mean-Field Controls

2.3.1 Time Inconsistency: An Example

Recall from Section 2.2.1 the Bellman equation for the Q-function in classical single-agent RL,

$$Q(s,a) = \mathbb{E}[r(s,a)] + \mathbb{E}_{s'\sim P(s,a)} \sup_{a'\in\mathcal{A}} Q(s',a').$$

It is tempting to define such Q-function for MFC in the learning framework. Unfortunately, such Q-function will not be time consistent, as demonstrated in the following example.

Example 2.3.1 *Take a two-state dynamic system with two choices of actions. Denote the state space as* $\mathcal{S} = \{L, H\}$ *and the action space as* $\mathcal{A} = \{\mathrm{ST}, \mathrm{MV}\}$. *The transition probability goes as follows:*

$$P(s,a,s') = \lambda_s \mathbf{1}_{\{a=\mathrm{MV}\}}, \quad \text{if } s' \neq s \in \mathcal{S}, \; P(s,a,s') = 1 - \lambda_s \mathbf{1}_{\{a=\mathrm{MV}\}}, \quad \text{if } s' = s \in \mathcal{S},$$

with $\lambda_s \in [0,1]$ *for* $s \in \mathcal{S}$. *Here* $P(s,a,s')$ *is the probability of moving to state* s' *when the agent in state* s *takes the action* a. *That is, when the agent in the state* s *takes action* ST, *she will stay at the current state* s; *when the agent in the state* s *takes the action* MV, *she will move to a different state* s' *with probability* $0 \leq \lambda_s \leq 1$, $s \in \mathcal{S}$ *and stay at state* s *with probability* $1-\lambda_s$, $s \in \mathcal{S}$. *After each action, the representative agent will receive a reward* $r_t = \mathbf{1}_{\{s_t=H\}} - \big(\mathbb{E}[\mathbf{1}_{\{s_t=H\}}]\big)^2 - \lambda \mathcal{W}_1(\mu_t, B)$. *Here* μ_t *denotes the probability distribution of the state at time* t, B *is a given Binomial distribution with parameter* p *(*$1-\lambda_L \leq p \leq \lambda_H$*), and* $\lambda > 0$ *is a scalar parameter. Fix any arbitrary initial state distribution* $\mu_0 = p_0 \mathbf{1}_{\{s_0=L\}} + (1-p_0)\mathbf{1}_{\{s_0=H\}}$ *with some* $0 \leq p_0 \leq 1$. *If taking the standard Q-function with the state variable* s *and the action variable* a *instead of the state distribution* μ *and the local policy* h, *this leads to the standard Q-learning update:*

$$Q_{t+1}(s_t,a_t) = (1-l_t)Q_t(s_t,a_t) + l_t \times \big(r_t + \gamma \times \max_{a'\in\mathcal{A}}(Q_t(s_{t+1},a')\big). \tag{2.3.1}$$

Here $a_t \in \mathcal{A}$ *is the action from all agents in the state* s_t *at* $0 \leq t \leq T$, l_t *is the learning rate of* Q *table, and* r_t *is the observed reward sampled from taking action* a_t. *Suppose that agents in both states (*L *and* H*) will choose actions according to an* ϵ*-greedy policy. Namely, in each iteration* t, *each agent in state* s *(*$s = L$ *or* $s = H$*) will choose an action from* $\arg\max_{a\in\mathcal{A}} Q_t(s,a)$ *with probability* $1-\epsilon$ *and choose an arbitrary action with probability* ϵ. *Then* μ_t *evolves according to Equation (2.2.10) with any initial population distribution* μ_0 *under this* ϵ*-greedy policy.*

For simplicity, the Q-function is initialized to be zero for every $s \in \mathcal{S}, a \in \mathcal{A}$. Following this Q-leaning update (2.3.1), the experiment result on the convergence of Q-function is reported below, with $T = 10000$, $p = 0.6$, $\lambda_L = 0.5$, $\lambda_H = 0.8$, $\lambda = 10$, $\gamma = 0.5$. Following

[55], the learning rate is set as $l_t = \frac{1}{\#(s_t,a_t)+1}$ with $\#(s_t, a_t)$ the number of total visits to state-action pair (s_t, a_t) up to iteration t.

Table 2.1: Convergence of Q-function with different initial distribution, following the Q-leaning update (2.3.1). Due to the incorrect form of the Q-function, Q table will converge to different values under different initial population distribution μ_0.

Initial distribution	$Q_T(L, \mathrm{ST})$	$Q_T(L, \mathrm{MV})$	$Q_T(H, \mathrm{ST})$	$Q_T(H, \mathrm{MV})$
$p_0 = 0.01$	-4.41	-4.41	-3.24	-3.58
$p_0 = 0.5$	-4.56	-4.36	-3.45	-3.45
$p_0 = 0.99$	-4.87	-4.69	-3.78	-3.78

Note the time inconsistency here: with different initial population distribution μ_0, Q table will converge to different values. The culprit: with this form of Q-function, the state space and the action space are not rich enough to ensure the DPP or the Bellman equation for (2.3.1).

2.3.2 IQ-function for MFCs with learning

Example 2.3.1 indicates the wrong form of the Q-function for MFCs with learning. Therefore, our first step is to define an appropriate Q-function. The question is, what is wrong with the previous one?

First, recall that MFC as a cooperative game is essentially an auxiliary control problem: instead of maximizing reward for each individual agent, the objective in MFC is to maximize the collective reward from the perspective of the auxiliary controller. The auxiliary controller's value function depends on the probability distribution of the state μ. Therefore, the Q-function for MFCs should be dependent on μ instead of s.

Secondly, Lemma 2.2.1 suggests that once a control $\pi \in \Pi$ is given, the dynamics of the state distribution is determined by $\mu_{t+1} = \Phi(\mu_t, h_t)$, which is a *deterministic process* through h_t in $\mathcal{P}(\mathcal{S})$. Therefore, an appropriate Q-function should be a function on $\mathcal{H}$, rather than of the single action in $\mathcal{A}$ or a probability distribution on $\mathcal{A}$. In other words, the learning problem for MFCs should be recast as control problems with the probability measure space as the new state-action space such that

$$v(\mu) := \sup_{\pi\in\Pi} \sum_{t=0}^{\infty} \gamma^t \mathbb{E}\big[\widehat{r}(\mu_t, \pi_t(\mu_t))\big], \tag{2.3.2}$$

subject to

$$\mu_{t+1} = \Phi(\mu_t, \pi_t(\mu_t)),\ t \in \mathbb{N} \cup \{0\}, \quad \mu_0 = \mu. \tag{2.3.3}$$

Accordingly, the appropriate Q-function for MFCs with learning should be defined by "lifting" the classical Q-function in RL, with lifting in the sense of replacing the state space

$\mathcal{S}$ and action space $\mathcal{A}$ by the state space $\mathcal{P}(\mathcal{S})$ and action space $\mathcal{H}$ respectively. Hence, the proper Q-function for MFCs with learning should take the following *integral* form, called, integrated Q-function (IQ).

Definition 2.3.2 (IQ-function) *Given the framework of MFC with learning in (2.2.6)-(2.2.8), the IQ-function is a measurable real-valued function defined on $\mathcal{P}(\mathcal{S}) \times \mathcal{H}$ such that*

$$Q(\mu, h) \;=\; \sup_{\pi \in \Pi_1} Q^{\pi}(\mu, h), \quad \textit{for any } \mu \in \mathcal{P}(\mathcal{S}),\ h \in \mathcal{H}, \tag{2.3.4}$$

$$\textit{with}\quad Q^{\pi}(\mu, h) \;=\; \mathbb{E}^{\pi}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, \mu_t, a_t, \nu_t) \,\middle|\, s_0 \sim \mu, a_0 \sim h(s_0), a_t \sim \pi_t(s_t, \mu_t), t \in \mathbb{N}\right].$$

2.3.3 DPP: Necessary for IQ-function

The above specification of the IQ-function enables us to establish the DPP for MFCs with learning, in the form of the following Bellman equation.

Theorem 2.3.3 *(DPP for IQ-function) Under Outstanding Assumption (A), for any $\mu \in \mathcal{P}(\mathcal{S})$ and $h \in \mathcal{H}$,*

$$Q(\mu, h) \;=\; \widehat{r}(\mu, h) + \gamma \sup_{h' \in \mathcal{H}} Q(\Phi(\mu, h), h'). \tag{2.3.5}$$

The idea for the proof of Theorem 2.3.3 is borrowed from Theorem 3.1 in [138]. Unlike [138], which considers the value function for MFC with strict controls, we consider the IQ-function for MFC with learning over an infinite time horizon with a stochastic reward function and with relaxed controls. For completeness, we highlight the key step for the proof of Theorem 2.3.3.

Proof of Theorem 2.3.3 To start, fix some arbitrary $\mu \in \mathcal{P}(\mathcal{S})$ and $\pi \in \Pi$, we have

$$\begin{aligned} v^{\pi}(\mu) \;&=\; \widehat{r}(\mu, \pi_0(\mu)) + \sum_{t=1}^{\infty} \gamma^t \widehat{r}(\mu_t, \pi_t(\mu_t)) \\ &=\; \widehat{r}(\mu, \pi_0(\mu)) + \gamma v^{\pi_-}(\Phi(\mu, \pi_0(\mu))) \\ &=\; Q^{\pi_-}(\mu, \pi_0(\mu)), \end{aligned} \tag{2.3.6}$$

where $\pi_- := \{\pi_t\}_{t=1}^{\infty} \in \Pi_1$, and the second equality uses the flow property of $\{\mu_t\}_{t=0}^{\infty}$ from Lemma 2.2.1.

Now we can establish the following relation between the value function v and the IQ-function,

$$v(\mu) \;=\; \sup_{h \in \mathcal{H}} Q(\mu, h), \quad \text{for any } \mu \in \mathcal{P}(\mathcal{S}). \tag{2.3.7}$$

To prove (2.3.7), we first show $v(\mu) \leq \sup_{h\in\mathcal{H}} Q(\mu, h)$ for any $\mu \in \mathcal{P}(\mathcal{S})$. To see this, note

$$v^{\pi}(\mu) \;=\; Q^{\pi^-}(\mu, \pi_0(\mu)) \;\leq\; Q(\mu, \pi_0(\mu)) \;\leq\; \sup_{h\in\mathcal{H}} Q(\mu, h), \tag{2.3.8}$$

where the first inequality is by definition of $Q(\mu, h)$, and by the fact that $\pi_0(\mu) \in \mathcal{H}$ for each $\mu \in \mathcal{H}$. Taking supremum over all policies $\pi \in \Pi$ in (2.3.8) shows that

$$v(\mu) \;\leq\; \sup_{h\in\mathcal{H}} Q(\mu, h). \tag{2.3.9}$$

To see for any $\mu \in \mathcal{P}(\mathcal{S})$, $v(\mu) \geq \sup_{h\in\mathcal{H}} Q(\mu, h)$, fix any arbitrary $\mu \in \mathcal{P}(\mathcal{S})$ and $\pi_0(\mu) \in \mathcal{H}$, for any $\epsilon > 0$, there exists $\pi^\epsilon = \{\pi_t^\epsilon\}_{t=1}^\infty \in \Pi_1$ such that

$$Q^{\pi^\epsilon}(\mu, \pi_0(\mu)) \geq Q(\mu, \pi_0(\mu)) - \epsilon. \tag{2.3.10}$$

Now define $\tilde{\pi} = \{\tilde{\pi}_t\}_{t=0}^\infty \in \Pi$ by $\tilde{\pi}_t = \pi_0 1_{\{t=0\}} + \pi_t^\epsilon 1_{\{t\in\mathbb{N}\}}$, then from (2.3.6) and (2.3.10),

$$v(\mu) \;\geq\; v^{\tilde{\pi}}(\mu) \;=\; Q^{\pi^\epsilon}(\mu, \pi_0(\mu)) \geq Q(\mu, \pi_0(\mu)) - \epsilon. \tag{2.3.11}$$

Taking supremum over all π_0 in (2.3.11), we obtain

$$v(\mu) \;\geq\; \sup_{\pi_0} Q(\mu, \pi_0(\mu)) - \epsilon \;=\; \sup_{h\in\mathcal{H}} Q(\mu, h) - \epsilon.$$

Since the above inequality holds for any $\epsilon > 0$,

$$v(\mu) \;\geq\; \sup_{h\in\mathcal{H}} Q(\mu, h). \tag{2.3.12}$$

(2.3.7) follows from (2.3.9) and (2.3.12).
Now we are ready to prove (2.3.5).

$$\begin{aligned}
Q(\mu, h) &= \sup_{\pi\in\Pi_1} \mathbb{E}^{\pi}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, \mu_t, a_t, \nu_t) \middle| s_0 \sim \mu, a_0 \sim h(s_0), a_t \sim \pi_t(s_t, \mu_t), t \in \mathbb{N}\right] \\
&= \sup_{\pi\in\Pi_1} \left[\widehat{r}(\mu, h) + \gamma v^{\pi}(\Phi(\mu, h))\right] \\
&= \widehat{r}(\mu, h) + \gamma v(\Phi(\mu, h)) \\
&= \widehat{r}(\mu, h) + \gamma \sup_{h'\in\mathcal{H}} Q(\Phi(\mu, h), h'),
\end{aligned}$$

where the second equality is from the flow property of $\{\mu_t\}_{t=0}^\infty$ in Lemma 2.2.1, the third equality is by the definition of the value function, and the last inequality is from (2.3.7). □

2.3.4 DPP: Sufficient for IQ-function

So far, we have established the necessary condition for the Bellman equation. That is, the IQ-function satisfies the Bellman equation and is time consistent. We can further establish that this Bellman equation is sufficient, in the form of the following verification theorem.

Theorem 2.3.4 *(Verification theorem)*

(1). Suppose $\tilde{Q} : \mathcal{P}(\mathcal{S}) \times \mathcal{H} \to \mathbb{R}$ *satisfies the Bellman equation* (2.3.5) *for any* $(\mu, h) \in \mathcal{P}(\mathcal{S}) \times \mathcal{H}$. *Suppose that for every* $\mu \in \mathcal{P}(\mathcal{S})$, *one can also find a stationary control* $\pi^*(\mu) \in \mathcal{H}$ *that achieves* $\sup_{h\in\mathcal{H}} \tilde{Q}(\mu, h)$, *then* π^* *is an optimal stationary control of problem* (2.2.6)-(2.2.8).

(2). If we further assume that there exists $0 \le r_{max} < \infty$ *such that the sup norm* $||r||_\infty \le r_{max}, a.s.$, *then* Q *defined in* (2.3.4) *is the unique solution in* $\{q \in \mathcal{M}(\mathcal{P}(\mathcal{S}) \times \mathcal{H}) : \|q\|_\infty \le V_{\max}\}$ *for the Bellman equation* (2.3.5), *with* $V_{\max} := \frac{r_{max}}{1-\gamma}$. *In this case, the stationary control* $\pi^*(\mu) \in \mathcal{H}$ *that achieves* $\sup_{h\in\mathcal{H}} \tilde{Q}(\mu, h)$ *is an optimal stationary control of problem* (2.2.6)-(2.2.8).

The idea for the proof of Theorem 2.3.4-(1) is borrowed from the proof for Theorem 3.2 in [138]. Nevertheless, the backward induction argument by [138] for a finite-time-horizon case needs appropriate modification for the IQ-function over an infinite time horizon. We hence highlight the key step here.

Proof of Theorem 2.3.4 (1) On one hand, given any $\mu \in \mathcal{P}(\mathcal{S})$, for any given control $\pi \in \Pi$, the evolution of $\{\mu_t\}_{t=0}^\infty$ is given by (2.2.10). From (2.3.5)

$$\tilde{Q}(\mu_t, \pi_t(\mu_t)) \;\ge\; \widehat{r}(\mu_t, \pi_t(\mu_t)) + \gamma \tilde{Q}(\mu_{t+1}, \pi_{t+1}(\mu_{t+1})), \quad t \in \mathbb{N} \cup \{0\}. \tag{2.3.13}$$

Multiplying (2.3.13) by γ^t and summing over $0 \le t \le T-1$ for any fixed T, we obtain

$$\tilde{Q}(\mu, \pi_0(\mu)) - \gamma^T \tilde{Q}(\mu_T, \pi_T(\mu_T)) \;\ge\; \sum_{t=0}^{T-1} \gamma^t \widehat{r}(\mu_t, \pi_t(\mu_t)).$$

As $\lim_{T\to\infty} \gamma^T \tilde{Q}(\mu, h) = 0$ for any fixed $(\mu, h) \in \mathcal{P}(\mathcal{S}) \times \mathcal{H}$, by taking the limit $T \to \infty$, $\tilde{Q}(\mu, \pi_0(\mu)) \ge \sum_{t=0}^\infty \gamma^t \widehat{r}(\mu_t, \pi_t(\mu_t)) = v^\pi(\mu)$, which leads to $\sup_{h\in\mathcal{H}} \tilde{Q}(\mu, h) \ge v(\mu)$.

On the other hand, since $\pi^*(\mu) \in \arg\max \tilde{Q}(\mu, h)$ holds for every $\mu \in \mathcal{P}(\mathcal{S})$, then

$$\tilde{Q}(\mu_t, \pi^*(\mu_t)) \;=\; \widehat{r}(\mu_t, \pi^*(\mu_t)) + \gamma \tilde{Q}(\mu_{t+1}, \pi^*(\mu_{t+1})).$$

Repeat the same argument for π^* as for π, $\sup_{h\in\mathcal{H}} \tilde{Q}(\mu, h) = \tilde{Q}(\mu, \pi^*(\mu)) = v^{\pi^*}(\mu)$, which shows that π^* is an optimal stationary control.

(2) First, since $||r||_\infty \le r_{\max}$ a.s., for any $\mu \in \mathcal{P}(\mathcal{S})$ and $h \in \mathcal{H}$, the aggregated reward function (2.2.13) satisfies

$$|\widehat{r}(\mu, h)| \le r_{\max} \cdot \int_{s\in\mathcal{S}} \mu(ds) \int_{a\in\mathcal{A}} h(s)(da) = r_{\max}.$$

In this case, for any $\mu \in \mathcal{P}(\mathcal{S})$ and $h \in \mathcal{H}$, $|Q(\mu, h)| \le r_{\max} \cdot \sum_{t=0}^{\infty} \gamma^t = V_{\max}$. Hence, $Q \in \{q \in \mathcal{M}(\mathcal{P}(\mathcal{S}) \times \mathcal{H}) : \|q\|_\infty \le V_{\max}\}$ and it satisfies the Bellman equation (2.3.5).

To see that Q is the unique function in $\{q \in \mathcal{M}(\mathcal{P}(\mathcal{S}) \times \mathcal{H}) : \|q\|_\infty \le V_{\max}\}$ satisfying (2.3.5), consider the Bellman operator $B : \mathcal{M}(\mathcal{P}(\mathcal{S}) \times \mathcal{H}) \to \mathcal{M}(\mathcal{P}(\mathcal{S}) \times \mathcal{H})$ defined by

$$(B\, q)(\mu, h) = \widehat{r}(\mu, h) + \gamma \sup_{\tilde{h}\in\mathcal{H}} q(\Phi(\mu, h), \tilde{h}). \tag{2.3.14}$$

Then B is a contraction operator on $\{q \in \mathcal{M}(\mathcal{P}(\mathcal{S}) \times \mathcal{H}) : \|q\|_\infty \le V_{\max}\}$: clearly B maps $\{q \in \mathcal{M}(\mathcal{P}(\mathcal{S}) \times \mathcal{H}) : \|q\|_\infty \le V_{\max}\}$ to itself, and for any $(\mu, h) \in \mathcal{P}(\mathcal{S}) \times \mathcal{H}$,

$$|Bq_1(\mu, h) - Bq_2(\mu, h)| \le \gamma \sup_{\tilde{h}\in\mathcal{H}} |q_1(\Phi(\mu, h), \tilde{h}) - q_2(\Phi(\mu, h), \tilde{h})| \le \gamma \|q_1 - q_2\|_\infty.$$

Thus, $\|Bq_1 - Bq_2\|_\infty \le \gamma\|q_1 - q_2\|_\infty$. Therefore, B is a contraction mapping with modulus $\gamma < 1$ under the sup norm on $\{q \in \mathcal{M}(\mathcal{P}(\mathcal{S}) \times \mathcal{H}) : \|q\|_\infty \le V_{\max}\}$. Hence the uniqueness by the contraction property. □

2.3.5 IQ-function vs classical Q-function.

Comparing IQ-function and the classical Q-function, there is an analytical connection between their respective Bellman equations.

To see this, consider the simplest problem of MFCs with learning where there are no state distribution nor action distribution in the probability transition function P or in the deterministic reward function r. Assume $\mathcal{S}$ and $\mathcal{A}$ are finite so that there exists $r_{\max} > 0$ such that $\|r\|_\infty \le r_{\max}$. Here for clarity, we shall distinguish the classical single-agent Q-function (2.2.3) and the IQ-function (2.3.4) by writing Q_{single} and Q_{mfc} respectively. Then we see that the IQ-function in (2.3.4) is the integral of Q-function in (2.2.3) such that

$$Q_{\text{mfc}}(\mu, h) \;=\; \sum_{s\in\mathcal{S}} \mu(s) \sum_{a\in\mathcal{A}} Q_{\text{single}}(s, a) h(s)(a). \tag{2.3.15}$$

To see this connection, define

$$\tilde{Q}(\mu, h) = \sum_{s\in\mathcal{S}} \mu(s) \sum_{a\in\mathcal{A}} Q_{\text{single}}(s, a) h(s)(a).$$

Note that $\tilde{Q}$ is linear in μ and h. From the Bellman equation (2.2.4) of Q_{single} (2.2.3), we have

$$\tilde{Q}(\mu, h) \;=\; \widehat{r}(\mu, h) + \gamma \sum_{s\in\mathcal{S}} \mu(s) \sum_{a\in\mathcal{A}} h(s)(a) \sum_{s'\in\mathcal{S}} P(s, a)(s') \max_{a'\in\mathcal{A}} Q_{\text{single}}(s', a'),$$

then we can see that

$$\sum_{s\in\mathcal{S}}\mu(s)\sum_{a\in\mathcal{A}}h(s)(a)\sum_{s'\in\mathcal{S}}P(s,a)(s')\max_{a'\in\mathcal{A}}Q_{\text{single}}(s',a') \;=\; \sup_{h'\in\mathcal{H}}\tilde{Q}(\Phi(\mu,h),h').$$

In fact, on one hand, for any $h'\in\mathcal{H}$,

$$\begin{aligned}
&\sum_{s\in\mathcal{S}}\mu(s)\sum_{a\in\mathcal{A}}h(s)(a)\sum_{s'\in\mathcal{S}}P(s,a)(s')\max_{a'\in\mathcal{A}}Q_{\text{single}}(s',a')\\
=\;&\sum_{s\in\mathcal{S}}\mu(s)\sum_{a\in\mathcal{A}}h(s)(a)\sum_{s'\in\mathcal{S}}P(s,a)(s')\sum_{\tilde{a}\in\mathcal{A}}h'(s')(\tilde{a})\max_{a'\in\mathcal{A}}Q_{\text{single}}(s',a')\\
\geq\;&\sum_{s\in\mathcal{S}}\mu(s)\sum_{a\in\mathcal{A}}h(s)(a)\sum_{s'\in\mathcal{S}}P(s,a)(s')\sum_{\tilde{a}\in\mathcal{A}}h'(s')(\tilde{a})Q_{\text{single}}(s',\tilde{a})\\
=\;&\sum_{s'\in\mathcal{S}}\Phi(\mu,h)(s')\sum_{\tilde{a}\in\mathcal{A}}h'(s')(\tilde{a})Q_{\text{single}}(s',\tilde{a})\\
=\;&\tilde{Q}(\Phi(\mu,h),h'),
\end{aligned}$$

where the first equality is from $\sum_{\tilde{a}\in\mathcal{A}}h(s')(\tilde{a})=1$, the second equality is by (2.2.11), and the last equality is by the definition of $\tilde{Q}$.
On the other hand, if we take

$$h'_*(s') \;=\; \begin{cases}\frac{1}{\#\arg\max_{a'\in\mathcal{A}}Q_{\text{single}}(s',a')}, & \text{if } a_*(s')\in\arg\max_{a'\in\mathcal{A}}Q_{\text{single}}(s',a'),\\ 0, & \text{otherwise,}\end{cases}$$

with $\#\arg\max_{a'\in\mathcal{A}}Q_{\text{single}}(s',a')$ the number of elements in $\arg\max_{a'\in\mathcal{A}}Q_{\text{single}}(s',a')$, then

$$\begin{aligned}
&\sup_{h'\in\mathcal{H}}\tilde{Q}(\Phi(\mu,h),h')\geq\tilde{Q}(\Phi(\mu,h),h'_*)\\
=\;&\sum_{s'\in\mathcal{S}}\Phi(\mu,h)(s')\sum_{a'\in\mathcal{A}}Q_{\text{single}}(s',a')h'_*(s')(a')\\
=\;&\sum_{s'\in\mathcal{S}}\Phi(\mu,h)(s')\max_{a'\in\mathcal{A}}Q_{\text{single}}(s',a')\\
=\;&\sum_{s\in\mathcal{S}}\mu(s)\sum_{a\in\mathcal{A}}h(s)(a)\sum_{s'\in\mathcal{S}}P(s,a)(s')\max_{a'\in\mathcal{A}}Q_{\text{single}}(s',a').
\end{aligned}$$

Therefore,

$$\tilde{Q}(\mu,h) \;=\; \widehat{r}(\mu,h)+\gamma\sup_{h'\in\mathcal{H}}\tilde{Q}(\Phi(\mu,h),h').$$

Since both $\tilde{Q}$ and Q_{mfc} satisfy Bellman equations (2.3.5), we have $Q_{\text{mfc}}=\tilde{Q}$ from the uniqueness of the fixed point of a contraction mapping B in (2.3.14).

Remark 2.3.5 *The relationship between Q_{mfc} and Q_{single} in (2.3.15) is intriguing for algorithmic designs: the "global" Q table (Q_{mfc}) needs to be trained in a centralized manner by observing the population state distribution; yet agents only need to maintain a "local" Q table (Q_{single}) for execution.*

2.3.6 DPP for the Value Function and Value-iteration Algorithms

For model-based learning algorithms such as the value iteration, we have the Bellman equation for the value function v from Theorem 2.3.3.

Theorem 2.3.6 *(DPP for value function) Under Outstanding Assumption (A), the value function v satisfies the Bellman equation*

$$v(\mu) \;=\; \sup_{h\in\mathcal{H}}\big\{\widehat{r}(\mu,h)+\gamma v(\Phi(\mu,h))\big\}, \quad \textit{for any } \mu\in\mathcal{P}(\mathcal{S}). \tag{2.3.16}$$

Given $\pi:\mathcal{P}(\mathcal{S})\to\mathcal{H}$, define the operator $T_\pi:\mathcal{M}(\mathcal{P}(\mathcal{S}))\to\mathcal{M}(\mathcal{P}(\mathcal{S}))$ such that

$$(T_\pi w)(\mu) \;:=\; \widehat{r}(\mu,\pi(\mu))+\gamma w(\Phi(\mu,\pi(\mu))), \tag{2.3.17}$$

and another operator $T:\mathcal{M}(\mathcal{P}(\mathcal{S}))\to\mathcal{M}(\mathcal{P}(\mathcal{S}))$ such that

$$(Tw)(\mu) \;:=\; \sup_{h\in\mathcal{H}}\big\{\widehat{r}(\mu,h)+\gamma w(\Phi(\mu,h))\big\}, \tag{2.3.18}$$

where $\widehat{r}(\mu,h)$ and $\Phi(\mu,h)$ are given in (2.2.13) and (2.2.11).

Proposition 2.3.7 *Assume without loss of generality $v_0=0$, then under Outstanding Assumption (A), we have for all $\mu\in\mathcal{P}(\mathcal{S})$,*

$$v(\mu)=\lim_{n\to\infty}(T^n v_0)(\mu),$$

where T^n is n-th composition of T such that such that $T^n=\underbrace{T\circ\cdots\circ T}_{n}$.

Proof of Proposition 2.3.7 relies on the following Lemma.

Lemma 2.3.8 *Assume Outstanding Assumption (A) and without loss of generality $v_0=0$, for any $\mu\in\mathcal{P}(\mathcal{S})$ and $\pi=\{\pi_t\}_{t=0}^n$ with $\pi_t:\mathcal{P}(\mathcal{S})\to\mathcal{H}$ for every $0\le t\le n$,*

$$(T_{\pi_0}\cdots T_{\pi_n}v_0)(\mu) \;=\; \sum_{t=0}^{n}\gamma^t\widehat{r}(\mu_t,\pi_t(\mu_t)), \tag{2.3.19}$$

$$(T^{n+1}v_0)(\mu) \;=\; \sup_{\{\pi_t\}_{t=0}^n}(T_{\pi_0}\cdots T_{\pi_n}v_0)(\mu), \tag{2.3.20}$$

where $T_{\pi_0}\cdots T_{\pi_n}$ is the composition of all T_{π_t}, $0\le t\le n$.

Proof of Lemma 2.3.8 We prove (2.3.20) (and similarly (2.3.19)) by the forward induction. The result clearly holds for $n=0$ as

$$\sup_{\pi_0}(T_{\pi_0}v_0)(\mu)=\sup_{\pi_0}\widehat{r}(\mu,\pi_0(\mu))=\sup_{h\in\mathcal{H}}\mathbb{E}[\widehat{r}(\mu,h)]=(Tv_0)(\mu).$$

Suppose that (2.3.20) holds for $n = k$, then for $n = k + 1$,

$$\begin{aligned}
(T^{k+1}v_0)(\mu) &= \sup_{h\in\mathcal{H}} \left\{\widehat{r}(\mu,h) + \gamma(T^k v_0)(\Phi(\mu,h))\right\} \\
&= \sup_{h\in\mathcal{H}} \left\{\widehat{r}(\mu,h) + \gamma \sup_{\{\tilde{\pi}_t\}_{t=0}^{k-1}} (T_{\tilde{\pi}_0}\cdots T_{\tilde{\pi}_{k-1}}v_0)(\Phi(\mu,h))\right\} \\
&= \sup_{h\in\mathcal{H}} \left\{\widehat{r}(\mu,h) + \gamma \sup_{\{\pi_t\}_{t=1}^{k}} (T_{\pi_1}\cdots T_{\pi_k}v_0)(\Phi(\mu,h))\right\} \\
&= \sup_{h\in\mathcal{H},\{\pi_t\}_{t=1}^{k}} \left\{\widehat{r}(\mu,h) + \gamma(T_{\pi_1}\cdots T_{\pi_k}v_0)(\Phi(\mu,h))\right\} \\
&= \sup_{\{\pi_t\}_{t=0}^{k+1}} (T_{\pi_0}\cdots T_{\pi_k}v_0)(\mu),
\end{aligned}$$

where the first equality is from the definition of T in (2.3.18); the second equality is by the assumption that (2.3.20) holds for $n = k$. □

Proof of Proposition 2.3.7 Rewrite $v^\pi(\mu)$ as

$$\begin{aligned}
v^\pi(\mu) &= \sum_{t=0}^{n-1} \gamma^t \widehat{r}(\mu_t, \pi_t(\mu_t)) + \sum_{t=n}^{\infty} \gamma^t \widehat{r}(\mu_t, \pi_t(\mu_t)) \\
&= (T_{\pi_0}\cdots T_{\pi_{n-1}}v_0)(\mu) + \sum_{t=n}^{\infty} \gamma^t \widehat{r}(\mu_t, \pi_t(\mu_t)),
\end{aligned} \tag{2.3.21}$$

where the second equality is by (2.3.19). Now Outstanding Assumption (A) implies

$$\lim_{n\to\infty} \sup_{\pi} \sum_{t=n}^{\infty} \gamma^t |\widehat{r}(\mu_t, \pi_t(\mu_t))| = 0.$$

Taking supremum over $\pi \in \Pi$ in (2.3.21) together with (2.3.20) gives

$$\begin{aligned}
v(\mu) &\le (T^n v_0)(\mu) + \sup_{\pi} \sum_{t=n}^{\infty} \gamma^t |\widehat{r}(\mu_t, \pi_t(\mu_t))|, \\
v(\mu) &\ge (T^n v_0)(\mu) - \sup_{\pi} \sum_{t=n}^{\infty} \gamma^t |\widehat{r}(\mu_t, \pi_t(\mu_t))|.
\end{aligned}$$

Taking the limit as $n \to \infty$ together with (2.3.20) yields $v(\mu) = \lim_{n\to\infty}(T^n v_0)(\mu)$. □

2.4 Example: Consistency of DPP

Example 2.4.1 *Take a two-state dynamic system with two choices of actions. The state space* $\mathcal{S} = \{L, H\}$ *and the action space* $\mathcal{A} = \{\mathrm{ST}, \mathrm{MV}\}$. *The transition probability goes as follows:*

$$P(s,a)(s') = \lambda_s \mathbf{1}_{\{a=\mathrm{MV}\}}, \quad \text{if } s' \neq s \in \mathcal{S}, \; P(s,a)(s') = 1 - \lambda_s \mathbf{1}_{\{a=\mathrm{MV}\}}, \quad \text{if } s' = s \in \mathcal{S},$$

with $\lambda_s \in [0,1]$ *for* $s \in \mathcal{S}$. *Here* $P(s,a)(s')$ *is the probability of moving to state* s' *when the agent in state* s *takes the action* a; *when the agent in the state* s *takes action* ST, *she will stay at the current state* s; *when the agent in the state* s *takes the action* MV, *she will move to a different state* s' *with probability* $0 \leq \lambda_s \leq 1$, $s \in \mathcal{S}$ *and stay at state* s *with probability* $1-\lambda_s$, $s \in \mathcal{S}$. *After each action, the representative agent will receive a reward* $r_t = \mathbf{1}_{\{s_t=H\}} - \big(\mathbb{E}[\mathbf{1}_{\{s_t=H\}}]\big)^2 - \lambda \mathcal{W}_1(\mu_t, B)$. *Here* μ_t *denotes the probability distribution of the state at time* t, B *is a given Binomial distribution with parameter* p $(1-\lambda_L \leq p \leq \lambda_H)$, *and* $\lambda > 0$ *is a scalar parameter. Fix any arbitrary initial state distribution* $\mu_0 = p_0 1_{\{s_0=L\}} + (1-p_0) 1_{\{s_0=H\}}$ *for some* $0 \leq p_0 \leq 1$.
Note that the expected value of immediate reward $\mathbb{E}[r_t]$ *at each time* t *is*

$$\mathbb{E}[r_t] = \mathbb{E}[\mathbf{1}_{\{s_t=H\}}] - \mathbb{E}[\mathbf{1}_{\{s_t=H\}}]^2 - \lambda \mathcal{W}_1(\mu_t, B) = \mu_t(H) - \mu_t(H)^2 - 2\lambda |\mu_t(H) - (1-p)|,$$

where $\mu_t(L)$ *and* $\mu_t(H)$ *are the population distribution on state* L *and* H *at time* t, *respectively. Suppose that* $\lambda > 0$ *is large enough, we have* $\max_\pi \big(\mathbb{E}[\mathbf{1}_{\{s_t=H\}}] - \mathbb{E}[\mathbf{1}_{\{s_t=H\}}]^2 - \lambda \mathcal{W}_1(\mu_t, B)\big) = 1 - p - (1-p)^2$ *when* $\mu_t = B$ *for any* $t \in \mathbb{N}$. *Therefore, the value function is optimal if and only if the population distribution* $\{\mu_t^*\}_{t=1}^\infty$ *corresponding to the optimal control* π^* *is given by*

$$\mu_t^* \;=\; B = p\mathbf{1}_{\{s=L\}} + (1-p)\mathbf{1}_{\{s=H\}}, \quad t \in \mathbb{N}, \qquad \mu_0 \;=\; p_0 \mathbf{1}_{\{s=L\}} + (1-p_0)\mathbf{1}_{\{s=H\}}.$$

From the flow property of $\{\mu_t^*\}_{t=1}^\infty$ *in* (2.2.10) *and* (2.2.11), *we get*

$$\begin{aligned} \mu_1^*(L) = p \;&=\; \Phi(\mu_0, \pi^*)(L) \;=\; \sum_{s\in\mathcal{S}} \mu_0(s) \sum_{a\in\mathcal{A}} P(s,a)(L)\pi^*(s)(a), \\ \mu_{t+1}^*(L) = p \;&=\; \Phi(\mu_t^*, \pi^*)(L) \;=\; \sum_{s\in\mathcal{S}} \mu_t^*(s) \sum_{a\in\mathcal{A}} P(s,a)(L)\pi^*(s)(a), \; t \in \mathbb{N}, \end{aligned}$$

which gives the optimal control and the optimal value

$$\pi^*(L) = (1 - \tfrac{1-p}{\lambda_L})\mathbf{1}_{\{a=\mathrm{ST}\}} + \tfrac{1-p}{\lambda_L}\mathbf{1}_{\{a=\mathrm{MV}\}}, \tag{2.4.1}$$

$$\pi^*(H) = (1 - \tfrac{p}{\lambda_H})\mathbf{1}_{\{a=\mathrm{ST}\}} + \tfrac{p}{\lambda_H}\mathbf{1}_{\{a=\mathrm{MV}\}}, \tag{2.4.2}$$

$$v^{\pi^*}(\mu_0) = 1 - p_0 - (1-p_0)^2 - 2\lambda|p_0 - p| + \tfrac{\gamma}{1-\gamma}\Big(1 - p - (1-p)^2\Big). \tag{2.4.3}$$

Now, the Q-learning update at each iteration t *using the IQ-function is*

$$Q_{t+1}(\mu, h) \;=\; Q_t(\mu, h) + l_t \times \left(\hat{r}_t + \gamma \sup_{h' \in \mathcal{H}} Q_t(\Phi(\mu, h), h') - Q_t(\mu, h)\right). \tag{2.4.4}$$

Here l_t *is the learning rate at iteration* t *and* γ *is the discount factor.*

This example is further studied by [129] later on.

Next, we design a simple algorithm (Algorithm 1) to show the performance of the IQ update (2.4.4), with the following specifications. We emphasize that the focus here is the time consistency property not the efficiency of the algorithm. In the experiment, we shall use element $(p, 1-p)$ in the Euclidean space $\mathbb{R}^2$ to denote the Binomial distribution with parameter p.

(a) **Dimension reduction**: Since $\mu_t(L) + \mu_t(H) = 1$ $(t = 0, 1, \cdots, T)$, $\pi(L, \mathrm{ST}) + \pi(L, \mathrm{ST}) = 1$ and $\pi(H, \mathrm{ST}) + \pi(H, \mathrm{MV}) = 1$ for any distribution μ_t and control π, we can reduce the dimension of the IQ-function. If we define $Q(\mu^L, \pi^L_{\mathrm{ST}}, \pi^H_{\mathrm{ST}})$ and $\Phi(\mu^L, \pi^L_{\mathrm{ST}}, \pi^H_{\mathrm{ST}})$, with $\mu^L := \mu(L)$ the probability of population state L, $\pi^L_{\mathrm{ST}} := \pi(L, \mathrm{ST})$ the probability of the action to "stay" at state L, and $\pi^H_{\mathrm{ST}} := \pi(H, \mathrm{ST})$ the probability of the action to "stay" at state H, then $Q(\mu, \pi) = Q(\mu^L, \pi^L_{\mathrm{ST}}, \pi^H_{\mathrm{ST}})$, $\Phi(\mu, \pi) = \Phi(\mu^L, \pi^L_{\mathrm{ST}}, \pi^H_{\mathrm{ST}})$ with a slight abuse of notation.

(b) **Distribution discretization**: To examine the time-consistency property of (2.3.5), we discretize the state and action distribution with finite precision and apply the classical Q-learning update to (2.4.6) with finite-dimensional inputs. For simplicity, we assume uniform discretization such that $\tilde{\mathcal{P}}(\mathcal{A}) := \{i/N_a : 0 \le i \le N_a\}$ and $\tilde{\mathcal{P}}(\mathcal{S}) := \{i/N_s : 0 \le i \le N_s\}$ for some constant integers $N_a > 0$ and $N_s > 0$. (For more refined discretization other than the uniform one, see for example the ϵ-Net approach in [72]).

(c) **Algorithmic design**: The algorithm is summarized in Algorithm 1. Note that (2.4.6) is the reduced form of the original update (2.4.4) with a discretized distribution. In order to perform the for-loop (Step 3, 4, and 5) in Algorithm 1, we assume the accessibility to a population simulator $(\mu', \widehat{r}) = \mathcal{G}(\mu, \pi)$. That is, for any pair $(\mu, \pi) \in \mathcal{P}(\mathcal{S}) \times \mathcal{H}$, we can sample the aggregated population reward $\widehat{r}$ and the next population state distribution μ' under control π.

(d) **Metric design**: Explicit calculations show that the stationary optimal control is given by (2.4.1). Therefore, we design the following metric to check the convergence of the Q table to the true value $v^{\pi^*}(\mu_0)$ in (2.4.3) and the speed of the convergence.

$$E(t) = \frac{1}{N_s} \sum_{i=0}^{N_s} \left| Q_t \left(\frac{i}{N_s}, \mathrm{Proj}(\pi^{L,*}_{\mathrm{ST}}, \tilde{\mathcal{P}}(\mathcal{A})), \mathrm{Proj}(\pi^{H,*}_{\mathrm{ST}}, \tilde{\mathcal{P}}(\mathcal{A})) \right) - v^{\pi^*} \left(\frac{i}{N_s}, 1 - \frac{i}{N_s} \right) \right|.$$

Here for simplicity we take $N_s = N_a$; $\pi^{s,*}_{\mathrm{ST}}$, $s \in \mathcal{S}$, is the optimal control π^* in (2.4.1) evaluated in state s and action ST; the projection is defined as $\mathrm{Proj}(\pi^{s,*}_{\mathrm{ST}}, \tilde{\mathcal{P}}(\mathcal{A})) := \arg\min_{\tilde{\pi}^s_{\mathrm{ST}} \in \tilde{\mathcal{P}}(\mathcal{A})} |\pi^{s,*}_{\mathrm{ST}} - \tilde{\pi}^s_{\mathrm{ST}}|$.

(e) **Parameter set-up**: Parameters are set as follows: $T = 20$, $p = 0.6$, a constant learning rate $l_t = l = 0.4$ for all t, $\gamma = 0.5$, $\lambda_L = 0.5$, $\lambda_R = 0.8$, $\lambda = 10$ and $N_a =$

$N_s = 20$. Each component in Q_0 is randomly initialized from a uniform distribution on $[0, 1]$. The experiments are repeated 20 times.

(f) **Performance analysis.** The experiments show that metric $E(t)$ converges in around 15 outer iterations (Figure 2.1). The standard deviation of 20 repeated experiments is very small. This is partially due to lifting of the state-action space which leads to the deterministic property of the underlying system.

Recall $\tilde{\mathcal{P}}(\mathcal{S}) = \{i/N_s : 0 \le i \le N_s\}$. Further denote the projection as

$$\text{Proj}(\Phi(\mu^L, \pi^L_{\text{ST}}, \pi^H_{\text{ST}})(L), \tilde{\mathcal{P}}(\mathcal{S})) := \arg\min_{\tilde{\mu}^L \in \tilde{P}(\mathcal{S})} |\Phi(\mu^L, \pi^L_{\text{ST}}, \pi^H_{\text{ST}})(L) - \tilde{\mu}^L|. \tag{2.4.5}$$

Then the algorithm is summarized as follows.

Algorithm 1 MFCs Q-learning with distribution discretization

Input: N_a and N_s.
Initialization: $Q_0(\mu^L, \pi^L_{\text{ST}}, \pi^H_{\text{ST}}) = 0$ for every $(\mu^L, \pi^L_{\text{ST}}, \pi^H_{\text{ST}}) \in \tilde{\mathcal{P}}(\mathcal{S}) \times (\tilde{\mathcal{P}}(\mathcal{A}))^2$.
for $t = 0, 1, \cdots, T-1$ **do**
  for $\pi^L_{\text{ST}} \in \{\frac{i}{N_a}, 0 \le i \le N_a\}$ **do**
    for $\pi^H_{\text{ST}} \in \{\frac{i}{N_a}, 0 \le i \le N_a\}$ **do**
      for $\mu^L \in \{\frac{i}{N_s}, 0 \le i \le N_s\}$ **do**
        $\mu^{L\prime} = \text{Proj}(\Phi(\mu^L, \pi^L_{\text{ST}}, \pi^H_{\text{ST}})(L), \tilde{\mathcal{P}}(\mathcal{S}))$

$$Q_{t+1}(\mu^L, \pi^L_{\text{ST}}, \pi^H_{\text{ST}}) = (1 - l_t) Q_t(\mu^L, \pi^L_{\text{ST}}, \pi^H_{\text{ST}}) + l_t \times \left(\widehat{r}_t + \gamma \max_{(\pi^{L\prime}_{\text{ST}}, \pi^{H\prime}_{\text{ST}}) \in (\tilde{\mathcal{P}}(\mathcal{A}))^2} Q_t(\mu^{L\prime}, \pi^{L\prime}_{\text{ST}}, \pi^{H\prime}_{\text{ST}})\right), \tag{2.4.6}$$

      end for
    end for
  end for
end for

Remark 2.4.2 *In general, distribution discretization is sample inefficient and suffers from the curse of dimensionality. For example, in Example 2.3.1, there are two states and two actions, with* $N_s = N_a = 20$ *with precision* 0.05. *The Q-function is a table of dimension* 8000. *This complexity grows exponentially with the number of states and actions. Moreover, although* $E(t)$ *converges relatively fast, there is unavoidable errors due to truncation, as seen in Figure 2.2. The optimal value* $Q_t\left(\frac{i}{N_s}, Proj(\pi^{L,*}_{\text{ST}}, \tilde{\mathcal{P}}(\mathcal{A})), Proj(\pi^{H,*}_{\text{ST}}, \tilde{\mathcal{P}}(\mathcal{A}))\right)$ *can not be distinguished from its surrounding areas, where the areas with the lightest color all correspond to the largest value. This is because the accuracy is only up to* 0.05 *in each iteration. Therefore,*

it is desirable to develop sample-efficient and accurate Q-learning algorithms for MFCs with learning with the correct Bellman equation (2.3.5). See Chapter 3 for such a development with kernel regression method applied to improve the sample efficiency.

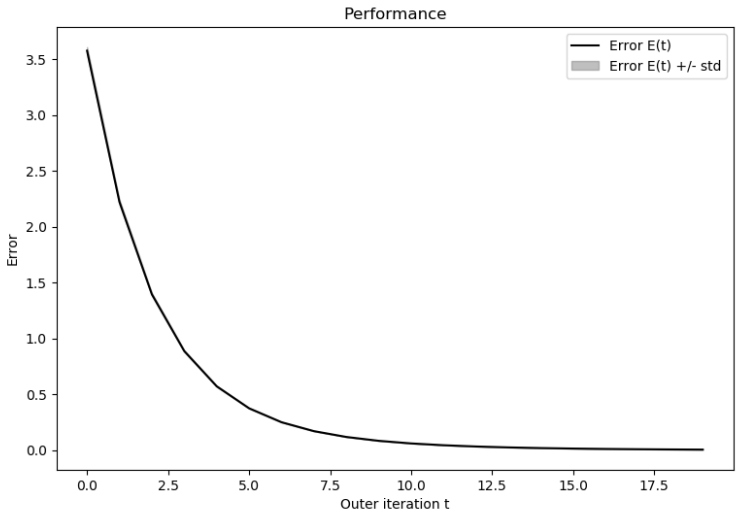
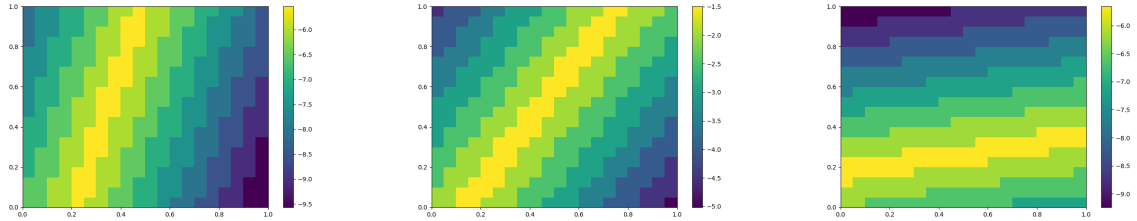


Figure 2.1: Numerical performance of Algorithm 1 in Example 2.3.1. The plot shows that the metric $E(t)$ converges in around 15 outer iterations.



(a) $Q_T(0.3, \cdot, \cdot)$ (b) $Q_T(0.5, \cdot, \cdot)$. (c) $Q_T(0.9, \cdot, \cdot)$

Figure 2.2: Snapshots of the IQ tables in Example 2.3.1, output by Algorithm 1 at the final iteration T.

Remark 2.4.1 (Comparison with time-inconsistency in Section 2.3.1) *Algorithm 1 in Example 2.3.1 is designed based on the classical single-agent DPP with the usual state and action spaces $\mathcal{S}$ and $\mathcal{A}$. This approach fails both theoretically and empirically in the mean-field regime. From a thoery point, the classical single-agent DPP is not "rich enough"*

to include all the necessary information. From an empirical perspective, the epsilon-greedy method and the time-dependent learning rate enables visiting each (s, a) pair sufficiently many times, yet without the convergence guarantee.

In contrast, Algorithm 1 finds the value of $Q(\mu, h)$ for any possible initial distribution μ including μ_0 used in Example 2.3.1. In addition, the convergence of the entire Q table in Figure 2.1 implies the convergence of $Q(\mu_0, \cdot)$ by the definition of $E(t)$.

2.5 Example: Equilibrium Pricing

Consider a continuum of firms that supply a homogeneous product. For a representative firm, the sell price follows

$$s_t = s_{t-1} + \kappa(d_t - \mathbb{E}[a_t]) + w_t, \tag{2.5.1}$$

where d_t is the (normalized) exogenous demand process per individual, a_t is stochastic representing the supply volume from the representative firm, $\{w_t\}_{t=1}^{\infty}$ are IID noise following some distribution such as the symmetric random walk, $\mathbb{E}[a_t] := \int_{\mathcal{A}} a\nu_t(d\,a)$ is the aggregated supply volume from all firms where ν_t is the action distribution of all firms, $\kappa > 0$ is a scalar amplifying the impact from the supply-demand imbalance on the price process. Namely, the price process of the product will have a positive drift when demand is bigger than the supply whereas the price process will experience a negative drift if the average supply exceeds the demand. Correspondingly, the per-period reward accruing to the representative firm with supply volume a_t is

$$r_t = (s_t - c)a_t,$$

with $c > 0$ the production cost.

Model set-up. Assume $d_t \sim \mathcal{N}(2, 0.25)$, $c = 1$, $a_t \in \mathcal{A} := \{0, 1, 2, \cdots, 4\}$, and $\kappa = 1$. To enable Q-learning based algorithms, we truncate the values of the price dynamics within $s_t \in \mathcal{S} := \{0, 1, 2, \cdots, 19\}$. Set the discount rate as $\gamma = 0.6$ in the objective function. Finally, we consider $\{w_t\}_{t=1}^{\infty}$ follow IID random walks with probability $1/2$ being 1 and with probability $1/2$ being -1.

Design of the IQ table and the algorithm. Recall that s_t defined in (2.5.1) is the selling price received by the representative agent who produce a_t amount of products during period t. Given the sets of actions and states specified above and from a population perspective, $\mu_t(i)$ denotes the proportion of firms who received price $i - 1$ and $\nu_t(i)$ denotes the proportion of firms taking action i (i.e., supply with amount $i - 1$) at time t. In addition, denote $\hat{\mathcal{P}}(\mathcal{S}) := \big\{\mu \in \mathcal{P}(\mathcal{S}) \text{ such that } \mu(s) \in \{j/N_s; j = 0, 1, \cdot, N_s\} \text{ for } s \in \mathcal{S}\big\}$ and $\hat{\mathcal{H}} := \big\{h \in \mathcal{H} \text{ such that } h(s; a) \in \{j/N_a; j = 0, 1, \cdot, N_a\}, \forall a \in \mathcal{A} \text{ and } s \in \mathcal{S}\big\}$ as the discretized probability measure spaces for the states and local policies, respectively. Therefore,

it is enough to consider the IQ table with the format of $Q(\mu, h)$ such that $\mu \in \hat{\mathcal{P}}(\mathcal{S})$ and $h \in \hat{H}$. Recall the projection defined in (2.4.5),

$$\text{Proj}(\Phi(\mu^0, h), \hat{\mathcal{P}}(\mathcal{S})) := \arg\min_{\hat{\mu} \in \hat{\mathcal{P}}(\mathcal{S})} |\Phi(\mu^0, h) - \hat{\mu}|.$$

We use this projection function to maintain the feasibility of the state distribution throughout training. See Algorithm 2 for the detailed design of the learning algorithm, where we set $T = 100$, $N_a = 20$, $N_s = 20$ and a constant learning rate $l_t = l = 0.1$.

Algorithm 2 MFCs Q-learning for the Supply Game

Input: N_a.
for $t = 1, \cdots, T$ **do**
  for $h \in \hat{\mathcal{H}}$ **do**
    for $\mu \in \hat{\mathcal{P}}(\mathcal{S})$ **do**
      $\mu' = \text{Proj}(\Phi^0(\mu, h), \hat{\mathcal{P}}(\mathcal{S}))$

$$Q_{t+1}(\mu, h) = (1 - l_t) Q_t(\mu, h) + l_t \times \Big(\hat{r}_t + \gamma \max_{h' \in \hat{\mathcal{H}}} Q_t(\mu', h') \Big), \tag{2.5.2}$$

    end for
  end for
end for

Results. The IQ table converges with error less than 0.01 within 60 outer iterations (see Figure 2.3).

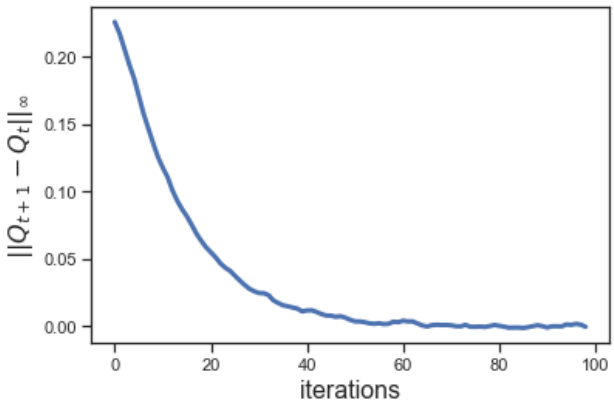


Figure 2.3: Numerical performance of Algorithm 2 in the Supply Game example. The plot shows that the learned IQ-function converges in around 60 outer iterations.

state (price s)	0	1	2	3	4	5	6	7	8	9
MFC solution	0.4	0.65	0.9	0.8	1.15	0.8	1.25	0.9	0.95	1.4
state (price s)	10	11	12	13	14	15	16	17	18	19
MFC solution	1.8	2.05	2.15	1.9	2.3	3.	2.85	2.35	3.15	3.1

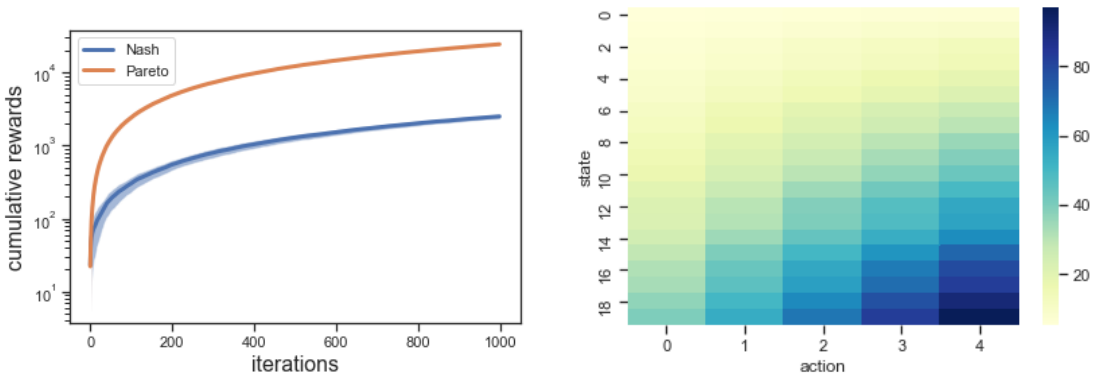
Table 2.2: Optimal aggregated supply volume from all firms $\mathbb{E}[a^*(s)]$ in the MFC solution, given different initial price.

Table 2.2 shows the average supply from the learned MKV solution given different initial price. When the price is small ($s_t = 0$), it is optimal for the agents to provide a small amount of supplies to reduce the cost. When the price is in the middle range ($s_t = 10$), it is optimal to suggest the allocation such that $\mathbb{E}[a_t] \approx 2$ with no impact on the price. When the price is high ($s_t = 19$), the price impact is tolerable by providing an excessive supply since it is highly profitable in this situation.

Comparison with Nash equilibrium. We also compare the performance of MFC solution under the Pareto optimality criterion with that of the mean-field game solution (MFG) under the Nash equilibrium criterion. The algorithm for learning the MFG solution is from [72]. The output of the MFG strategy follows a Boltzmann type of policy $\pi(s)(a) \sim \exp(\beta Q(s, a))$ with a temperature parameter $\beta > 0$. Here we take $\beta = 1$ and train the algorithm until the error falls below 10^{-2}.

The trained Q table is provided in Figure 2.4b, which indicates that in the equilibrium agents provide the largest supply (i.e., action 5) with a high probability. This is also consistent with the mean-field information provided in Table 2.3. In the mean-field equilibrium, the expected supply is always bigger than $\mathbb{E}[d_t] = 2$. This implies that, in a competitive market, agents are more aggressive in making and selling productions.

In Figure 2.4a, we compare the cumulative rewards under the trained MFC policy with the trained MFG policy for 1000 rounds. We observe that the cumulative rewards from the MFC policy is ten times bigger than those from the MFG policy. This implies that the aggressive behavior due to competition may leads to inefficiency from the market perspective, which indicates the necessity of understanding Pareto optimal solution for large-scale decision making problems.



(a) Cumulative rewards for 1000 rounds.

(b) Q table of the MFG solution.

Figure 2.4: Comparison between the MFG solution and the MFC solution in the Supply Game example. Figure 2.4a compares the cumulative rewards of the learned MFC policy, with the cumulative rewards of the learned MFG policy in 1000 rounds. The cumulative rewards from the MFC policy is ten times bigger than those from the MFG policy. The MFG Q table is provided in Figure 2.4b, which indicates that in the equilibrium agents provide the largest supply (i.e., action 5) with a high probability.

state (price s)	0	1	2	3	4	5	6	7	8	9
MFG solution	2.08	2.19	2.37	2.37	2.51	2.59	2.75	2.81	2.92	3.01
state (price s)	10	11	12	13	14	15	16	17	18	19
MFG solution	3.08	3.22	3.32	3.34	3.42	3.51	3.56	3.60	3.65	3.68

Table 2.3: Optimal aggregated supply volume from all firms $\mathbb{E}[a^*(s)]$ in the MFG solution, given different initial price.

Chapter 3

Q-Learning for Cooperative Mean-Field MARL

Multi-agent reinforcement learning (MARL), despite its popularity and empirical success, suffers from the curse of dimensionality. This chapter builds the mathematical framework to approximate cooperative MARL by a mean-field control (MFC) approach, and shows that the approximation error is of $\mathcal{O}(\frac{1}{\sqrt{N}})$. Based on the dynamic programming principle for both the value function and the Q-function of learning MFC (Chapter 2), it proposes a model-free kernel-based Q-learning algorithm (MFC-K-Q), which is shown to have a linear convergence rate for the MFC problem, the first of its kind in the MARL literature. It further establishes that the convergence rate and the sample complexity of MFC-K-Q are independent of the number of agents N, which provides an $\mathcal{O}(\frac{1}{\sqrt{N}})$ approximation to the MARL problem with N agents in the learning environment. Empirical studies for the network traffic congestion problem demonstrate that MFC-K-Q outperforms existing MARL algorithms when N is large, for instance when $N > 50$.

3.1 Motivation and Related Works

Multi-agent reinforcement learning (MARL) has enjoyed substantial successes for analyzing the otherwise challenging games, including two-agent or two-team computer games [155, 176], self-driving vehicles [154], real-time bidding games [87], ride-sharing [100], and traffic routing [54]. Despite its empirical success, MARL suffers from the curse of dimensionality known also as the *combinatorial nature* of MARL: its sample complexity by existing algorithms for stochastic dynamics grows exponentially with respect to the number of agents N. (See [75] and also Proposition 3.2.1 in Section 3.2). In practice, this N can be on the scale of thousands or more, for instance, in rider match-up for Uber-pool and network routing for Zoom.

One classical approach to tackle this curse of dimensionality is to focus on *local policies*, namely by exploiting special structures of MARL problems and by designing problem-

dependent algorithms to reduce the complexity. For instance, [98] developed value-based distributed Q-learning algorithm for deterministic and finite Markov decision problems (MDPs), and [142] exploited special dependence structures among agents. (See the reviews by [196] and [205] and the references therein).

Another approach is to consider MARL in the regime with a large number of homogeneous agents. In this paradigm, by functional strong law of large numbers (a.k.a. propagation of chaos) [88, 119, 169, 64], non-cooperative MARLs can be approximated under Nash equilibrium by mean-field games with learning, and cooperative MARLs can be studied under Pareto optimality by analyzing mean-field controls (MFC) with learning. This approach is appealing not only because the dimension of MFC or MFG is independent of the number of agents N, but also because solutions of MFC/MFG (without learning) have been shown to provide good approximations to the corresponding N-agent game in terms of both game values and optimal strategies [79, 96, 129, 147, 149].

MFG with learning has gained popularity in the reinforcement learning (RL) community [59, 72, 82, 195, 199], with its sample complexity shown to be similar to that of single-agent RL [59, 72]. Yet MFC with learning is by and large an uncharted field despite its potentially wide range of applications [100, 104, 180, 187]. The main challenge for MFC with learning is to deal with probability measure space over the state-action space, which is shown in Chapter 2 to be the minimal space for which the Dynamic Programming Principle will hold. One of the open problems for MFC with learning is therefore, as pointed out in [129], to design efficient RL algorithms on probability measure space.

To circumvent designing algorithms on probability measure space, [36] proposed to add common noises to the underlying dynamics. This approach enables them to apply the standard RL theory for stochastic dynamics. Their model-free algorithm, however, suffers from high sample complexity as illustrated in Table 3.1 below, and with weak performance as demonstrated in Section 3.7. For special classes of linear-quadratic MFCs with stochastic dynamics, [35] explored the policy gradient method and [113] developed an actor-critic type algorithm.

Contributions. This chapter builds the mathematical framework to approximate cooperative MARL by MFCs with learning. The approximation error is shown to be of $\mathcal{O}(\frac{1}{\sqrt{N}})$. It then identifies the minimum space on which the Dynamic Programming Principle holds, and proposes an efficient approximation algorithm (MFC-K-Q) for MFC with learning. This model-free Q-learning-based algorithm combines the technique of kernel regression with approximated Bellman operator. The convergence rate and the sample complexity of this algorithm are shown to be independent of the number of agents N, and rely only on the size of the state-action space of the underlying single-agent dynamics (Table 3.1). As far as we are aware of, there is no prior algorithm with linear convergence rate for cooperative MARL.

Mathematically, the DPP is established through lifting the state-action space and by aggregating the reward and the underlying dynamics. This lifting idea has been used in previous MFC framework ([138, 189] without learning and Chapter 2 with learning). Our

work finds that this lifting idea is critical for efficient algorithm design for MFC with learning: the resulting deterministic dynamics from this lifting trivialize the choice of the learning rate for the convergence analysis and significantly reduce the sample complexity.

Our experiment in Section 3.7 demonstrates that MFC-K-Q avoids the curse of dimensionality and outperforms both existing MARL algorithms (when $N > 50$) and the MFC algorithm in [36]. Table 3.1 summarizes the complexity of our MFC-K-Q algorithm along with these relevant algorithms.

Work	MFC/N-agent	Method	Sample Complexity Guarantee
Our work	MFC	Q-learning	$\Omega(T_{cov} \cdot \log(1/\delta))$
[36]	MFC	Q-learning	$\Omega((T_{cov} \cdot \log(1/\delta))^l \cdot \text{poly}(\log(1/(\delta\epsilon))/\epsilon))$
Vanilla N-agent	N-agent	Q-learning	$\Omega(\text{poly}((\lvert\mathcal{S}\rvert\lvert\mathcal{A}\rvert)^N \cdot \log(1/(\delta\epsilon)) \cdot N/\epsilon))$
[142]	N-agent	Actor-critic	$\Omega(\text{poly}((\lvert\mathcal{S}\rvert\lvert\mathcal{A}\rvert)^{f(\log(1/\epsilon))} \cdot \log(1/\delta) \cdot N/\epsilon))$

Table 3.1: Comparison of the sample complexity of MFC-K-Q algorithm with these relevant algorithms.

T_{cov} in Table 3.1 is the covering time of the exploration policy and $l = \max\{3 + 1/\kappa, 1/(1-\kappa)\} > 4$ for some $\kappa \in (0.5, 1)$. Other parameters are as in Proposition 3.2.1 and also in Theorem 3.5.6. Note that [142] assumed that agents interact locally through a given graph so that local policies can approximate the global one, yet $f(\log(1/\epsilon))$ can scale as N for a dense graph.

Organizations. Section 3.2 introduces the set-up of cooperative MARL and MFC with learning. Section 3.3 establishes the Dynamical Programming Principle for MFC with learning. Section 3.4 proposes the algorithm (MFC-K-Q) for MFC with learning, with convergence and sample complexity analysis. Section 3.5 is dedicated to the proof of the main theorem. Section 3.6 connects cooperative MARL and MFC with learning. Section 3.7 tests performance of MFC-K-Q in a network congestion control problem. Finally, some future directions and discussions are provided in Section 3.9. For ease of exposition, proofs for all lemmas are in the Appendix.

Notation. For a measurable space $(\mathcal{S}, \mathcal{B})$, where $\mathcal{B}$ is σ-algebra on $\mathcal{S}$, denote $\mathbb{R}^{\mathcal{S}}$ for the set of all real-valued measurable functions on $\mathcal{S}$, $\mathbb{R}^{\mathcal{S}} := \{f : \mathcal{S} \to \mathbb{R} | f \text{ is measurable}\}$. For each bounded $f \in \mathbb{R}^{\mathcal{S}}$, define the sup norm of f as $||f||_\infty = \sup_{s\in\mathcal{S}} |f(s)|$. In addition, when $\mathcal{S}$ is finite, we denote $|\mathcal{S}|$ for the size of $\mathcal{S}$, and $\mathcal{P}(\mathcal{S})$ for the set of all probability measures on $\mathcal{S}$: $\{p : p(s) \geq 0, \sum_{s\in\mathcal{S}} p(s) = 1\}$, which is equivalent to the probability simplex in $\mathbb{R}^{|\mathcal{S}|}$. Moreover, in $\mathcal{P}(\mathcal{S})$, let $d_{\mathcal{P}(\mathcal{S})}$ be the metric induced by the l_1 norm: for any $u, v \in \mathcal{P}(\mathcal{S})$, $d_{\mathcal{P}(\mathcal{S})}(u, v) = \sum_{s\in\mathcal{S}} |u(s) - v(s)|$. $\mathcal{P}(\mathcal{S})$ is endowed with Borel σ-algebra induced by l_1 norm.

$1(x \in A)$ denotes the indicator function, i.e., $1(x \in A) = 1$ if $x \in A$, and $1(x \notin A) = 0$ if $x \notin A$.

Notation	Definition
$\mathbb{R}^{\mathcal{X}}$	set of real-valued measurable functions on measurable space $\mathcal{X}$
$\mathcal{P}(\mathcal{X})$	set of all probability measures on $\mathcal{X}$
$d_{\mathcal{P}(\mathcal{X})}$	metric induced by l_1 norm: $d_{\mathcal{P}(\mathcal{X})}(u,v) = \sum_{s\in\mathcal{X}} \lvert u(s) - v(s)\rvert$
γ	discount factor
$1(x \in A)$	indicator function of event $\{x \in A\}$
N	number of agents
$\mathcal{S}$	state space of single agent
$\mathcal{X}$	action space of single agent
$\mu_t^N \in \mathcal{P}(\mathcal{S})$	empirical state distribution of N agents at time t
$\nu_t^N \in \mathcal{P}(\mathcal{A})$	empirical action distribution of N agents at time t
$\mu_t \in \mathcal{P}(\mathcal{S})$	state distribution of the MFC problem at time t
$\nu_t \in \mathcal{P}(\mathcal{A})$	action distribution of of the MFC problem at time t
$\mathcal{H}$	$\mathcal{H} := \{h : \mathcal{S} \to \mathcal{P}(\mathcal{A})\}$ is the set of local policies
$\mathcal{C}$	$\mathcal{C} := \mathcal{P}(\mathcal{S}) \times \mathcal{H}$, the product space of $\mathcal{P}(\mathcal{S})$ and $\mathcal{H}$
Π	$\Pi := \{\pi = \{\pi_t\}_{t=0}^{\infty} \,\vert\, \pi_t : \mathcal{P}(\mathcal{S}) \to \mathcal{H}\}$, set of admissible policies
$\tilde{r}(s,\mu,a,\nu(\mu,h))$	individual reward
R	bound of the reward, i.e., $\lvert\tilde{r}\rvert < R$
$r(\mu,h)$	aggregated population reward in (3.2.6)
L_P	Lipschitz constant for transition matrix P
$L_{\tilde{r}}$	Lipschitz constant for reward $\tilde{r}$
$L_r := \tilde{R} + 2L_{\tilde{r}}$	Lipschitz constant for r
$L_\Phi := 2L_P + 1$	Lipschitz constant for Φ
$\mathcal{C}_\epsilon$	ϵ-net on $\mathcal{C}$
N_ϵ	size of the ϵ-net $\mathcal{C}_\epsilon$ on $\mathcal{C}$
$N_{\mathcal{H}_\epsilon}$	size of the ϵ-net $\mathcal{H}_\epsilon$ on $\mathcal{H}$
$K(c^i, c)$	weighted kernel function with $c^i \in \mathcal{C}_\epsilon$ and $c \in \mathcal{C}$
L_K	Lipschitz constant for kernel K
N_K	at most N_K number of $c^i \in \mathcal{C}_\epsilon$ satisfies $K(c, c^i) > 0$
Γ_K	kernel regression operator from $\mathbb{R}^{\mathcal{C}_\epsilon} \to \mathbb{R}^{\mathcal{C}}$
$T_{\mathcal{C},\pi}$	covering time of the ϵ-net under policy π
M_ϵ	constant appearing in Assumption 3.5.1

Table 3.2: Summary of mathematical notations in Chapter 3.

3.2 MARL and MFC with Learning

3.2.1 MARL and its Complexity

We first recall cooperative MARL in an infinite time horizon, where there are N agents whose game strategies are coordinated by a central controller. Let us assume the state space $\mathcal{S}$ and the action space $\mathcal{A}$ are all finite.

At each step $t = 0, 1, \cdots$, the state of agent j $(= 1, 2, \cdots, N)$ is $s_t^{j,N} \in \mathcal{S}$ and she takes an action $a_t^{j,N} \in \mathcal{A}$. Given the current state profile $\boldsymbol{s}_t = (s_t^{1,N}, \cdots, s_t^{N,N}) \in \mathcal{S}^N$ and the current action profile $\boldsymbol{a}_t = (a_t^{1,N}, \cdots, a_t^{N,N}) \in \mathcal{A}^N$ of N-agents, agent j will receive a reward $\tilde{r}^j(\boldsymbol{s}_t, \boldsymbol{a}_t)$ and her state will change to $s_{t+1}^{j,N}$ according to a transition probability function $P^j(\boldsymbol{s}_t, \boldsymbol{a}_t)$. A Markovian game further restricts the admissible policy for agent j to be of the form $a_t^{j,N} \sim \pi_t^j(\boldsymbol{s}_t)$. That is, $\pi_t^j : \mathcal{S}^N \to \mathcal{P}(\mathcal{A})$ maps each state profile $\boldsymbol{s} \in \mathcal{S}^N$ to a randomized action, with $\mathcal{P}(\mathcal{A})$ the probability measure space on space $\mathcal{A}$.

In this cooperative MARL, the central controller is to maximize the expected discounted aggregated accumulated rewards over all policies and averaged over all agents. That is to find

$$\sup_{\boldsymbol{\pi}} \frac{1}{N} \sum_{j=1}^{N} v^j(\boldsymbol{s}, \boldsymbol{\pi}), \text{ where } v^j(\boldsymbol{s}, \boldsymbol{\pi}) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \tilde{r}^j(\boldsymbol{s}_t, \boldsymbol{a}_t) \big| \boldsymbol{s}_0 = \boldsymbol{s} \right]$$

is the accumulated reward for agent j, given the initial state profile $\boldsymbol{s}_0 = \boldsymbol{s}$ and policy $\boldsymbol{\pi} = \{\boldsymbol{\pi}_t\}_{t=0}^{\infty}$ with $\boldsymbol{\pi}_t = (\pi_t^1, \ldots, \pi_t^N)$. Here $\gamma \in (0, 1)$ is a discount factor, $a_t^{j,N} \sim \pi_t^j(\boldsymbol{s}_t)$, and $s_{t+1}^{j,N} \sim P^j(\boldsymbol{s}_t, \boldsymbol{a}_t)$.

The sample complexity of the Q learning algorithm of this cooperative MARL is exponential with respect to N. Indeed, take Theorem 4 in [55] and note that the corresponding covering time for the policy of the central controller will be at least $(|\mathcal{S}||\mathcal{A}|)^N$, then we see

Proposition 3.2.1 *Let $|\mathcal{S}|$ and $|\mathcal{A}|$ be respectively the size of the state space $\mathcal{S}$ and the action space $\mathcal{A}$. Let Q^* and Q_T be respectively the optimal value and the value of the asynchronous Q-learning algorithm in [55] using polynomial learning rate with time $T = \Omega\left(poly\left((|\mathcal{S}||\mathcal{A}|)^N \cdot \frac{N}{\epsilon} \cdot \ln(\frac{1}{\delta\epsilon}) \right) \right)$. Then with probability at least $1 - \delta$, $\|Q_T - Q^*\|_\infty \leq \epsilon$.*

This exponential growth in sample complexity makes the algorithm difficult to scale up. The classical approach for this curse of dimensionality is to explore special network structures (e.g., sparsity or local interactions among agents) for MARL problems. Here we shall propose an alternative approach in the regime when there is a large number of homogeneous agents.

3.2.2 MFC with Learning: Set-up, Assumptions and Some Preliminary Results

To overcome the curse of dimensionality in N, we now propose a mean-field control (MFC) framework to approximate this cooperative MARL when agents are homogeneous.

In this MFC framework, all agents are assumed to be identical, indistinguishable, and interchangeable, and each agent $j(= 1, \cdots, N)$ is assumed to depend on all other agents only through the empirical distribution of their states and actions. That is, denote $\mathcal{P}(\mathcal{S})$ and $\mathcal{P}(\mathcal{A})$ as the probability measure spaces over the state space $\mathcal{S}$ and the action space $\mathcal{A}$, respectively. The empirical distribution of the states is $\mu_t^N(s) = \frac{\sum_{j=1}^N 1(s_t^{j,N}=s)}{N} \in \mathcal{P}(\mathcal{S})$, and the empirical distribution of the actions is $\nu_t^N(a) = \frac{\sum_{j=1}^N 1(a_t^{j,N}=a)}{N} \in \mathcal{P}(\mathcal{A})$. Then, by law of large numbers, this coperative MARL becomes an MFC with learning when $N \to \infty$. Moreover, as all agents are indistinguishable, one can focus on a single representative agent.

Mathematically, this MFC with learning is as follows. At each time $t = 0, 1, \cdots$, the representative agent in state s_t takes an action $a_t \in \mathcal{A}$ according to the admissible policy $\pi_t(s_t, \mu_t) : \mathcal{S} \times \mathcal{P}(\mathcal{S}) \to \mathcal{P}(\mathcal{A})$ assigned by the central controller, who can observe the population state distribution $\mu_t \in \mathcal{P}(\mathcal{S})$. Further denote $\Pi := \{\pi = \{\pi_t\}_{t=0}^\infty | \pi_t : \mathcal{S} \times \mathcal{P}(\mathcal{S}) \to \mathcal{P}(\mathcal{A}) \text{ is measurable}\}$ as the set of admissible policies. The agent will then receive a reward $\tilde{r}(s_t, \mu_t, a_t, \nu_t)$ and move to the next state $s_{t+1} \in \mathcal{S}$ according to a probability transition function $P(s_t, \mu_t, a_t, \nu_t)$. Here P and $\tilde{r}$ rely on the state distribution μ_t and the action distribution $\nu_t(\cdot) := \sum_{s\in\mathcal{S}} \pi_t(s, \mu_t)(\cdot)\mu_t(s)$, and are possibly unknown.

The objective for this MFC with learning is to find v the maximal expected discounted accumulated reward over all admissible policies $\pi = \{\pi_t\}_{t=0}^\infty$, namely

$$v(\mu) = \sup_{\pi\in\Pi} v^\pi(\mu) := \sup_{\pi\in\Pi} \mathbb{E}\left[\sum_{t=0}^\infty \gamma^t \tilde{r}(s_t, \mu_t, a_t, \nu_t) \,\middle|\, s_0 \sim \mu\right], \tag{MFC}$$
$$\text{subject to} \quad s_{t+1} \sim P(s_t, \mu_t, a_t, \nu_t), \quad a_t \sim \pi_t(s_t, \mu_t).$$

with initial condition $\mu_0 = \mu$.

Note that after observing μ_t, the policy from the central controller $\pi_t(\cdot, \mu_t)$ can be viewed as a mapping from $\mathcal{S}$ to $\mathcal{P}(\mathcal{A})$. In this case, we set

$$h_t(\cdot) := \pi_t(\cdot, \mu_t) \tag{3.2.1}$$

for notation simplicity and denote $\mathcal{H} := \{h : \mathcal{S} \to \mathcal{P}(\mathcal{A})\}$ as the space for $h_t(\cdot)$. Note that $\mathcal{H}$ is isomorphic to the product of $|\mathcal{S}|$ copies of $\mathcal{P}(\mathcal{A})$. Therefore, the set of admissible policies Π can be rewritten as

$$\Pi := \left\{\pi = \{\pi_t\}_{t=0}^\infty \,|\, \pi_t : \mathcal{P}(\mathcal{S}) \to \mathcal{H} \text{ is measurable}\right\}. \tag{3.2.2}$$

This reformulation of the admissible policy set is key for deriving the Dynamic Programming Principle (DPP) of (MFC): it enables us to show that the objective in (MFC) is law-invariant

and the probability distribution of the dynamics in (MFC) satisfies flow property. This flow property is also crucial for establishing the convergence of the associated cooperative MARL by (MFC).

Lemma 3.2.2 *Under any admissible policy $\pi = \{\pi_t\}_{t=0}^{\infty} \in \Pi$, and the initial state distribution $s_0 \sim \mu_0 = \mu$, the evolution of the state distribution $\{\mu_t\}_{t\geq 0}$, is given by*

$$\mu_{t+1} \quad = \quad \Phi(\mu_t, h_t), \tag{3.2.3}$$

where $h_t(\cdot)$ is defined in (3.2.1) and the dynamics Φ is defined as

$$\Phi(\mu, h) \quad := \quad \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} P(s, \mu, a, \nu(\mu, h))\mu(s)h(s)(a) \in \mathcal{P}(\mathcal{S}), \tag{3.2.4}$$

for any $(\mu, h) \in \mathcal{P}(\mathcal{S}) \times \mathcal{H}$ and $\nu(\mu, h)(\cdot) := \sum_{s\in\mathcal{S}} h(s)(\cdot)\mu(s) \in \mathcal{P}(\mathcal{A})$. Moreover, the value function v^{π} defined in (MFC) can be rewritten as

$$v^{\pi}(\mu) \quad = \quad \sum_{t=0}^{\infty} \gamma^t r(\mu_t, h_t), \tag{3.2.5}$$

where for any $(\mu, h) \in \mathcal{P}(\mathcal{S}) \times \mathcal{H}$, the reward r is defined as

$$r(\mu, h) \quad := \quad \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} \tilde{r}(s, \mu, a, \nu(\mu, h))\mu(s)h(s)(a). \tag{3.2.6}$$

Remark 3.2.3 *Because of the aggregated forms of Φ and r from (3.2.4) and (3.2.6), they are also called the aggregated dynamics and the aggregated reward, respectively.*

We start with some standard regularity assumptions for MFC problems [33]. These assumptions are necessary for the mean-field approximation to cooperative MARL and for the subsequent convergence and sample complexity analysis of the learning algorithm.

Let us use the l_1 distance for the metrics $d_{\mathcal{P}(\mathcal{S})}$ and $d_{\mathcal{P}(\mathcal{A})}$ of $\mathcal{P}(\mathcal{S})$ and $\mathcal{P}(\mathcal{A})$, and define $d_{\mathcal{H}}(h_1, h_2) = \max_{s\in\mathcal{S}} ||h_1(s) - h_2(s)||_1$ and $d_{\mathcal{C}}((\mu_1, h_1), (\mu_2, h_2)) = ||\mu_1 - \mu_2||_1 + d_{\mathcal{H}}(h_1, h_2)$ for the space $\mathcal{H}$ and $\mathcal{C} := \mathcal{P}(\mathcal{S}) \times \mathcal{H}$, respectively. Moreover, we endow $\mathcal{C}$ with Borel σ algebra generated by open sets in $d_{\mathcal{C}}$.

Assumption 3.2.4 (Continuity and boundedness of $\tilde{r}$) *There exist $\tilde{R} > 0, L_{\tilde{r}} > 0$, such that for all $s \in \mathcal{S}, a \in \mathcal{A}$, $\mu_1, \mu_2 \in \mathcal{P}(\mathcal{S}), \nu_1, \nu_2 \in \mathcal{P}(\mathcal{A})$,*

$$|\tilde{r}(s, \mu_1, a, \nu_1)| \leq \tilde{R}, \ |\tilde{r}(s, \mu_1, a, \nu_1) - \tilde{r}(s, \mu_2, a, \nu_2)| \leq L_{\tilde{r}} \cdot (||\mu_1 - \mu_2||_1 + ||\nu_1 - \nu_2||_1).$$

Assumption 3.2.5 (Continuity of P) *There exists $L_P > 0$ such that for all $s \in \mathcal{S}, a \in \mathcal{A}, \mu_1, \mu_2 \in \mathcal{P}(\mathcal{S}), \nu_1, \nu_2 \in \mathcal{P}(\mathcal{A})$,*

$$||P(s, \mu_1, a, \nu_1) - P(s, \mu_2, a, \nu_2)||_1 \leq L_P \cdot (||\mu_1 - \mu_2||_1 + ||\nu_1 - \nu_2||_1).$$

Note that l_1 distance between transition kernels $P(s, \mu, a, \nu)$ in Assumption 3.2.5 is equivalent to 1-Wasserstein distance when $\mathcal{S}$ and $\mathcal{A}$ are equipped with discrete metrics $1(s_1 \neq s_2)$ for $s_1, s_2 \in \mathcal{S}$ and $1(a_1 \neq a_2)$ for $a_1, a_2 \in \mathcal{A}$, respectively, see e.g., [67], [76]. Under Assumptions 3.2.4 and 3.2.5, it is clear that the probability measure ν over the action space, the aggregated reward r in (3.2.6), and the aggregated dynamics Φ in (3.2.4) are all Lipschitz continuous, which will be useful for subsequent analysis.

Lemma 3.2.6 (Continuity of ν)

$$\|\nu(\mu, h) - \nu(\mu', h')\|_1 \leq d_{\mathcal{C}}((\mu, h), (\mu', h')). \tag{3.2.7}$$

Lemma 3.2.7 (Continuity of r) *Under Assumption 3.2.4,*

$$|r(\mu, h) - r(\mu', h')| \leq (\tilde{R} + 2L_{\tilde{r}}) d_{\mathcal{C}}((\mu, h), (\mu', h')). \tag{3.2.8}$$

Lemma 3.2.8 (Continuity of Φ) *Under Assumption 3.2.5,*

$$\|\Phi(\mu, h) - \Phi(\mu', h')\|_1 \leq (2L_P + 1) d_{\mathcal{C}}((\mu, h), (\mu', h')). \tag{3.2.9}$$

3.3 DPP for Q-function in MFC with learning

In this section, we establish the DPP of the Q-function for (MFC). Different from the well-understood DPP for single-agent control problem (see for example [122, chapter 9] and [121]), DPP for mean-field control problem has been established only recently on the lifted probability measure space [70, 138, 189]. We extend the approach of Chapter 2 to allow P and $\tilde{r}$ to depend on the population's action distribution ν_t.

First, by Lemma 3.2.2, (MFC) can be recast as a general Markov decision problem (MDP) with probability measure space as the new state-action space. More specifically, recall the set of admissible policies Π in (3.2.2), if one views the policy π_t to be a mapping from $\mathcal{P}(\mathcal{S})$ to $\mathcal{H}$, then (MFC) can be restated as the following MDP with unknown r and Φ:

$$\begin{aligned} v(\mu) &:= \sup_{\pi \in \Pi} \sum_{t=0}^{\infty} \gamma^t r(\mu_t, h_t) \\ \text{subject to} \quad & \mu_{t+1} = \Phi(\mu_t, h_t), \quad \mu_0 = \mu, \text{and } h_t(\cdot) \text{ in (3.2.1)}. \end{aligned} \tag{MDP}$$

With this reformulation, we can define the associated optimal Q-function for (MDP) starting from arbitrary $(\mu, h) \in \mathcal{C} = \mathcal{P}(\mathcal{S}) \times \mathcal{H}$,

$$Q(\mu, h) := \sup_{\pi \in \Pi} \left[\sum_{t=0}^{\infty} \gamma^t r(\mu_t, h_t) \middle| \mu_0 = \mu, \pi_0(\mu_0) = h \right], \tag{3.3.1}$$

with $h_t(\cdot)$ defined in (3.2.1). Similarly, define Q^π as the Q-function associated with a policy π:

$$Q^\pi(\mu, h) := \left[\sum_{t=0}^{\infty} \gamma^t r(\mu_t, h_t) \middle| \mu_0 = \mu, \pi_0(\mu_0) = h \right], \tag{3.3.2}$$

with $h_t(\cdot)$ defined in (3.2.1).

Remark 3.3.1 *With this reformulation,* (MFC) *is now lifted from the finite state-action space $\mathcal{X}$ and $\mathcal{U}$ to a compact continuous state-action space $\mathcal{C}$ embedded in an Euclidean space. In addition, the dynamics become deterministic by the aggregation over the original state-action space. Due to this aggregation for r, Φ, and the Q-function, we will subsequently refer this Q in* (3.3.1) *as an Integrated Q (IQ) function, to underline the difference between the Q-function for RL of single agent and that for MFC with learning.*

The following theorem shows Bellman equation for the IQ-function in (3.3.1).

Theorem 3.3.2 *For any $\mu \in \mathcal{P}(\mathcal{S})$,*

$$v(\mu) \quad = \quad \sup_{h\in\mathcal{H}} Q(\mu,h) = \sup_{h\in\mathcal{H}}\sup_{\pi\in\Pi} Q^{\pi}(\mu,h). \tag{3.3.3}$$

Moreover, the Bellman equation for $Q : \mathcal{C} \to \mathbb{R}$ is

$$Q(\mu,h) = r(\mu,h) + \gamma \sup_{\tilde{h}\in\mathcal{H}} Q(\Phi(\mu,h),\tilde{h}). \tag{3.3.4}$$

Proof of Theorem 3.3.2 Recall the definition of v in (MDP) and Q in (3.3.1). For $v(\mu)$, the supremum is taken over all the admissible policies Π, while for $Q(\mu,h)$, the supremum is taken over all the admissible policies Π with a further restriction that $\pi_0(\mu) = h$. Now in $\sup_{h\in\mathcal{H}} Q(\mu,h)$, since we are free to choose h, it is equivalent to v. Moreover,

$$\begin{aligned}
v(\mu) &= \sup_{\pi\in\Pi}\left[\sum_{t=0}^{\infty}\gamma^t r(\mu_t,\pi_t(\mu_t))\middle|\mu_0=\mu\right] \\
&= \sup_{\pi\in\Pi,\pi_0(\mu)=h,h\in\mathcal{H}}\left[\sum_{t=0}^{\infty}\gamma^t r(\mu_t,\pi_t(\mu_t))\middle|\mu_0=\mu,\pi_0(\mu_0)=h\right] \\
&= \sup_{h\in\mathcal{H}}\sup_{\pi\in\Pi,\pi_0(\mu)=h}\left[\sum_{t=0}^{\infty}\gamma^t r(\mu_t,\pi_t(\mu_t))\middle|\mu_0=\mu,\pi_0(\mu_0)=h\right] \\
&= \sup_{h\in\mathcal{H}} Q(\mu,h).
\end{aligned}$$

$$\begin{aligned}
Q(\mu,h) &= \sup_{\pi\in\Pi}\left[\sum_{t=0}^{\infty}\gamma^t r(\mu_t,\pi_t(\mu_t))\middle|\mu_0=\mu,\pi_0(\mu_0)=h\right] \\
&= r(\mu,h) + \sup_{\{\pi_t\}_{t=1}^{\infty}}\left[\sum_{t=1}^{\infty}\gamma^t r(\mu_t,\pi_t(\mu_t))\middle|\mu_1=\Phi(\mu,h)\right] \\
&= r(\mu,h) + \sup_{\{\pi_t\}_{t=0}^{\infty}}\gamma\left[\sum_{t=0}^{\infty}\gamma^t r(\mu_t,\pi_t(\mu_t))\middle|\mu_0=\Phi(\mu,h)\right] \\
&= r(\mu,h) + \gamma v(\Phi(\mu,h)) = r(\mu,h) + \gamma\sup_{h\in\mathcal{H}} Q(\Phi(\mu,h),h),
\end{aligned}$$

where the third equality is from shifting the time index by one. □

Next, we have the following verification theorem for this IQ-function.

Proposition 3.3.3 (Verification) *Assume Assumption 3.2.4. Define* $V_{\max} := \frac{R}{1-\gamma}$. *Then,*

- *Q defined in (3.3.1) is the unique function in $\{f \in \mathbb{R}^{\mathcal{C}} : \|f\|_\infty \le V_{\max}\}$ satisfying the Bellman equation (3.3.4).*
- *Suppose that for every $\mu \in \mathcal{P}(\mathcal{S})$, one can find an $h_\mu \in \mathcal{H}$ such that $h_\mu \in \arg\max_{h\in\mathcal{H}} Q(\mu, h)$, then $\pi^* = \{\pi_t^*\}_{t=0}^\infty$, where $\pi_t^*(\mu) = h_\mu$ for any $\mu \in \mathcal{P}(\mathcal{S})$ and $t \ge 0$, is an optimal stationary policy of (MDP).*

In order to prove the proposition, let us first define the following two operators.

- Define the operator $B : \mathbb{R}^{\mathcal{C}} \to \mathbb{R}^{\mathcal{C}}$ for (MDP)
$$(B\,q)(c) = r(c) + \gamma \max_{\tilde{h}\in\mathcal{H}} q(\Phi(c), \tilde{h}). \tag{3.3.5}$$
- Define the operator $B^\pi : \mathbb{R}^{\mathcal{C}} \to \mathbb{R}^{\mathcal{C}}$ for (MDP) under a given stationary policy $\{\pi_t = \pi : \mathcal{P}(\mathcal{S}) \to \mathcal{H}\}_{t=0}^\infty$
$$(B^\pi\, q)(c) = r(c) + \gamma q(\Phi(c), \pi(\Phi(c))). \tag{3.3.6}$$

Proof. Since $||\tilde{r}||_\infty \le R$, for any $\mu \in \mathcal{P}(\mathcal{S})$ and $h \in \mathcal{H}$, the aggregated reward function (3.2.6) satisfies $|r(\mu, h)| \le R \cdot \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} \mu(s)h(s)(a) = R$. In this case, for any $\mu \in \mathcal{P}(\mathcal{S})$, $h \in \mathcal{H}$ and policy π, $|Q^\pi(\mu, h)| \le R \cdot \sum_{t=0}^\infty \gamma^t = V_{\max}$. Hence, Q of (3.3.1) and Q^π of (3.3.2) both belong to $\{f \in \mathbb{R}^{\mathcal{C}} : \|f\|_\infty \le V_{\max}\}$. Meanwhile, by definition, it is easy to show that B and B^π map $\{f \in \mathbb{R}^{\mathcal{C}} : \|f\|_\infty \le V_{\max}\}$ to itself.

Next, we notice that B is a contraction operator with modulus $\gamma < 1$ under the sup norm on $\{f \in \mathbb{R}^{\mathcal{C}} : \|f\|_\infty \le V_{\max}\}$: for any $(\mu, h) \in \mathcal{C}$,

$$|Bq_1(\mu, h) - Bq_2(\mu, h)| \le \gamma \max_{\tilde{h}\in\mathcal{H}} |q_1(\Phi(\mu, h), \tilde{h}) - q_2(\Phi(\mu, h), \tilde{h})| \le \gamma \|q_1 - q_2\|_\infty.$$

Thus, $\|Bq_1 - Bq_2\|_\infty \le \gamma\|q_1 - q_2\|_\infty$. By Banach Fixed Point Theorem, B has a unique fixed point in $\{f \in \mathbb{R}^{\mathcal{C}} : \|f\|_\infty \le V_{\max}\}$. By (3.3.4) in Theorem 3.3.2, the unique fixed point is Q.

Similarly, we can show that for any stationary policy π, B^π is also a contraction operator with modulus $\gamma < 1$. Meanwhile, by the standard DPP argument as in Theorem 3.3.2, we have $Q^\pi = B^\pi Q^\pi$. This implies Q^π is the unique fixed point for B^π in $\{f \in \mathbb{R}^{\mathcal{C}} : \|f\|_\infty \le V_{\max}\}$.

Now let π^* be the stationary policy defined in the statement of Proposition 3.3.3. By definition, for any $c \in \mathcal{C}$, $Q(c) = r(c) + \gamma \max_{\tilde{h}\in\mathcal{H}} Q(\Phi(c), \tilde{h}) = r(c) + \gamma Q(\Phi(c), \pi^*(\Phi(c))) = B^{\pi^*}Q(c)$. Since B^{π^*} has a unique fixed point Q^{π^*} in $\{f \in \mathbb{R}^{\mathcal{C}} : \|f\|_\infty \le V_{\max}\}$, which is the IQ-function for the stationary policy π^*, clearly $Q^{\pi^*} = Q$, and the optimal IQ-function is attained by the optimal policy π^*. □

Lemma 3.3.1 (Characterization of Q) *Assume Assumptions 3.2.4 and 3.2.5, and $\gamma \cdot (2L_P + 1) < 1$. Q of (3.3.1) is continuous.*

The continuity property of Q from Lemma 3.3.1, along with the compactness of $\mathcal{H}$ and Proposition 3.3.3, leads to the following existence of stationary optimal policy.

Lemma 3.3.4 *Assume Assumptions 3.2.4, 3.2.5 and $\gamma \cdot (2L_P + 1) < 1$. There exists an optimal stationary policy $\pi^* : \mathcal{P}(\mathcal{S}) \to \mathcal{H}$ such that $Q^{\pi^*} = Q$.*

This existence of a stationary optimal policy is essential for the convergence analysis of our algorithm MFC-K-Q in Algorithm 3. In particular, it allows for comparing the optimal values of two MDPs with different action spaces: (MDP) and its variant defined in (3.5.9)-(3.5.10).

Note that the existence of a stationary optimal policy is well known when the state and action spaces are *finite* (see for example [163]) or *countably infinite* (see for example [122, chapter 9]). Yet, we are unable to find any prior corresponding result for the case with continuous state-action space.

3.4 MFC-K-Q Algorithm via Kernel Regression and Approximated Bellman Operator

In this section, we will develop a kernel-based Q-learning algorithm (MFC-K-Q) for the MFC problem with learning based on (3.3.4).

Note from (3.3.4), the MFC problem with learning is different from the classical MDP [163] in two aspects. First, the lifted state space $\mathcal{P}(\mathcal{S})$ and lifted action space $\mathcal{H}$ are continuous, rather than discrete or finite. Second, the maximum in the Bellman operator is taken over a continuous space $\mathcal{H}$.

To handle the lifted continuous state-action space, we use a kernel regression method on the discretized state-action space. Kernel regression is a local averaging approach for approximating *unknown* state-action pair from *observed* data on a discretized space called *ϵ-net*. Mathematically, a set $\mathcal{C}_\epsilon = \{c^i = (\mu^i, h^i)\}_{i=1}^{N_\epsilon}$ is an ϵ-net for $\mathcal{C}$ if $\min_{1 \le i \le N_\epsilon} d_{\mathcal{C}}(c, c^i) < \epsilon$ for all $c \in \mathcal{C}$. Here N_ϵ is the size of $\mathcal{C}_\epsilon$. Note that compactness of $\mathcal{C}$ implies the existence of such an ϵ-net $\mathcal{C}_\epsilon$. The choice of ϵ is critical for the convergence and the sample complexity analysis.

Correspondingly, we define the so-called *kernel regression operator* $\Gamma_K : \mathbb{R}^{\mathcal{C}_\epsilon} \to \mathbb{R}^{\mathcal{C}}$:

$$\Gamma_K f(c) = \sum_{i=1}^{N_\epsilon} K(c^i, c) f(c^i), \tag{3.4.1}$$

where $K(c^i, c) \ge 0$ is a weighted kernel function such that for all $c \in \mathcal{C}$ and $c^i \in \mathcal{C}_\epsilon$,

$$\sum_{i=1}^{N_\epsilon} K(c^i, c) = 1, \text{ and } K(c^i, c) = 0 \text{ if } d_{\mathcal{C}}(c^i, c) > \epsilon. \tag{3.4.2}$$

In fact, K can be of any form

$$K(c^i, c) = \frac{\phi(c^i, c)}{\sum_{i=1}^{N_\epsilon} \phi(c^i, c)}, \tag{3.4.3}$$

with some function ϕ satisfying $\phi \geq 0$ and $\phi(x, y) = 0$ when $d_{\mathcal{C}}(x, y) \geq \epsilon$. (See Section 3.7 for some choices of ϕ).

Meanwhile, to avoid maximizing over a continuous space $\mathcal{H}$ as in the Bellman equation (3.3.4), we take the maximum over the ϵ-net $\mathcal{H}_\epsilon$ on $\mathcal{H}$. Here $\mathcal{H}_\epsilon$ is an ϵ-net on $\mathcal{H}$ induced from $\mathcal{C}_\epsilon$, i.e., $\mathcal{H}_\epsilon$ contains all the possible action choices in $\mathcal{C}_\epsilon$, whose size is denoted by $N_{\mathcal{H}_\epsilon}$.

The corresponding approximated Bellman operator B_ϵ acting on functions is then defined on the ϵ-net $\mathcal{C}_\epsilon$: $\mathbb{R}^{\mathcal{C}_\epsilon} \to \mathbb{R}^{\mathcal{C}_\epsilon}$ such that

$$(B_\epsilon\, q)(c^i) = r(c^i) + \gamma \max_{\tilde{h} \in \mathcal{H}_\epsilon} \Gamma_K q(\Phi(c^i), \tilde{h}). \tag{3.4.4}$$

Since $(\Phi(c^i), \tilde{h})$ may not be on the ϵ-net, one needs to approximate the value at that point via the kernel regression $\Gamma_K q(\Phi(c^i), \tilde{h})$.

In practice, one may only have access to noisy estimations $\{\widehat{r}(c^i), \widehat{\Phi}(c^i)\}_{i=1}^{N_\epsilon}$ instead of the accurate data $\{r(c^i), \Phi(c^i)\}_{i=1}^{N_\epsilon}$ on $\mathcal{C}_\epsilon$. Taking this into consideration, Algorithm 3 consists of two steps. First, it collects samples on $\mathcal{C}$ given an exploration policy. For each component c^i on the ϵ-net $\mathcal{C}_\epsilon$, the estimated data $(\widehat{r}(c^i), \widehat{\Phi}(c^i))$ is computed by averaging samples in the ϵ-neighborhood of c^i. Second, the fixed point iteration is applied to the approximated Bellman operator B_ϵ with $\{\widehat{r}(c^i), \widehat{\Phi}(c^i)\}_{i=1}^{N_\epsilon}$. Under appropriate conditions, Algorithm 3 provides an accurate estimation of the true Q-function with efficient sample complexity (See Theorem 3.5.5).

3.5 Convergence and Sample Complexity Analysis of MFC-K-Q

In this section, we will establish the convergence of MFC-K-Q algorithm and analyze its sample complexity. The convergence analysis in Section 3.5.1 relies on studying the fixed point iteration of B_ϵ; and the complexity analysis in Section 3.5.2 is based on an upper bound of the necessary sample size to visit each ϵ-neighborhood of the ϵ-net at least once.

In addition to Assumptions 3.2.4 and 3.2.5, the following conditions are needed for the convergence and the sample complexity analysis.

Assumption 3.5.1 (Controllability of the dynamics) *For all ϵ, there exists $M_\epsilon \in \mathbb{N}$ such that for any ϵ-net $\mathcal{H}_\epsilon$ on $\mathcal{H}$ and $\mu, \mu' \in \mathcal{P}(\mathcal{S})$, there exists an action sequence $(h^1, \ldots, h^m)$ with $h^i \in \mathcal{H}_\epsilon$ and $m < M_\epsilon$, with which the state μ will be driven to an ϵ-neighborhood of μ'.*

Algorithm 3 Kernel-based Q-learning Algorithm for MFC (MFC-K-Q)

Input: Initial state distribution μ_0, $\epsilon > 0$, ϵ-net on $\mathcal{C}$: $\mathcal{C}_\epsilon = \{c^i = (\mu^i, h^i)\}_{i=1}^{N_\epsilon}$, exploration policy π taking actions from $\mathcal{H}_\epsilon$ induced from $\mathcal{C}_\epsilon$, regression kernel K on $\mathcal{C}_\epsilon$.
Initialize: $\widehat{r}(c^i) = 0$, $\widehat{\Phi}(c^i) = 0$, $N(c^i) = 0, \forall i$.
repeat
  At the current state distribution μ_t, act h_t according to π, observe $\mu_{t+1} = \Phi(\mu_t, h_t)$ and $r_t = r(\mu_t, h_t)$.
  for $1 \le i \le N_\epsilon$ **do**
    if $d_{\mathcal{C}}(c^i, (\mu_t, h_t)) < \epsilon$ **then**
      $N(c^i) \leftarrow N(c^i) + 1$.
      $\widehat{r}(c^i) \leftarrow \frac{N(c^i)-1}{N(c^i)} \cdot \widehat{r}(c^i) + \frac{1}{N(c^i)} \cdot r_t$
      $\widehat{\Phi}(c^i) \leftarrow \frac{N(c^i)-1}{N(c^i)} \cdot \widehat{\Phi}(c^i) + \frac{1}{N(c^i)} \cdot \mu_t$
    end if
  end for
until $N(c^i) > 0, \forall i$.
Initialize: $\widehat{q}_0(c^i) = 0, \forall c^i \in \mathcal{C}_\epsilon$, $l = 0$.
repeat
  for $c^i \in \mathcal{C}_\epsilon$ **do**
    $\widehat{q}_{l+1}(c^i) \leftarrow \Big(\widehat{r}(c^i) + \gamma \max_{\tilde{h} \in \mathcal{H}_\epsilon} \Gamma_K \widehat{q}_l(\widehat{\Phi}(c^i), \tilde{h})\Big)$.
  end for
  $l = l + 1$.
until converge

Assumption 3.5.2 (Regularity of kernels) *For any point $c \in \mathcal{C}$, there exist at most N_K points c^i's in $\mathcal{C}_\epsilon$ such that $K(c^i, c) > 0$. Moreover, there exists an $L_K > 0$ such that for all $c \in \mathcal{C}_\epsilon, c', c'' \in \mathcal{C}, |K(c, c') - K(c, c'')| \le L_K \cdot d_{\mathcal{C}}(c', c'')$.*

Assumption 3.5.1 ensures the dynamics to be controllable. Assumption 3.5.2 is easy to be satisfied: take a uniform grid as the ϵ-net, then N_K is roughly bounded from above by $2^{\dim(\mathcal{C})}$; meanwhile, a number of commonly used kernels, including the triangular kernel in Section 3.7, satisfy the Lipschitz condition in Assumption 3.5.2.

3.5.1 Convergence Analysis

To start, recall the Lipschitz continuity of the aggregated rewards r and dynamics Φ from Lemma 3.2.7 and Lemma 3.2.8. To simplify the notation, denote $L_r := \tilde{R} + 2L_{\tilde{r}}$ as the Lipschitz constant of r and $L_\Phi := 2L_P + 1$ as the Lipschitz constant of Φ.

Next, recall that there are three sources of the approximation error in Algorithm 3: the kernel regression Γ_K on $\mathcal{C}$ with the ϵ-net $\mathcal{C}_\epsilon$, the discretized action space $\mathcal{H}_\epsilon$ on $\mathcal{H}$, and the

sampled data $\widehat{r}$ and $\widehat{\Phi}$ for both the dynamics and the rewards.

The key idea for the convergence analysis is to decompose the error based on these sources and to analyze each decomposed error accordingly. That is to consider the following different types of Bellman operators:

- the operator B in (3.3.5) for (MDP);
- the operator $B_{\mathcal{H}_\epsilon} : \mathbb{R}^{\mathcal{C}} \to \mathbb{R}^{\mathcal{C}}$ which involves the discretized action space $\mathcal{H}_\epsilon$

$$B_{\mathcal{H}_\epsilon} q(c) = r(c) + \gamma \max_{\tilde{h} \in \mathcal{H}_\epsilon} q(\Phi(c), \tilde{h}); \tag{3.5.1}$$

- the operator B_ϵ in (3.4.4) defined on the ϵ-net $\mathcal{C}_\epsilon$, which involves the discretized action space $\mathcal{H}_\epsilon$, and the kernel approximation;
- the operator $\widehat{B}_\epsilon : \mathbb{R}^{\mathcal{C}_\epsilon} \to \mathbb{R}^{\mathcal{C}_\epsilon}$ defined by

$$(\widehat{B}_\epsilon\, q)(c^i) = \widehat{r}(c^i) + \gamma \max_{\tilde{h} \in \mathcal{H}_\epsilon} \Gamma_K q(\widehat{\Phi}(c^i), \tilde{h}), \tag{3.5.2}$$

  which involves the discretized action space $\mathcal{H}_\epsilon$, the kernel approximation, and the estimated data.

- the operator T that maps $\{f \in \mathbb{R}^{\mathcal{P}(\mathcal{S})} : \|f\|_\infty \le V_{\max}\}$ to itself, such that

$$Tv(\mu) = \max_{h \in \mathcal{H}_\epsilon} (r(\mu, h) + \gamma v(\Phi(\mu, h))). \tag{3.5.3}$$

We show that under mild assumptions, each of the above operators admits a unique fixed point.

Lemma 3.5.3 *Assume Assumption 3.2.4. Let* $V_{\max} := \frac{R}{1-\gamma}$. *Then,*

- *B in (3.3.5) has a unique fixed point in $\{f \in \mathbb{R}^{\mathcal{C}} : \|f\|_\infty \le V_{\max}\}$. That is, there exists a unique Q such that*

$$(B\, Q)(c) = r(c) + \gamma \max_{\tilde{h} \in \mathcal{H}} Q(\Phi(c), \tilde{h}). \tag{3.5.4}$$

- *$B_{\mathcal{H}_\epsilon}$ in (3.5.1) has a unique fixed point in $\{f \in \mathbb{R}^{\mathcal{C}} : \|f\|_\infty \le V_{\max}\}$. That is, there exists a unique $Q_{\mathcal{H}_\epsilon}$ such that*

$$B_{\mathcal{H}_\epsilon} Q_{\mathcal{H}_\epsilon}(c) = r(c) + \gamma \max_{\tilde{h} \in \mathcal{H}_\epsilon} Q_{\mathcal{H}_\epsilon}(\Phi(c), \tilde{h}). \tag{3.5.5}$$

- *B_ϵ in (3.4.4) has a unique fixed point in $\{f \in \mathbb{R}^{\mathcal{C}_\epsilon} : \|f\|_\infty \le V_{\max}\}$. That is, there exists a unique Q_ϵ such that for any $c^i \in \mathcal{C}_\epsilon$,*

$$(B_\epsilon\, Q_\epsilon)(c^i) = r(c^i) + \gamma \max_{\tilde{h} \in \mathcal{H}_\epsilon} \Gamma_K Q_\epsilon(\Phi(c^i), \tilde{h}). \tag{3.5.6}$$

- *$\widehat{B}_\epsilon$ in (3.5.2) has a unique fixed point in $\{f \in \mathbb{R}^{\mathcal{C}_\epsilon} : \|f\|_\infty \leq V_{\max}\}$. That is, there exists a unique $\widehat{Q}_\epsilon$ such that for any $c^i \in \mathcal{C}_\epsilon$, and $\widehat{r}, \widehat{\Phi}$ sampled from c^i's ϵ-neighborhood,*

$$(\widehat{B}_\epsilon \widehat{Q}_\epsilon)(c^i) = \widehat{r}(c^i) + \gamma \max_{\tilde{h} \in \mathcal{H}_\epsilon} \Gamma_K \widehat{Q}_\epsilon(\widehat{\Phi}(c^i), \tilde{h}). \tag{3.5.7}$$

- *T has a unique fixed point $V_{\mathcal{H}_\epsilon}$ in $\{f \in \mathbb{R}^{\mathcal{P}(\mathcal{S})} : \|f\|_\infty \leq V_{\max}\}$. That is*

$$T V_{\mathcal{H}_\epsilon}(\mu) = \max_{h \in \mathcal{H}_\epsilon} (r(\mu, h) + \gamma V_{\mathcal{H}_\epsilon}(\Phi(\mu, h))). \tag{3.5.8}$$

Lemma 3.5.4 (Characterization of $Q_{\mathcal{H}_\epsilon}$) *Assume Assumption 3.2.4. $V_{\mathcal{H}_\epsilon}$ in (3.5.8) is the optimal value function for the following MFC problem with continuous state space $\mathcal{P}(\mathcal{S})$ and discretized action space $\mathcal{H}_\epsilon$.*

$$V_{\mathcal{H}_\epsilon}(\mu) = \sup_{\pi \in \Pi_\epsilon} \sum_{t=0}^{\infty} \gamma^t r(\mu_t, \pi_t(\mu_t)) \tag{3.5.9}$$

with $\Pi_\epsilon := \{\pi = \{\pi_t\}_{t=0}^\infty | \pi_t : \mathcal{P}(\mathcal{S}) \to \mathcal{H}_\epsilon\}$, subject to

$$\mu_{t+1} = \Phi(\mu_t, \pi_t(\mu_t)), \mu_0 = \mu. \tag{3.5.10}$$

Moreover, $Q_{\mathcal{H}_\epsilon}$ in (3.5.5) and $V_{\mathcal{H}_\epsilon}$ in (3.5.8) satisfy the following relation:

$$Q_{\mathcal{H}_\epsilon}(\mu, h) = r(\mu, h) + \gamma V_{\mathcal{H}_\epsilon}(\Phi(\mu, h)), \tag{3.5.11}$$

and $Q_{\mathcal{H}_\epsilon}$ is Lipschitz continuous.

This connection between $Q_{\mathcal{H}_\epsilon}$ and the optimal value function $V_{\mathcal{H}_\epsilon}$ of the MFC problem with continuous state space $\mathcal{P}(\mathcal{S})$ and discretized action space $\mathcal{H}_\epsilon$, is critical for estimating the error bounds in the convergence analysis.

Theorem 3.5.5 (Convergence) *Given $\epsilon > 0$. Assume Assumptions 3.2.4, 3.2.5, 3.5.1, and 3.5.2, and $\gamma \cdot L_\Phi < 1$. Let $\widehat{B}_\epsilon : \mathbb{R}^{\mathcal{C}_\epsilon} \to \mathbb{R}^{\mathcal{C}_\epsilon}$ be the operator defined in (3.5.2)*

$$(\widehat{B}_\epsilon q)(c^i) = \widehat{r}(c^i) + \gamma \max_{\tilde{h} \in \mathcal{H}_\epsilon} \Gamma_K q(\widehat{\Phi}(c^i), \tilde{h}),$$

where $\widehat{r}(c)$ and $\widehat{\Phi}(c)$ are sampled from an ϵ-neighborhood of c, then it has a unique fixed point $\widehat{Q}_\epsilon$ in $\{f \in \mathbb{R}^{\mathcal{C}_\epsilon} : ||f||_\infty \leq V_{\max}\}$. Moreover, the sup distance between $\Gamma_K \widehat{Q}_\epsilon$ in (3.4.1) and Q in (3.3.1) is

$$||Q - \Gamma_K \widehat{Q}_\epsilon||_\infty \leq \frac{L_r + 2\gamma N_K L_K V_{\max} L_\Phi}{1 - \gamma} \cdot \epsilon + \frac{2L_r}{(1 - \gamma L_\Phi)(1 - \gamma)} \cdot \epsilon. \tag{3.5.12}$$

In particular, for a fixed ϵ, Algorithm 3 converges linearly to $\widehat{Q}_\epsilon$.

Proof of Theorem 3.5.5 The proof of the the convergence is to quantify $||Q - \Gamma_K \widehat{Q}_\epsilon||_\infty$ from the following estimate

$$||Q - \Gamma_K \widehat{Q}_\epsilon||_\infty \leq \underbrace{||Q - Q_{\mathcal{H}_\epsilon}||_\infty}_{(I)} + \underbrace{||Q_{\mathcal{H}_\epsilon} - \Gamma_K Q_\epsilon||_\infty}_{(II)} + \underbrace{||\Gamma_K Q_\epsilon - \Gamma_K \widehat{Q}_\epsilon||_\infty}_{(III)}. \tag{3.5.13}$$

(I) can be regarded as the approximation error from discretizing the lifted action space $\mathcal{H}$ by $\mathcal{H}_\epsilon$; (II) is the error from the kernel regression on $\mathcal{C}$ with the ϵ-net $\mathcal{C}_\epsilon$; and (III) is estimating the error introduced by the sampled data $\widehat{r}$ and $\widehat{\Phi}$.

Step 1. We shall use 3.3.4 and Lemmas 3.5.4 to show that

$$||Q - Q_{\mathcal{H}_\epsilon}||_\infty \leq \frac{L_r}{(1 - \gamma L_\Phi)(1 - \gamma)} \cdot \epsilon.$$

By Lemma 3.5.4, $Q(c) - Q_{\mathcal{H}_\epsilon}(c) = \gamma\big(V(\Phi(c)) - V_{\mathcal{H}_\epsilon}(\Phi(c))\big)$, where V is the optimal value function of the problem on $\mathcal{P}(\mathcal{S})$ and $\mathcal{H}$ in (MDP), and $V_{\mathcal{H}_\epsilon}$ is the optimal value function of the problem on $\mathcal{P}(\mathcal{S})$ and $\mathcal{H}_\epsilon$ (3.5.9)-(3.5.10). Hence it suffices to prove that

$$||V - V_{\mathcal{H}_\epsilon}||_\infty \leq \frac{L_r}{(1 - \gamma L_\Phi)(1 - \gamma)} \cdot \epsilon.$$

We adopt the similar strategy as in the proof of Lemma 3.3.4.

Let π^* be the optimal policy of (MDP), whose existence is shown in Lemma 3.3.4. For any $\mu \in \mathcal{P}(\mathcal{S})$, let $(\mu, h) = (\mu_0, h_0), (\mu_1, h_1), (\mu_2, h_2), \ldots, (\mu_t, h_t), \ldots$ be the trajectory of the system under the optimal policy π^*, starting from μ. We have $V(\mu) = \sum_{t=0}^\infty \gamma^t r(\mu_t, h_t)$.

Now let h^{i_t} be the nearest neighbor of h_t in $\mathcal{H}_\epsilon$. $d_{\mathcal{H}}(h^{i_t}, h_t) \leq \epsilon$. Consider the trajectory of the system starting from μ and then taking $h^{i_0}, \ldots, h^{i_t}, \ldots$, denote the corresponding state by μ'_t. We have $V_{\mathcal{H}_\epsilon} \geq \sum_{t=0}^\infty \gamma^t r(\mu'_t, h^{i_t})$, since $V_{\mathcal{H}_\epsilon}$ is the optimal value function.

$$d_{\mathcal{P}(\mathcal{S})}(\mu'_t, \mu_t) = d_{\mathcal{P}(\mathcal{S})}\big(\Phi(\mu'_{t-1}, h^{i_{t-1}}), \Phi(\mu_{t-1}, h_t)\big) \leq L_\Phi \cdot \big(d_{\mathcal{P}(\mathcal{S})}(\mu'_{t-1}, \mu_{t-1}) + \epsilon\big)$$

By the iteration, we have $d_{\mathcal{P}(\mathcal{S})}(\mu'_t, \mu_t) \leq \frac{L_\Phi - L_\Phi^{t+1}}{1 - L_\Phi} \cdot \epsilon$, and $|r(\mu'_t, h^{i_t}) - r(\mu_t, h_t)| \leq L_r \cdot \big(d_{\mathcal{P}(\mathcal{S})}(\mu'_t, \mu_t) + \epsilon\big) \leq L_r \cdot \frac{L_\Phi^{t+1} - 1}{L_\Phi - 1} \cdot \epsilon$, which implies

$$\begin{aligned} 0 &\leq V(\mu) - V_{\mathcal{H}_\epsilon}(\mu) \\ &\leq \sum_{t=0}^\infty \gamma^t (r(\mu_t, h_t) - r(\mu'_t, h^{i_t})) \\ &\leq \sum_{t=0}^\infty \gamma^t \cdot L_r \cdot \frac{L_\Phi^{t+1} - 1}{L_\Phi - 1} \cdot \epsilon \\ &= \frac{L_r}{(1 - \gamma L_\Phi)(1 - \gamma)} \cdot \epsilon. \end{aligned}$$

Here $0 \leq V(\mu) - V_{\mathcal{H}_\epsilon}(\mu)$ is by the optimality of $V_{\mathcal{C}}$.

Step 2. We shall use Lemmas 3.5.3 and 3.5.4 to show that

$$||Q_{\mathcal{H}_\epsilon} - \Gamma_K Q_\epsilon||_\infty \leq \frac{L_r}{(1-\gamma L_\Phi)(1-\gamma)} \cdot \epsilon.$$

Note that

$$\begin{aligned}
& ||\Gamma_K Q_\epsilon - Q_{\mathcal{H}_\epsilon}||_\infty \\
= \; & ||\Gamma_K B_\epsilon Q_\epsilon - Q_{\mathcal{H}_\epsilon}||_\infty \\
= \; & ||\Gamma_K B_{\mathcal{H}_\epsilon} \Gamma_K Q_\epsilon - Q_{\mathcal{H}_\epsilon}||_\infty \\
\leq \; & ||\Gamma_K B_{\mathcal{H}_\epsilon} \Gamma_K Q_\epsilon - \Gamma_K B_{\mathcal{H}_\epsilon} Q_{\mathcal{H}_\epsilon}||_\infty + ||\Gamma_K B_{\mathcal{H}_\epsilon} Q_{\mathcal{H}_\epsilon} - Q_{\mathcal{H}_\epsilon}||_\infty \\
= \; & ||\Gamma_K B_{\mathcal{H}_\epsilon} \Gamma_K Q_\epsilon - \Gamma_K B_{\mathcal{H}_\epsilon} Q_{\mathcal{H}_\epsilon}||_\infty + ||\Gamma_K Q_{\mathcal{H}_\epsilon} - Q_{\mathcal{H}_\epsilon}||_\infty \\
\leq \; & \gamma ||\Gamma_K Q_\epsilon - Q_{\mathcal{H}_\epsilon}||_\infty + ||\Gamma_K Q_{\mathcal{H}_\epsilon} - Q_{\mathcal{H}_\epsilon}||_\infty.
\end{aligned}$$

Here the first and the third equalities hold since Q_ϵ is the fixed point of B_ϵ and $Q_{\mathcal{H}_\epsilon}$ is the fixed point of $B_{\mathcal{H}_\epsilon}$. The second inequality is by the fact that Γ_K is a non-expansion mapping, i.e., $\|\Gamma_K f\|_\infty \leq \|f\|_\infty$, and that $B_{\mathcal{H}_\epsilon}$ is a contraction with modulus γ with the supremum norm. Meanwhile, for any Lipschitz function $f \in \mathbb{R}^{\mathcal{C}}$ with Lipschitz constant L, we have for all $c \in \mathcal{C}$,

$$|\Gamma_K f(c) - f(c)| = \sum_{i=1}^{N_\epsilon} K(c, c^i)|f(c^i) - f(c)| \leq \sum_{i=1}^{N_\epsilon} K(c, c^i)\epsilon L = \epsilon L.$$

Note here the inequality follows from $K(c, c^i) = 0$ for all $d_{\mathcal{C}}(c, c^i) \geq \epsilon$. Therefore, $||\Gamma_K Q_\epsilon - Q_{\mathcal{H}_\epsilon}||_\infty \leq \frac{L_{Q_{\mathcal{H}_\epsilon}}}{1-\gamma}\epsilon$, where $L_{Q_{\mathcal{H}_\epsilon}} = \frac{L_r}{1-\gamma L_\Phi}$ is the Lipschitz constant for $Q_{\mathcal{H}_\epsilon}$.

Final step. Let q_0 denote the zero function on $\mathcal{C}_\epsilon$. By Lemma 3.5.3, $Q_\epsilon = \lim_{n\to\infty} B_\epsilon^n q_0$, and $\widehat{Q}_\epsilon = \lim_{n\to\infty} \widehat{B}_\epsilon^n q_0$. Denote $q_n := B_\epsilon^n q_0$, $\widehat{q}_n := \widehat{B}_\epsilon^n q_0$, and $e_n := ||q_n - \widehat{q}_n||_\infty$. For any $c \in \mathcal{C}_\epsilon$,

$$\begin{aligned}
e_{n+1}(c) \; &= \; \left| \widehat{r}(c) + \gamma \max_{\tilde{h}\in\mathcal{H}_\epsilon} \Gamma_K \widehat{q}_n(\widehat{\Phi}(c), \tilde{h}) - r(c) - \gamma \max_{\tilde{h}\in\mathcal{H}_\epsilon} \Gamma_K q_n(\Phi(c), \tilde{h}) \right| \\
&\leq \; |\widehat{r}(c) - r(c)| + \gamma \max_{\tilde{h}\in\mathcal{H}_\epsilon} \left| \Gamma_K \widehat{q}_n(\widehat{\Phi}(c), \tilde{h}) - \Gamma_K q_n(\Phi(c), \tilde{h}) \right| \\
&\leq \; \epsilon L_r + \gamma \max_{\tilde{h}\in\mathcal{H}_\epsilon} \Bigg[\left| \Gamma_K \widehat{q}_n(\widehat{\Phi}(c), \tilde{h}) - \Gamma_K \widehat{q}_n(\Phi(c), \tilde{h}) \right| \\
&\qquad + \left| \Gamma_K \widehat{q}_n(\Phi(c), \tilde{h}) - \Gamma_K q_n(\Phi(c), \tilde{h}) \right| \Bigg].
\end{aligned}$$

Here $|\widehat{r}(c) - r(c)| \leq \epsilon L_r$ because $\widehat{r}(c)$ is sampled from an ϵ-neighborhood of c and by Assumption 3.2.4. Moreover, for any fixed $\tilde{h}$,

$$\begin{aligned}\left|\Gamma_K \widehat{q}_n(\widehat{\Phi}(c), \tilde{h}) - \Gamma_K \widehat{q}_n(\Phi(c), \tilde{h})\right| &= \left|\sum_{i=1}^{N_\epsilon} (K(c^i, (\widehat{\Phi}(c), \tilde{h})) - K(c^i, (\Phi(c), \tilde{h})))\widehat{q}_n(c^i)\right| \\ &\leq 2N_K L_K V_{\max} \cdot d_{\mathcal{P}(\mathcal{S})}(\widehat{\Phi}(c), \Phi(c)) \leq 2N_K L_K V_{\max} L_\Phi \epsilon.\end{aligned}$$

The first inequality comes from Assumption 3.5.2, because $K(c^i, (\widehat{\Phi}(c), \tilde{h})) - K(c^i, (\Phi(c), \tilde{h}))$ is nonzero for at most $2N_K$ index $i \in \{1, 2, \ldots, N_\epsilon\}$, K is Lipschitz continuous, and $||\widehat{q}_n||_\infty \leq V_{\max}$. The second inequality comes from the fact that $\widehat{\Phi}(c)$ is sampled from an ϵ-neighborhood of c and by Assumption 3.2.5. Meanwhile,

$$\left|\Gamma_K \widehat{q}_n(\Phi(c), \tilde{h}) - \Gamma_K q_n(\Phi(c), \tilde{h})\right| \leq ||q_n - \widehat{q}_n||_\infty = e_n,$$

since Γ is non-expansion. Putting these pieces together, we have

$$e_{n+1} = \max_{c \in \mathcal{C}_\epsilon} e_{n+1}(c) \leq \epsilon L_r + \epsilon\gamma 2N_K L_K V_{\max} L_\Phi + \gamma e_n.$$

In this case, elementary algebra shows that

$$e_n \leq \epsilon \cdot \frac{L_r + \gamma 2N_K L_K V_{\max} L_\Phi}{1 - \gamma}, \forall n.$$

Then since Γ_K is non-expansion,

$$||\Gamma_K Q_{\mathcal{C}_\epsilon} - \Gamma_K \widehat{Q}_\epsilon||_\infty \leq \epsilon \cdot \frac{L_r + \gamma 2N_K L_K V_{\max} L_\Phi}{1 - \gamma},$$

hence the error bound (3.5.12).

The claim regarding the convergence rate follows from the $\gamma-$contraction of operator $\widehat{B}_\epsilon$. □

3.5.2 Sample Complexity Analysis

In classical Q-learning for MDPs with stochastic environment, every component in the ϵ-net is required to be visited a number of times in order to get desirable estimate for the Q-function. The usual terminology *covering time* refers to the expected number of steps to visit every component in the ϵ-net at least once, for a given exploration policy. The complexity analysis thus focuses on the necessary rounds of the covering time.

In contrast, visiting each component in the ϵ-net *once* is sufficient with deterministic dynamics. We will demonstrate that using deterministic mean-field dynamics to approximate N-agent stochastic environment will indeed significantly reduce the complexity analysis.

To start, denote $T_{\mathcal{C},\pi}$ as the covering time of the ϵ-net under (random) policy π, such that

$$T_{\mathcal{C},\pi} := \sup_{\mu\in\mathcal{P}(\mathcal{S})} \inf \Big\{ t>0 : \mu_0=\mu, \forall c^i \in \mathcal{C}_\epsilon, \exists t_i \le t, \\ (\mu_{t_i}, h_{t_i}) \text{ in the } \epsilon\text{-neighborhood of } c^i, \text{ under the policy } \pi \Big\}.$$

Recall that an ϵ'-greedy policy on $\mathcal{H}_\epsilon$ is a policy which with probability at least ϵ' will uniformly explore the actions on $\mathcal{H}_\epsilon$. Note that this type of policy always exists. And we have the following sample complexity result.

Theorem 3.5.6 (Sample complexity) *Given $\epsilon, \delta > 0$ and Assumption 3.5.1, for any $\epsilon' > 0$, let $\pi_{\epsilon'}$ be an ϵ'-greedy policy on $\mathcal{H}_\epsilon$. Then*

$$\mathbb{E}[T_{\mathcal{C},\pi_{\epsilon'}}] \le \frac{(M_\epsilon+1)\cdot(N_{\mathcal{H}_\epsilon})^{M_\epsilon+1}}{(\epsilon')^{M_\epsilon+1}} \cdot \log(N_\epsilon). \tag{3.5.14}$$

Here M_ϵ is defined in Assumption 3.5.1. Moreover, with probability $1-\delta$, for any initial state μ, under the ϵ'-greedy policy, the dynamics will visit each ϵ-neighborhood of elements in $\mathcal{C}_\epsilon$ at least once, after

$$\frac{(M_\epsilon+1)\cdot(N_{\mathcal{H}_\epsilon})^{M_\epsilon+1}}{(\epsilon')^{M_\epsilon+1}} \cdot \log(N_\epsilon) \cdot e \cdot \log(1/\delta). \tag{3.5.15}$$

time steps, where $\log(N_\epsilon) = \Theta(|\mathcal{S}||\mathcal{A}|\log(1/\epsilon))$, and $N_{\mathcal{H}_\epsilon} = \Theta((\frac{1}{\epsilon})^{(|\mathcal{A}|-1|)|\mathcal{S}|})$.

Theorem 3.5.6 provides an upper bound $\Omega(\text{poly}((1/\epsilon)\cdot\log(1/\delta)))$ for the covering time under the ϵ'-greedy policy, in terms of the size of the ϵ-net and the accuracy $1/\delta$. The proof of Theorem 3.5.6 relies on the following lemma.

Lemma 3.5.7 *Assume for some policy π, $\mathbb{E}[T_{\mathcal{C},\pi}] \le T < \infty$. Then with probability $1-\delta$, for any initial state μ, under the policy π, the dynamics will visit each ϵ-neighborhood of elements in $\mathcal{C}_\epsilon$ at least once, after $T\cdot e\cdot\log(1/\delta)$ time steps, i.e. $\mathbb{P}(T_{\mathcal{C},\pi} \le T\cdot e\cdot\log(1/\delta)) \ge 1-\delta$.*

Proof of Theorem 3.5.6 Recall there are N_ϵ different pairs in the ϵ-net. Denote the ϵ-neighborhoods of those pairs by $B_\epsilon = \{B^i\}_{i=1}^{N_\epsilon}$. Without loss of generality, we may assume that B^i are disjoint, since the covering time will only become smaller if they overlap with each other. Let $T_k := \min\{t>1 : \text{k of } B_\epsilon \text{ is visited}\}$. $T_k - T_{k-1}$ is the time to visit a new neighborhood after $k-1$ neighborhoods are visited. By Assumption 3.5.1, for any $B^i \in B_\epsilon$ with center (μ^i, h^i), $\mu \in \mathcal{P}(\mathcal{S})$, there exists a sequence of actions in $\mathcal{H}_\epsilon$, whose length is at most M_ϵ, such that starting from μ and taking that sequence of actions will lead the visit of the ϵ-neighborhood of μ^i. Then, at that point, taking h^i will yield the visit of B^i. Hence

$\forall B^i \in B_\epsilon$, $\mu \in \mathcal{P}(\mathcal{S})$,

$$\mathbb{P}(B^i \text{ is visited in } M_\epsilon + 1 \text{ steps} \,|\, \mu_{T_{k-1}} = \mu) \geq \left(\frac{\epsilon'}{N_{\mathcal{H}_\epsilon}}\right)^{M_\epsilon+1}.$$

$$\mathbb{P}(\text{a new neighborhood is visited in } M_\epsilon + 1 \text{ steps} \,|\mu_{T_{k-1}} = \mu) \geq (N_\epsilon - k + 1) \cdot \left(\frac{\epsilon'}{N_{\mathcal{H}_\epsilon}}\right)^{M_\epsilon+1}.$$

This implies $\mathbb{E}[T_k - T_{k-1}] \leq \frac{M_\epsilon+1}{N_\epsilon-k+1} \cdot (\frac{N_{\mathcal{H}_\epsilon}}{\epsilon'})^{M_\epsilon+1}$. Summing $\mathbb{E}[T_k - T_{k-1}]$ from $k = 1$ to $k = N_\epsilon$ yields the desired result. The second part follows directly from Lemma 3.5.7. Meanwhile, $N_{\mathcal{H}_\epsilon}$, the size of the ϵ-net in $\mathcal{H}$ is $\Theta((\frac{1}{\epsilon})^{(|\mathcal{A}|-1)|\mathcal{S}|})$, because $\mathcal{H}$ is a compact $(|\mathcal{A}| - 1)|\mathcal{S}|$ dimensional manifold. Similarly, $N_\epsilon = \Theta((\frac{1}{\epsilon})^{|\mathcal{A}||\mathcal{S}|-1})$ as $\mathcal{C}$ is a compact $|\mathcal{A}||\mathcal{S}|-1$ dimensional manifold. □

3.6 Mean-Field Approximation to Cooperative MARL

In this section, we provide a complete description of the connections between cooperative MARL and MFC, in terms of the value function approximation and algorithmic approximation under the context of learning.

3.6.1 Value Function Approximation

First we will show that under the Pareto optimality criterion, (MFC) is an approximation to its corresponding cooperative MARL, with an error of $\mathcal{O}(\frac{1}{\sqrt{N}})$.

Recall the admissible policy $\pi = \{\pi_t\}_{t=0}^\infty \in \Pi$. Note that the cooperative MARL in Section 3.2.1 with N identical, indistinguishable, and interchangeable agents becomes

$$\sup_\pi a_N^\pi(\mu^N) := \sup_\pi \frac{1}{N}\sum_{j=1}^N v^{j,\pi}(s^{j,N}, \mu^N) = \sup_\pi \frac{1}{N}\sum_{j=1}^N \mathbb{E}\Big[\sum_{t=0}^\infty \gamma^t \tilde{r}(s_t^{j,N}, \mu_t^N, a_t^{j,N}, \nu_t^N)\Big], \tag{MARL}$$

$$\text{subject to } \; s_{t+1}^{j,N} \sim P(s_t^{j,N}, \mu_t^N, a_t^{j,N}, \nu_t^N), \; a_t^{j,N} \sim \pi_t(s_t^{j,N}, \mu_t^N), \quad 1 \leq j \leq N,$$

with initial conditions $s_0^{j,N} = s^{j,N}$ $(j = 1, 2, \cdots, N)$ and $\mu_0^N(s) = \mu^N(s) := \frac{\sum_{j=1}^N 1(s^{j,N}=s)}{N}$ for $s \in \mathcal{S}$. By symmetry, one can denote $a_N^\pi(\mu^N) := \frac{1}{N}\sum_{j=1}^N v^{j,\pi}(s^{j,N}, \mu^N)$.

Definition 3.6.1 *π^ϵ is ϵ-Pareto optimal for (MARL) if*

$$a_N^{\pi^\epsilon} \geq \sup_\pi a_N^\pi - \epsilon.$$

Assumption 3.6.2 (Continuity of π) *There exists $L_\Pi > 0$ such that for all $s \in \mathcal{S}, \mu_1, \mu_2 \in \mathcal{P}(\mathcal{S})$, and $\pi \in \Pi$,*

$$\|\pi_t(\mu_1, s) - \pi_t(\mu_2, s)\|_1 \leq L_\Pi \|\mu_1 - \mu_2\|_1, \quad \text{for any } t \geq 0.$$

This Lipschitz assumption for admissible policies is commonly used to bridge games in the N-player setting and the mean-field setting [80, 72].

We are now ready to show that the optimal policy for (MFC) is approximately Pareto optimal for (MARL) when $N \to \infty$.

Theorem 3.6.3 (Approximation) *Assume $\gamma \cdot (2L_P + 1)(1 + L_\Pi) < 1$ and Assumptions 3.2.4, 3.2.5 and 3.6.2, then there exists constant $C = C(L_P, L_{\tilde{r}}, L_\Pi, |\mathcal{S}|, |\mathcal{A}|, \tilde{R}, \gamma)$, depending on the dimensions of the state and action spaces in a sublinear order $(\sqrt{|\mathcal{S}|} + \sqrt{|\mathcal{A}|})$, and independent of the number of agents N, such that*

$$\sup_\pi \left| a_N^\pi(\mu^N) - v^\pi(\mu^N) \right| \leq C \frac{1}{\sqrt{N}}, \tag{3.6.1}$$

for any initial condition $s_0^{j,N} = s^j$ $(j = 1, 2, \cdots, N)$ and $\mu^N(s) = \frac{\sum_{j=1}^N 1_{(s^{j,N}=s)}}{N}$ $(s \in \mathcal{S})$. Here v^π and a_N^π are given in (MFC) and (MARL) respectively. Consequently, for any $\epsilon_1 > 0$, there exists an integer $D_{\epsilon_1} \in \mathbb{N}$ such that when $N \geq D_{\epsilon_1}$, any ϵ_2-optimal policy for (MFC) with learning is $(\epsilon_1 + \epsilon_2)$-Pareto optimal for (MARL) with N players.

Corollary 3.6.4 (Optimal value approximation) *Assume the same conditions as in Theorem 3.6.3. Further assume that there exists an optimal policy satisfying Assumption 3.6.2 for (MFC) and (MARL). Denote $\pi^* \in \arg\sup_{\pi \in \Pi} v^\pi$ and $\widetilde{\pi} \in \arg\sup_{\pi \in \Pi} a_N^\pi$, there exists a constant $C = C(L_P, L_{\tilde{r}}, L_\Pi, |\mathcal{S}|, |\mathcal{A}|, \tilde{R}, \gamma)$, depending on the dimensions of the state and action spaces in a sublinear order $(\sqrt{|\mathcal{S}|} + \sqrt{|\mathcal{A}|})$, such that*

$$\left| v^{\pi^*}(\mu^N) - a^{\widetilde{\pi}}(\mu^N) \right| \leq \frac{C}{\sqrt{N}}, \tag{3.6.2}$$

with initial conditions $s_0^{j,N} = s^{j,N}$ and $\mu^N := \frac{\sum_{j=1}^N 1_{(s^{j,N}=s)}}{N}$.

Corollary 3.6.4 follows directly from Theorem 3.6.3 and the proof is deferred to Section 3.8.

Proof of Theorem 3.6.3 First, by (3.2.6)

$$\begin{aligned} a_N^\pi(\mu^N) &= \frac{1}{N} \sum_{j=1}^N \sum_{t=0} \gamma^t \mathbb{E}\big[\tilde{r}(s_t^{j,N}, \mu_t^N, a_t^{j,N}, \nu_t^N)\big] - \frac{1}{N} \sum_{j=1}^N \sum_{t=0} \gamma^t \mathbb{E}\big[\tilde{r}(s_t^{j,N}, \mu_t^N, a_t^{j,N}, \tilde{\nu}_t^N)\big] \\ &\quad + \sum_{t=0}^\infty \gamma^t \mathbb{E}\big[r(\mu_t^N, \pi_t(\mu_t^N))\big], \\ v^\pi(\mu^N) &= \sum_{t=0}^\infty \gamma^t \mathbb{E}[\tilde{r}(s_t, \mu_t, a_t, \nu_t)] = \sum_{t=0}^\infty \gamma^t r(\mu_t, \pi_t(\mu_t)), \end{aligned}$$

where $\tilde{\nu}_t^N(a) := \sum_{s\in\mathcal{S}} \pi_t(\mu,s)(a)\mu_t^N(s) = \frac{1}{N}\sum_{j=1}^N \pi_t(\mu_t^N, s_t^{j,N})(a)$.

By the continuity of r from Lemma 3.2.7 and Assumption 3.2.4,

$$\begin{aligned}
&\sup_\pi \left| a_N^\pi(\mu^N) - v^\pi(\mu^N) \right| \\
\leq\ & (\tilde{R} + 2L_{\tilde{r}}) \sum_{t=0}^\infty \gamma^t \sup_\pi \left(\mathbb{E}\Big[\|\mu_t^{N,\pi} - \mu_t^\pi\|_1 \Big] + \mathbb{E}\Big[\|\pi_t(\mu_t) - \pi_t(\mu_t^N)\|_1 \Big] \right) \\
&+ L_{\tilde{r}} \sum_{t=0}^\infty \gamma^t \sup_\pi \mathbb{E}\Big[\|\nu_t^{N,\pi} - \tilde{\nu}_t^{N,\pi}\|_1 \Big] \\
\leq\ & (\tilde{R} + 2L_{\tilde{r}})(1 + L_\Pi) \sum_{t=0}^\infty \gamma^t \sup_\pi \mathbb{E}\Big[\|\mu_t^{N,\pi} - \mu_t^\pi\|_1 \Big] + L_{\tilde{r}} \sum_{t=0}^\infty \gamma^t \sup_\pi \mathbb{E}\Big[\|\nu_t^{N,\pi} - \tilde{\nu}_t^{N,\pi}\|_1 \Big].
\end{aligned}$$

To prove (3.6.1), it is sufficient to estimate

$$\delta_t^{1,N} := \sup_\pi \mathbb{E}[\|\mu_t^{N,\pi} - \mu_t^\pi\|_1], \quad \delta_t^{2,N} := \sup_\pi \mathbb{E}[\|\nu_t^{N,\pi} - \tilde{\nu}_t^{N,\pi}\|_1.$$

First, we show that $\delta_t^{2,N} = \mathcal{O}(\frac{1}{\sqrt{N}})$. Denote for any $\nu \in \mathcal{P}(\mathcal{A})$ and $f : \mathcal{A} \to \mathbb{R}$, $\nu(f) := \sum_{a\in\mathcal{A}} f(a)\nu(a)$. Then for any $t \geq 0$

$$\begin{aligned}
&\mathbb{E}\left[\left\| \tilde{\nu}_t^N - \nu_t^N \right\|_1 \right] = \mathbb{E}\left[\mathbb{E}\left[\left\| \tilde{\nu}_t^N - \nu_t^N \right\|_1 \middle| s_t^{1,N}, \cdots, s_t^{N,N} \right] \right] \qquad (3.6.3) \\
=\ & \mathbb{E}\left[\mathbb{E}\left[\sup_{f:\mathcal{A}\to\{-1,1\}} \left(\tilde{\nu}_t^N(f) - \nu_t^N(f) \right) \middle| s_t^{1,N}, \cdots, s_t^{N,N} \right] \right] \\
=\ & \mathbb{E}\left[\mathbb{E}\left[\sup_{f:\mathcal{A}\to\{-1,1\}} \frac{1}{N}\sum_{j=1}^N \sum_{a\in\mathcal{A}} \pi_t(\mu_t^N, s_t^{j,N})(a) f(a) - \frac{1}{N}\sum_{j=1}^N f(a_t^{j,N}) \middle| s_t^{1,N}, \cdots, s_t^{N,N} \right] \right],
\end{aligned}$$

where the first equality is by law of total expectation and the last equality is by the definitions of $\tilde{\nu}_t^N$ and ν_t^N. Now consider a fixed $f : \mathcal{A} \to \{-1, 1\}$. Conditioned on $s_t^{1,N}, \cdots, s_t^{N,N}$, $\{a_t^{j,N}\}_{j=1}^N$ is a sequence of independent random variables with $a_t^{j,N} \sim \pi_t(\mu_t^N, s_t^{j,N})(\cdot)$. Therefore, conditioned on $s_t^{1,N}, \cdots, s_t^{N,N}$,

$$\left\{ \sum_{a\in\mathcal{A}} \pi_t(\mu_t^N, s_t^{j,N})(a) f(a) - f(a_t^{j,N}) \right\}_{j=1}^N$$

is a sequence of independent mean-zero random variables bounded in $[-2, 2]$. The boundedness further implies that each

$$\sum_{a\in\mathcal{A}} \pi_t(\mu_t^N, s_t^{j,N})(a) f(a) - f(a_t^{j,N})$$

is a sub-Gaussian random variable with variance bounded by 4. (See Chapter 2 of [177] for the general introduction to sub-Gaussian random variables.) Meanwhile, the independence implies that conditioned on $s_t^{1,N}, \cdots, s_t^{N,N}$,

$$\frac{1}{N}\sum_{j=1}^{N}\sum_{a\in\mathcal{A}}\pi_t(\mu_t^N, s_t^{j,N})(a)f(a) - \frac{1}{N}\sum_{j=1}^{N} f(a_t^{j,N})$$

is a mean-zero sub-Gaussian random variable with variance $\frac{4}{N}$. In general, for a sequence of mean-zero sub-Gaussian random variables $\{X_i\}_{i=1}^M$ with parameter σ^2, by Eqn.(2.66) in [177], we have

$$\mathbb{E}\left[\sup_{i=1,\cdots,M} X_i\right] \leq \sqrt{2\sigma^2 \ln(M)}.$$

Therefore, conditioned on $s_t^{1,N}, \cdots, s_t^{N,N}$,

$$\mathbb{E}\left[\sup_{f:\mathcal{A}\to\{-1,1\}} \frac{1}{N}\sum_{j=1}^{N}\sum_{a\in\mathcal{A}}\pi_t(\mu_t^N, s_t^{j,N})(a)f(a) - \frac{1}{N}\sum_{j=1}^{N} f(a_t^{j,N})\middle| s_t^{1,N}, \cdots, s_t^{N,N}\right]$$
$$\leq \sqrt{8\ln(2)|\mathcal{A}|/N}$$

holds since we have in total $2^{|\mathcal{A}|}$ different choices for $f : \mathcal{A} \to \{-1, 1\}$ when taking the supremum. Thus, following (3.6.3), we have

$$\delta_t^{2,N} = \sup_{\pi} \mathbb{E}\left[\left\|\tilde{\nu}_t^N - \nu_t^N\right\|_1\right] \leq \sqrt{8\ln(2)|\mathcal{A}|/N}. \tag{3.6.4}$$

Second, we estimate $\delta_t^{1,N}$ and claim that $\delta_t^{1,N} = \mathcal{O}(\frac{1}{\sqrt{N}})$. This is done by induction. The claim holds for $t = 0$ because $\delta_0^{1,N} = 0$. Suppose the claim holds for t and consider $t + 1$.

Given $s_t^{1,N}, \cdots, s_t^{N,N}$, $\mu_t^N = \frac{1}{N}\sum_{j=1}^N \delta_{s_t^{j,N}}$ and policy $\pi_t(\mu_t^N)$ at time t, for any $\nu \in \mathcal{P}(\mathcal{A})$, let $\mu_t^N P_{\mu_t^N,\nu}$ denote a $\mathcal{P}(\mathcal{S})$-valued random variable, with

$$\mu_t^N P_{\mu_t^N,\nu}(s) := \frac{1}{N}\sum_{j=1}^{N} P(s_t^{j,N}, \mu_t^N, a_t^{j,N}, \nu)(s), \quad a_t^{j,N} \sim \pi_t(\mu_t^N, s_t^{j,N}).$$

We consider the following decomposition,

$$\mathbb{E}\left[\|\mu_{t+1}^N - \mu_{t+1}\|_1\right] \leq \underbrace{\mathbb{E}\left[\left\|\mu_{t+1}^N - \mu_t^N P_{\mu_t^N,\nu_t^N}\right\|_1\right]}_{(I)} + \underbrace{\mathbb{E}\left[\left\|\mu_t^N P_{\mu_t^N,\nu_t^N} - \mu_t^N P_{\mu_t^N,\tilde{\nu}_t^N}\right\|_1\right]}_{(II)}$$
$$+ \underbrace{\mathbb{E}\left[\left\|\mu_t^N P_{\mu_t^N,\tilde{\nu}_t^N} - \Phi(\mu_t^N, \pi_t(\mu_t^N))\right\|_1\right]}_{(III)} + \underbrace{\mathbb{E}\left[\left\|\Phi(\mu_t^N, \pi_t(\mu_t^N)) - \mu_{t+1}\right\|_1\right]}_{(IV)}. \tag{3.6.5}$$

Bounding (I) in RHS of (3.6.5): We proceed the similar argument as (3.6.3),

$$\begin{aligned}
&\mathbb{E}\left[\left\|\mu_{t+1}^N - \mu_t^N P_{\mu_t^N,\nu_t^N}\right\|_1\right] \\
=\ &\mathbb{E}\left[\mathbb{E}\left[\left\|\mu_{t+1}^N - \mu_t^N P_{\mu_t^N,\nu_t^N}\right\|_1 \middle| s_t^{1,N},\cdots,s_t^{N,N},a_t^{1,N},\cdots,a_t^{N,N}\right]\right] \\
=\ &\mathbb{E}\Bigg[\mathbb{E}\Bigg[\sup_{f:\mathcal{S}\to\{-1,1\}}\frac{1}{N}\sum_{j=1}^N f(s_{t+1}^{j,N}) \\
&\qquad -\frac{1}{N}\sum_{j=1}^N\sum_{s\in\mathcal{S}} P(s_t^{j,N},\mu_t^N,a_t^{j,N},\nu_t^N)(s)f(s)\Bigg| s_t^{1,N},\cdots,s_t^{N,N},a_t^{1,N},\cdots,a_t^{N,N}\Bigg]\Bigg] \\
\le\ &\sqrt{8\ln(2)|\mathcal{S}|/N}.
\end{aligned}$$

Bounding (II) in RHS of (3.6.5):

$$\begin{aligned}
&\mathbb{E}\left[\left\|\mu_t^N P_{\mu_t^N,\nu_t^N} - \mu_t^N P_{\mu_t^N,\tilde{\nu}_t^N}\right\|_1\right] \\
=\ &\mathbb{E}\left[\left\|\frac{1}{N}\sum_{j=1}^N P(s_t^{j,N},\mu_t^N,a_t^{j,N},\nu_t^N) - \frac{1}{N}\sum_{j=1}^N P(s_t^{j,N},\mu_t^N,a_t^{j,N},\tilde{\nu}_t^N)\right\|_1\right] \\
\le\ &\frac{1}{N}\sum_{j=1}^N \mathbb{E}\left[\left\|P(s_t^{j,N},\mu_t^N,a_t^{j,N},\nu_t^N) - P(s_t^{j,N},\mu_t^N,a_t^{j,N},\tilde{\nu}_t^N)\right\|_1\right] \\
\le\ &L_P\cdot\mathbb{E}\left[\left\|\nu_t^N-\tilde{\nu}_t^N\right\|_1\right] \le L_P\sqrt{8\ln(2)|\mathcal{A}|/N},
\end{aligned}$$

in which the second last inequality holds by the Lipschitz property from Assumption 3.2.5 and the last inequality holds by (3.6.4).

Bounding (III) in RHS of (3.6.5):

$$\begin{aligned}
&\mathbb{E}\left[\left\|\mu_t^N P_{\mu_t^N,\tilde{\nu}_t^N} - \Phi(\mu_t^N,\pi_t(\mu_t^N))\right\|_1\right] \\
=\ &\mathbb{E}\Bigg[\mathbb{E}\Bigg[\sup_{g:\mathcal{S}\to\{-1,1\}}\frac{1}{N}\sum_{j=1}^N\sum_{s\in\mathcal{S}} P(s_t^{j,N},\mu_t^N,a_t^{j,N},\tilde{\nu}_t^N)(s)g(s) \\
&\quad -\frac{1}{N}\sum_{j=1}^N\sum_{a\in\mathcal{A}}\sum_{s\in\mathcal{S}} P(s_t^{j,N},\mu_t^N,a,\tilde{\nu}_t^N)(s)\pi_t(\mu_t^N,s_t^{j,N})(a)g(s)\Bigg| s_t^{1,N},\ldots,s_t^{N,N}\Bigg]\Bigg]
\end{aligned}$$

For a fixed $g:\mathcal{S}\to\{-1,1\}$, conditioned on $s_t^{1,N},\cdots,s_t^{N,N}$,

$$\left\{\sum_{s\in\mathcal{S}} P(s_t^{j,N},\mu_t^N,a_t^{j,N},\tilde{\nu}_t^N)(s)g(s) - \sum_{a\in\mathcal{A}}\sum_{s\in\mathcal{S}} P(s_t^{j,N},\mu_t^N,a,\tilde{\nu}_t^N)(s)\pi_t(\mu_t^N,s_t^{j,N})(a)g(s)\right\}_{j=1}^N$$

are independent mean-zero sub-Gaussian random variables. Meanwhile, since by definition, we have for each $j = 1, \cdots, N$, $\sum_{s\in\mathcal{S}} P(s_t^{j,N}, \mu_t^N, a_t^{j,N}, \tilde{\nu}_t^N)(s) = 1$, it is easy to show that $\sum_{s\in\mathcal{S}} P(s_t^{j,N}, \mu_t^N, a_t^{j,N}, \tilde{\nu}_t^N)(s)g(s)$ is bounded by $[-1, 1]$. Therefore, using the same argument applied in the proof of (3.6.4), we can show that

$$\mathbb{E}\left[\left\|\mu_t^N P_{\mu_t^N,\tilde{\nu}_t^N} - \Phi(\mu_t^N, \pi_t(\mu_t^N))\right\|_1\right] \leq \sqrt{8\ln(2)|\mathcal{S}|/N}.$$

Bounding (IV) in RHS of (3.6.5):

$$\begin{aligned}\mathbb{E}\big[\|\Phi(\mu_t^N, \pi_t(\mu_t^N)) - \mu_{t+1}\|_1\big] &= \mathbb{E}\big[\|\Phi(\mu_t^N, \pi_t(\mu_t^N)) - \Phi(\mu_t, \pi_t(\mu_t))\|_1\big] \\ &\leq (2L_P + 1)(1 + L_\Pi)\mathbb{E}\big[\|\mu_t^N - \mu_t\|_1\big],\end{aligned}$$

where the first equality is from the flow of probability measure $\mu_{t+1} = \Phi(\mu_t, \pi_t(\mu_t))$ by Lemma 3.2.2, and the first inequality is by the continuity of Φ from Lemma 3.2.8.

By taking supremum over π on both sides of (3.6.5), we have $\delta_{t+1}^{1,N} \leq (2L_P + 1)(1 + L_\Pi)\delta_t^{1,N} + (L_P\sqrt{|\mathcal{A}|} + 2\sqrt{|\mathcal{S}|})\sqrt{8\ln(2)/N}$, hence $\delta_t^{1,N} \leq \frac{(L_P\sqrt{|\mathcal{A}|}+2\sqrt{|\mathcal{S}|})}{(2L_P+1)(1+L_\Pi)-1}\Big((2L_P+1)^t(1+L_\Pi)^t - 1\Big)\sqrt{8\ln(2)/N}$. Therefore

$$\begin{aligned}&\sup_\pi \left|a_N^\pi(\mu^N) - v^\pi(\mu^N)\right| \leq (\tilde{R} + 2L_{\tilde{r}})\sum_{t=0}^{\infty}\gamma^t\delta_t^{1,N} + L_{\tilde{r}}\sum_{t=0}^{\infty}\gamma^t\delta_t^{2,N} \\ \leq\ &\left\{\frac{(\tilde{R} + 2L_{\tilde{r}})(L_P\sqrt{|\mathcal{A}|} + 2\sqrt{|\mathcal{S}|})}{(2L_P + 1)(1 + L_\Pi) - 1}\Big(\frac{1}{1 - (2L_P + 1)(1 + L_\Pi)\gamma} - \frac{1}{1 - \gamma}\Big)\right. \\ &\qquad \left. + \frac{\sqrt{|\mathcal{A}|}L_{\tilde{r}}}{1 - \gamma}\right\}\sqrt{8\ln(2)/N}.\end{aligned}$$

This proves (3.6.1). □

3.6.2 Q-function Approximation under Learning

In this section we show that, with $\mathcal{O}(\log(1/\epsilon))$ samples and with ϵ the size of ϵ-set, the kernel-based Q-function from Algorithm 3 provides an approximation to the Q-function of cooperative MARL, with an error of $\mathcal{O}(\epsilon + \frac{1}{\sqrt{N}})$,

For the (MARL) problem specified in Section 3.6.1 and given the initial states $s^{j,N}$ and actions $a^{j,N}$ from all agents $(j = 1, 2, \ldots, N)$, let us define the corresponding Q-function,

$$Q_N^\pi(\mu^N, h^N) = \frac{1}{N}\sum_{j=1}^{N}\tilde{r}(s^{j,N}, \mu^N, a^{j,N}, \nu^N) + \frac{1}{N}\sum_{j=1}^{N}\mathbb{E}\Big[\sum_{t=1}^{\infty}\gamma^t\tilde{r}(s_t^{j,N}, \mu_t^N, a_t^{j,N}, \nu_t^N)\Big] \quad (3.6.6)$$

subject to

$$s_1^{j,N} \sim P(s^{j,N}, \mu^N, a^{j,N}, \nu^N),$$
$$s_{t+1}^{j,N} \sim P(s_t^{j,N}, \mu_t^N, a_t^{j,N}, \nu_t^N),\ a_t^{j,N} \sim \pi(\mu_t^N, s_t^{j,N}), \quad 1 \le j \le N, \text{ and } t \ge 1.$$

where $\mu^N(s) = \frac{\sum_{j=1}^N 1(s^{j,N}=s)}{N}$, $\nu^N(a) = \frac{\sum_{j=1}^N 1(a^{j,N}=a)}{N}$ and $h^N(s)(a) = \frac{\sum_{j=1}^N 1(s^{j,N}=s;\, a^{j,N}=a)}{\sum_{j=1}^N 1(s^{j,N}=s)}$ with the convention $\frac{0}{0} = 0$, and define

$$Q_N(\mu^N, h^N) = \sup_\pi Q_N^\pi(\mu^N, h^N). \tag{3.6.7}$$

Theorem 3.6.1 *Fix $\epsilon > 0$. Assume the same conditions as in Theorem 3.5.5, Theorem 3.6.3 and Corollary 3.6.4. Then there exists some $\widetilde{C} = \widetilde{C}(L_P, L_\Pi, |\mathcal{S}|, |\mathcal{A}|, \tilde{R}, L_{\tilde{r}}, \gamma) > 0$, depending on the dimensions of the state and action spaces in a sublinear order $(\sqrt{|\mathcal{S}|} + \sqrt{|\mathcal{A}|})$, such that*

$$||Q_N - \Gamma_K \widehat{Q}_\epsilon||_\infty \le \frac{L_r + 2\gamma N_K L_K V_{\max} L_\Phi}{1-\gamma} \cdot \epsilon + \frac{2L_r}{(1-\gamma L_\Phi)(1-\gamma)} \cdot \epsilon + \frac{\widetilde{C}}{\sqrt{N}}. \tag{3.6.8}$$

Combining Theorem 3.5.6 and Theorem 3.6.1 implies the following: fix any $\epsilon > 0$, there exists an integer $D_\epsilon \in \mathbb{N}$ such that Algorithm 3 outputs a kernel-based Q-function with $C\log(1/\epsilon)$ samples. With high probability, this kernel-based Q-function is ϵ close to the Q-function of MARL when the agent number $N > D_\epsilon$. Here $C = C(L_P, L_\Pi, |\mathcal{S}|, |\mathcal{A}|, \tilde{R}, L_{\tilde{r}}, \gamma)$ is sublinear with respect to $|\mathcal{S}|$ and $|\mathcal{A}|$ and independent of the number of agents N.

Proof of Theorem 3.6.1 First we have

$$Q_N(\mu^N, h^N) = \frac{1}{N}\sum_{j=1}^N \tilde{r}(s^{j,N}, \mu^N, a^{j,N}, \nu^N) + \gamma \sup_\pi \mathbb{E}[a_N^\pi(\mu_1^N)] \tag{3.6.9}$$

On the other hand, by the definitions of Q in (3.3.1), μ^N and h^N,

$$\begin{aligned} Q(\mu^N, h^N) &= \frac{1}{N}\sum_{j=1}^N \tilde{r}(s^{j,N}, \mu^N, a^{j,N}, \nu^N) + \sup_{\pi\in\Pi}\left[\sum_{t=1}^\infty \gamma^t r(\mu_t, h_t)\middle|\mu_1 = \Phi(\mu^N, h^N)\right] \\ &= \frac{1}{N}\sum_{j=1}^N \tilde{r}(s^{j,N}, \mu^N, a^{j,N}, \nu^N) + \gamma \sup_{\pi\in\Pi} v^\pi(\Phi(\mu^N, h^N)) \end{aligned} \tag{3.6.10}$$

with $h_t = \pi_t(\mu_t)$. Therefore,

$$\begin{aligned} &|Q(\mu^N, h^N) - Q_N(\mu^N, h^N)| = \gamma\left|\sup_{\pi\in\Pi} v^\pi(\Phi(\mu^N, h^N)) - \sup_\pi \mathbb{E}[a_N^\pi(\mu_1^N)]\right| \\ \le\ & \gamma\left|v^{\pi^*}(\Phi(\mu^N, h^N)) - \mathbb{E}[v^{\pi^*}(\mu_1^N)]\right| + \gamma\left|\mathbb{E}\left[v^{\pi^*}(\mu_1^N) - a_N^{\tilde{\pi}}(\mu_1^N)\right]\right| \end{aligned} \tag{3.6.11}$$

where $\pi^* \in \arg\sup_{\pi\in\Pi} v^\pi$, $\widetilde{\pi} \in \arg\sup_{\pi\in\Pi} a_N^\pi$, and the expectation in (3.6.11) is taking with respect to μ_1^N.

For the second term in (3.6.11),

$$\left|\mathbb{E}\left[v^{\pi^*}(\mu_1^N) - a_N^{\widetilde{\pi}}(\mu_1^N)\right]\right| \leq \mathbb{E}\left[\left|v^{\pi^*}(\mu_1^N) - a_N^{\widetilde{\pi}}(\mu_1^N)\right|\right] \leq \frac{C}{\sqrt{N}}, \tag{3.6.12}$$

in which the first inequality holds by convexity and the second inequality holds due to Corollary 3.6.4.

For the first term in (3.6.11),

$$\begin{aligned} &\left|v^{\pi^*}(\Phi(\mu^N, h^N)) - \mathbb{E}_{\mu_1^N}[v^{\pi^*}(\mu_1^N)]\right| \\ \leq\ & (\tilde{R} + 2L_{\tilde{r}}) \sum_{t=0}^{\infty} \gamma^t \mathbb{E}\left[\left\|\mu_t^{\pi^*} - \overline{\mu}_t^{\pi^*}\right\|_1\right] + L_{\tilde{r}} \sum_{t=0}^{\infty} \gamma^t \mathbb{E}\left[\left\|\nu_t^{\pi^*} - \overline{\nu}_t^{\pi^*}\right\|_1\right] \qquad (3.6.13) \\ \leq\ & (\tilde{R} + 3L_{\tilde{r}} + L_{\tilde{r}} L_\Pi) \sum_{t=0}^{\infty} \gamma^t \mathbb{E}\left[\left\|\mu_t^{\pi^*} - \overline{\mu}_t^{\pi^*}\right\|_1\right], \qquad (3.6.14) \end{aligned}$$

in which

$$\mu_{t+1}^{\pi^*} = \Phi(\mu_t^{\pi^*}, \pi^*(\mu_t^{\pi^*}))$$

with initial condition $\mu_0^{\pi^*} = \Phi(\mu^N, h^N)$, and

$$\overline{\mu}_{t+1}^{\pi^*} = \Phi(\overline{\mu}_t^{\pi^*}, \pi^*(\overline{\mu}_t^{\pi^*}))$$

with initial condition $\overline{\mu}_0^{\pi^*} = \mu_1^N$. In addition,

$$\nu_t^{\pi^*}(a) = \sum_{s\in\mathcal{S}} \pi^*(\mu_t^{\pi^*}, s)(a)\mu_t^{\pi^*}(s), \quad \overline{\nu}_t^{\pi^*}(a) = \sum_{s\in\mathcal{S}} \pi^*(\overline{\mu}_t^{\pi^*}, s)(a)\overline{\mu}_t^{\pi^*}(s).$$

(3.6.13) holds by the continuity of r from Lemma 3.2.7 and Assumption 3.2.4. (3.6.14) holds since by Lemma 3.2.6 and Assumption 3.6.2,

$$\left\|\nu_t^{\pi^*} - \overline{\nu}_t^{\pi^*}\right\|_1 \leq \left\|\mu_t^{\pi^*} - \overline{\mu}_t^{\pi^*}\right\|_1 + \max_{s\in\mathcal{S}} \left\|\pi^*(\mu_t^{\pi^*}, s) - \pi^*(\overline{\mu}_t^{\pi^*}, s)\right\|_1 \leq (1 + L_\Pi)\left\|\mu_t^{\pi^*} - \overline{\mu}_t^{\pi^*}\right\|_1.$$

For $t = 0$,

$$\begin{aligned} &\mathbb{E}\left[\left\|\mu_0^{\pi^*} - \overline{\mu}_0^{\pi^*}\right\|_1\right] = \mathbb{E}\left[\left\|\mu_1^N - \Phi(\mu^N, h^N)\right\|_1\right] \qquad (3.6.15) \\ =\ & \mathbb{E}\left[\left\|\mu_1^N - \frac{1}{N}\sum_{j=1}^{N}\sum_{a\in\mathcal{A}} P(s^{j,N}, \mu^N, a, \nu^N)(s) h^N(s^{j,N})(a)\right\|_1\right] \\ =\ & \mathbb{E}\left[\sup_{g:\mathcal{S}\to\{-1,1\}} \frac{1}{N}\sum_{j=1}^{N} g(s_1^{j,N}) - \frac{1}{N}\sum_{j=1}^{N}\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} P(s^{j,N}, \mu^N, a, \nu^N)(s) h^N(s^{j,N})(a) g(s)\right] \\ \leq\ & \sqrt{8|\mathcal{S}|\ln(2)/N}, \end{aligned}$$

where the second equality is by $\nu^N(a) = \sum_{s\in\mathcal{S}} \mu^N(s)h^N(s)(a)$ and by the definition of Φ, and in the last inequality,

$$\{g(s_1^{j,N}) - \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} P(s^{j,N},\mu^N,a,\nu^N)(s)h^N(s^{j,N})(a)g(s)\}_{j=1}^N$$

are independent mean-zero sub-Gaussian random variables bounded by $[-2,2]$ and thus we proceed the similar arguments as (3.6.3).

We now prove by induction it holds for all $t \geq 0$ that

$$\mathbb{E}\left[\|\mu_t^{\pi^*} - \overline{\mu}_t^{\pi^*}\|_1\right] \leq ((2L_P+1)(L_\Pi+1))^t\sqrt{8|\mathcal{S}|\ln(2)/N}. \tag{3.6.16}$$

(3.6.16) holds when $t = 0$ given (3.6.15). Now assume (3.6.16) holds for $t \leq s$. When $t = s+1$, we have

$$\begin{aligned}
\mathbb{E}\left[\|\mu_{s+1}^{\pi^*} - \overline{\mu}_{s+1}^{\pi^*}\|_1\right] &= \mathbb{E}\left[\left\|\Phi(\mu_s^{\pi^*},\pi^*(\mu_s^{\pi^*})) - \Phi(\overline{\mu}_s^{\pi^*},\pi^*(\overline{\mu}_s^{\pi^*}))\right\|_1\right] \\
&\leq (2L_P+1)d_{\mathcal{C}}\left(\left(\mu_s^{\pi^*},\pi^*(\mu_s^{\pi^*})\right),\left(\overline{\mu}_s^{\pi^*},\pi^*(\overline{\mu}_s^{\pi^*})\right)\right) \\
&= (2L_P+1)\left(\|\mu_s^{\pi^*} - \overline{\mu}_s^{\pi^*})\|_1 + \|\pi^*(\mu_s^{\pi^*}) - \pi^*(\overline{\mu}_s^{\pi^*})\|_1\right) \\
&\leq (2L_P+1)(1+L_\Pi)\|\mu_s^{\pi^*} - \overline{\mu}_s^{\pi^*}\|_1 \\
&\leq ((2L_P+1)(1+L_\Pi))^{s+1}\sqrt{8|\mathcal{S}|\ln(2)/N},
\end{aligned} \tag{3.6.17}$$

where the first inequality holds by Lemma 3.2.8 and the second inequality holds by Assumption 3.6.2, and the third inequality holds by induction. Finally when $(2L_P+1)(1+L_\Pi)\gamma < 1$,

$$\begin{aligned}
(3.6.14) &\leq (\tilde{R} + 3L_{\tilde{r}} + L_{\tilde{r}}L_\Pi)\sum_{t=0}^{\infty}\sqrt{8|\mathcal{S}|\ln(2)|/N}((2L_P+1)(1+L_\Pi)\gamma)^t \qquad (3.6.18)\\
&= \sqrt{8|\mathcal{S}|\ln(2)|/N}\frac{\tilde{R} + 3L_{\tilde{r}} + L_{\tilde{r}}L_\Pi}{1-(2L_P+1)(1+L_\Pi)\gamma}.
\end{aligned}$$

Therefore, combining (3.6.11), (3.6.12) and (3.6.18), we have proven that there exists some $\widetilde{C} = \widetilde{C}(L_P, L_\Pi, |\mathcal{S}|, |\mathcal{A}|, \tilde{R}, L_{\tilde{r}}, \gamma) > 0$ such that $\|Q - Q_N\|_\infty \leq \frac{\widetilde{C}}{\sqrt{N}}$. Here $\widetilde{C}$ depends on the dimensions of the state and action spaces in a sublinear order $(\sqrt{|\mathcal{S}|} + \sqrt{|\mathcal{A}|})$ and is independent of the number of agents N. Theorem 3.6.1 follows from combining the result above with Theorem 3.5.5. □

3.7 Experiments

We will test the MFC-K-Q algorithm on a network traffic congestion control problem. In the network there are senders and receivers. Multiple senders share a single communication

link which has an unknown and limited bandwidth. When the total sending rates from these senders exceed the shared bandwidth, packages may be lost. Sender streams data packets to the receiver and receives feedback from the receiver on success or failure in the form of packet acknowledgements (ACKs). (See Figure 3.1 for illustration and [83] for a similar set-up). The control problem for each sender is to send the packets as fast as possible and with the risk of packet loss as little as possible. Given a large interactive population of senders, the exact dynamics of the system and the rewards are unknown, thus it is natural to formulate this control problem in the framework of learning MFC.

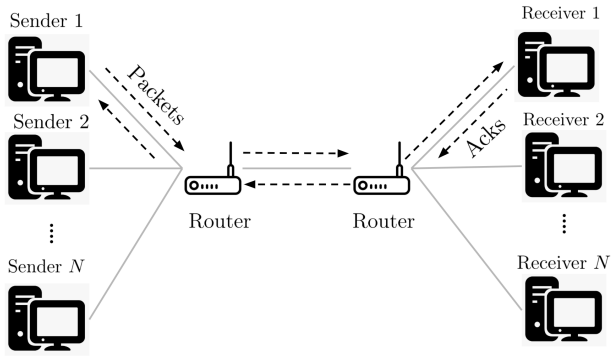


Figure 3.1: Illustration of the network traffic congestion control problem. Multiple network traffic flows share the same link with a limited bandwidth.

3.7.1 Set-up

States. For a representative agent in MFC problem with learning, at the beginning of each round t, the state s_t is her inventory (current unsent packet units) taking values from $\mathcal{S} = \{0, \dots, |\mathcal{S}| - 1\}$. Denote $\mu_t := \{\mu_t(s)\}_{s\in\mathcal{S}}$ as the population state distribution over $\mathcal{S}$.

Actions. The action is the sending rate. At the beginning of each round t, the agent can adjust her sending rate a_t, which remains fixed in $[t, t+1)$. Here we assume $a_t \in \mathcal{A} = \{0, \dots, |\mathcal{A}| - 1\}$. Denote $h_t = \{h_t(s)(a)\}_{s\in\mathcal{S},a\in\mathcal{A}}$ as the policy from the central controller.

Limited bandwidth and packet loss. A system with N agents has a shared link of unknown bandwidth cN ($c > 0$). In the mean-field limit with $N \to \infty$,

$$F_t = \sum_{s\in\mathcal{S},a\in\mathcal{A}} u h_t(s)(a)\mu_t(s)$$

is the average sending rate at time t. If $F_t > c$, with probability $\frac{(F_t - c)}{F_t}$, each agent's packet will be lost.

MFC dynamics. At time $t+1$, the state of the representative agent moves from s_t to $s_t - a_t$. Overshooting is not allowed: $a_t \leq s_t$. Meanwhile, at the end of each round, there are some packets added to each agent's packet sending queue. The packet fulfillment consists of two scenarios. First a lost package will be added to the original queue. Then once the inventory hits zero, a random fulfillment with uniform distribution Unif($\mathcal{S}$) will be added to her queue. That is, $s_{t+1} = s_t - a_t + a_t 1_t(L) + (1 - 1_t(L)1(a_t = s_t) \cdot a_t$, where $1_t(L) = 1$(packet is lost in round t), with 1 an indicator function and $a_t \sim$ Unif($\mathcal{S}$).

Evolution of population state distribution μ_t. Define, for $s \in \mathcal{S}$,

$$\tilde{\mu}_t(s) = \sum_{s' \geq x} \mu_t(s') h_t(s')(s' - x) \left(1 - 1(F_t > c)\tfrac{F_t - c}{F_t}\right) + \mu_t(s) 1(F_t > c)\tfrac{F_t - c}{F_t}.$$

Then $\tilde{\mu}_t$ represents the state of the population distribution after the first step of task fulfillment and before the second step of task fulfillment. Finally, for $s \in \mathcal{S}$, $\mu_{t+1}(s) = \left(\tilde{\mu}_t(s) + \frac{\tilde{\mu}_t(0)}{|\mathcal{S}|}\right) 1(x \neq 0) + \frac{\tilde{\mu}_t(0)}{|\mathcal{S}|} 1(x = 0)$, describes the transition of the flows $\mu_{t+1} = \Phi(\mu_t, h_t)$.

Rewards. Consistent with [51] and [83], the reward function depending on throughput, latency, with loss penalty is defined as $\tilde{r} = a * \text{throughput} - b * \text{latency}^2 - d * \text{loss}$, with $a, b, d \geq 0$.

3.7.2 Performance of MFC-K-Q Algorithm

We first test the convergence property and performance of MFC-K-Q (Algorithm 3) for this traffic control problem with different kernel choices and with varying N. We then compare MFC-K-Q with MFQ Algorithm [36] on MFC, Deep PPQ [83], and PCC-VIVACE [51] on MARL.

We assume the access to an MFC simulator $\mathcal{G}(\mu, h) = (\mu', r)$. That is, for any pair $(\mu, h) \in \mathcal{C}$, we can sample the aggregated population reward r and the next population state distribution μ' under policy h. We sample $\mathcal{G}(\mu, h) = (\mu', r)$ once for all $(\mu, h) \in \mathcal{C}_\epsilon$. In each outer iteration, each update on $(\mu, h) \in \mathcal{C}_\epsilon$ is one inner-iteration. Therefore, the total number of inner iterations within each outer iteration equals $|\mathcal{C}_\epsilon|$.

Applying MFC policy to N-agent game. To measure the performance of the MFC policy π for an N-agent set-up, we apply π to the empirical state distribution of N agents.

Performance criteria. We assume the access to an N-agent simulator $\mathcal{G}^N(\boldsymbol{s}, \boldsymbol{a}) = (\boldsymbol{s}', \boldsymbol{r})$. That is, if agents take joint action $\boldsymbol{a}$ from state $\boldsymbol{s}$, we can observe the joint reward $\boldsymbol{r}$ and the next joint state $\boldsymbol{s}'$. We evaluate different policies in the N-agent environment.

We randomly sample K initial states $\{\boldsymbol{s}_0^k \in \mathcal{S}^N\}_{k=1}^K$ and apply policy π to each initial state $\boldsymbol{s}_0^k$ and collect the continuum rewards in each path for T_0 rounds $\{\bar{r}_{k,t}^\pi\}_{t=1}^{T_0}$. Here $\bar{r}_{k,t}^\pi = \frac{\sum_{i=1}^N r_k^{\pi,i}}{N}$

is the average reward from N agents in round t under policy π. Then

$$R_N^\pi(\boldsymbol{s}_0^k) := \sum_{t=1}^{T_0} \gamma^t \bar{r}_{k,t}^\pi$$

is used to approximate the value function V_C^π with policy π, when T_0 is large.

Two performance criteria are used: the first one

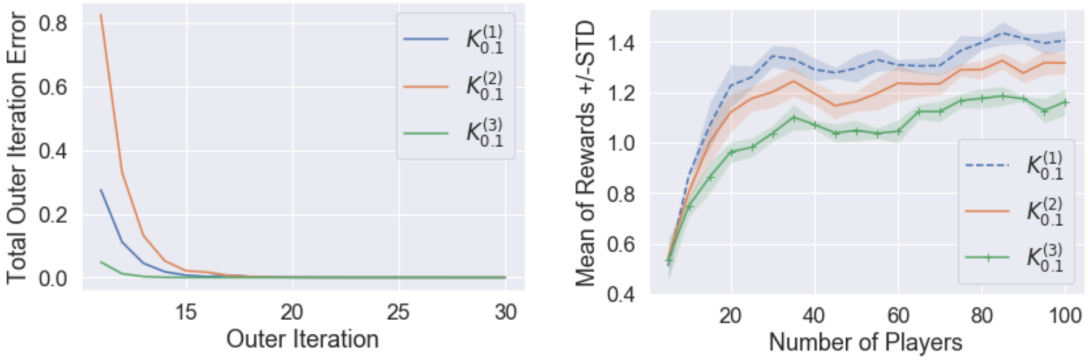
$$C_N^{(1)}(\pi) = \frac{1}{K} \sum_{k=1}^{K} R_N^\pi(\boldsymbol{s}_0^k)$$

measures the average reward from policy π; and the second criterion

$$C_N^{(2)}(\pi^1, \pi^2) = \frac{1}{K} \sum_{k=1}^{K} \frac{R_N^{\pi_1}(\boldsymbol{s}_0^k) - R_N^{\pi_2}(\boldsymbol{s}_0^k)}{R_N^{\pi_1}(\boldsymbol{s}_0^k)}$$

measures the relative improvements of using policy π^1 instead of policy π^2.

Experiment set-up. We set $\gamma = 0.5$, $a = 30$, $b = 10$, $d = 50$, $c = 0.4$, $M = 2$, $K = 500$ and $T_0 = 30$, and compare policies with $N = 5n$ agents ($n = 1, 2, \cdots, 20$). For the ϵ-net, we take uniform grids with ϵ distance between adjacent points on the net. The confidence intervals are calculated with 20 repeated experiments.



(a) Convergence of Q-function.

(b) $C_N^{(1)}$: Average reward.

Figure 3.2: Performance of MFC-K-Q under three different kernels (3.7.1) - (3.7.3). Figure 3.2a shows that all kernels lead to the convergence of Q-functions within 15 outer iterations. Figure 3.2b compares the performance of learned policies from different choices of kernels, with different number of agents.

Results with different kernels. We use the following kernels with hyper-parameter ϵ: triangular, (truncated) Gaussian, and (truncated) constant kernels. That is,

$$\phi_\epsilon^{(1)}(x,y) = \mathbf{1}_{\{\|x-y\|_2\le\epsilon\}}\big|\epsilon - \|x-y\|_2\big|, \tag{3.7.1}$$

$$\phi_\epsilon^{(2)}(x,y) = \mathbf{1}_{\{\|x-y\|_2\le\epsilon\}}\frac{1}{\sqrt{2\pi}}\exp(-|\epsilon - \|x-y\|_2|^2), \tag{3.7.2}$$

and

$$\phi_\epsilon^{(3)}(x,y) = \mathbf{1}_{\{\|x-y\|_2\le\epsilon\}}. \tag{3.7.3}$$

We run the experiments for

$$K_\epsilon^{(j)}(c^i,c) = \frac{\phi_\epsilon^{(j)}(c^i,c)}{\sum_{i=1}^{N_\epsilon}\phi_\epsilon^{(j)}(c^i,c)},$$

with $j=1,2,3$ and $\epsilon = 0.1$.

All kernels lead to the convergence of Q-functions within 15 outer iterations (Figure 3.2a). When $N \le 10$, the performances of all kernels are similar since ϵ-net is accurate for games with $N = \frac{1}{\epsilon}$ agents. When $N \ge 15$, $K_{0.1}^{(1)}$ performs the best and $K_{0.1}^{(3)}$ does the worst (Figure 3.2b): treating all nearby ϵ-net points with equal weights yields relatively poor performance.

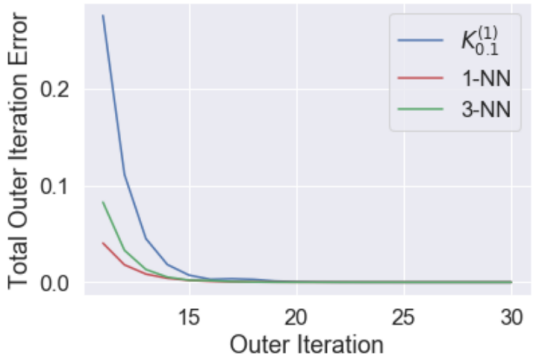
Further comparison of $K_{0.1}^{(j)}$'s suggests that appropriate choices of kernels for specific problems with particular structures of Q-functions help reducing errors from a fixed ϵ-net.

Results with different k-nearest neighbors. We compare kernel $K_{0.1}^{(1)}(x,y)$ with the k-nearest-neighbor (k-NN) method ($k=1,3$), with 1-NN the projection approach by which each point is projected onto the closest point in $\mathcal{C}_\epsilon$, a simple method for continuous state and action spaces [130, 174].

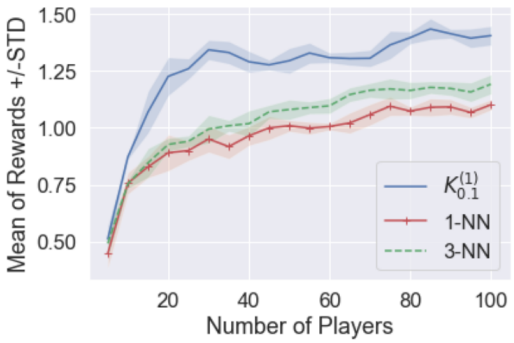
All $K_{0.1}^{(1)}(x,y)$ and k-NN converge within 15 outer iterations. The performances of $K_{0.1}^{1}(x,y)$ and k-NN are similar when $N \le 10$. However, $K_{0.1}^{(1)}(x,y)$ outperforms both 1-NN and 3-NN for large N under both criteria $C_N^{(1)}$ and $C_N^{(2)}$: under $C_N^{(1)}$, $K_{0.1}^{(1)}(x,y)$, 1-NN, and 3-NN have respectively average rewards of 1.4, 1.07, and 1.2 when $N \ge 65$; under $C_N^{(2)}$, $K_{0.1}^{(1)}(x,y)$ outperforms 1-NN and 3-NN by 15% and 13% respectively when $N=10$, by 29% and 21% respectively when $N=15$, and by 25% and 16% respectively when $N \ge 60$.

Comparison with other algorithms. We compare MFC-K-Q with kernel $K_{0.1}^{(1)}$ with three representative algorithms, MFQ from [36], Deep PPQ from [83], and PCC-VIVACE from [51] on MARL. Our experiment demonstrates superior performances of MFC-K-Q.

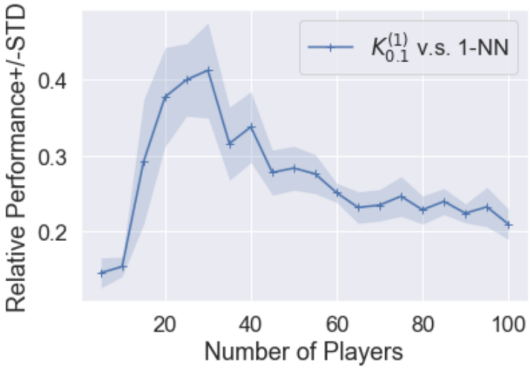
- When $N > 40$, MFC-K-Q dominates all these three algorithms (Figure 3.4a) and it learns the bandwidth parameter c most accurately (Figure 3.4b). Despite being the best performer when $N < 35$, Deep PPQ suffers from the "curse of dimensionality" and the performance gets increasingly worse when N increases;



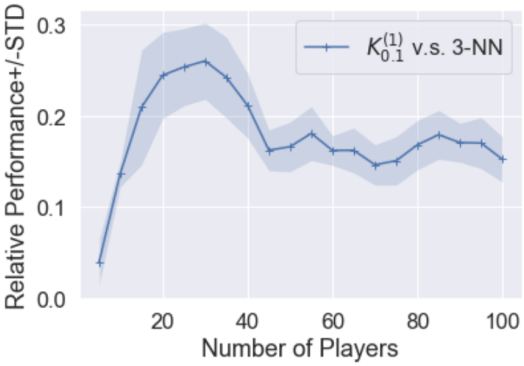
(a) Convergence of Q-function



(b) $C_N^{(1)}$: Average reward.



(c) $C_N^{(2)}$: Improvement of $K_{0.1}^{(1)}$ from 1-NN.



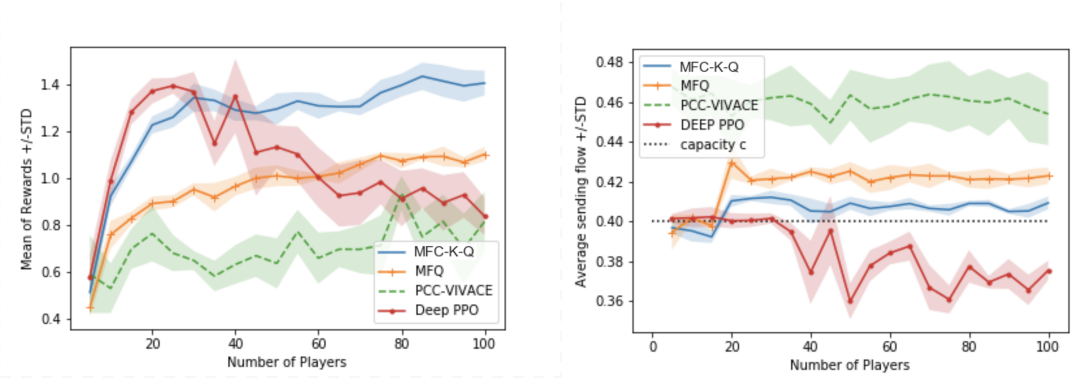
(d) $C_N^{(2)}$: Improvement of $K_{0.1}^{(1)}$ from 3-NN.

Figure 3.3: Comparison between MFC-K-Q with kernel $K_{0.1}^1(x, y)$ in (3.7.1) and MFC-K-Q with k-NN method ($k = 1, 3$). More specifically, convergence of Q-function in Figure 3.3a; average reward in Figure 3.3b; relative reward improvement in Figure 3.3c and 3.3d.

- MFC-K-Q with $K_{0.1}^{(1)}$ dominates MFQ, which is similar to our worst performer MFC-K-Q with 1-NN. In general, kernel regression performs better than simple projection (adopted in MFQ) where only one point is used to estimate Q;
- the decentralized PCC-VIVACE has the worst performance. Moreover, it is insensitive to the bandwidth parameter c. See Figure 3.4b.

3.8 Proofs of Lemmas

Proof of Lemma 3.2.2 At time step t, assume $s_t \sim \mu_t$. Under the policy π_t, it is easy to check via direct computation that the corresponding action distribution ν_t is $\nu(\mu_t, \pi_t(\cdot, \mu_t))$.



(a) $C_N^{(1)}$: Average reward.

(b) Average sending flow.

Figure 3.4: Performance of four algorithms on the network traffic congestion control problem: MFC-K-Q proposed in this chapter, MFQ from [36], Deep PPQ from [83], and PCC-VIVACE from [51] on MARL. Figure 3.4a shows that MFC-K-Q dominates all other three algorithms in terms of the accumulated rewards, especially when the number of agents is large ($N > 40$). Figure 3.4b indicates MFC-K-Q learns the bandwidth parameter c most accurately.

Meanwhile, for any bounded function φ on $\mathcal{S}$, by the law of iterated conditional expectation:

$$\begin{aligned}\mathbb{E}^\pi[\varphi(s_{t+1})] &= \mathbb{E}^\pi\Big[\mathbb{E}^\pi\big[\varphi(s_{t+1})|s_0\ldots,s_t\big]\Big] = \mathbb{E}^\pi\Big[\sum_{x'\in\mathcal{S}}\varphi(x')P(s_t,\mu_t,a_t,\nu_t)(x')\Big]\\ &= \sum_{x'\in\mathcal{S}}\varphi(x')\mathbb{E}^\pi\Big[P(s_t,\mu_t,a_t,\nu_t)(x')\Big]\\ &= \sum_{x'\in\mathcal{S}}\varphi(x')\sum_{s\in\mathcal{S}}\mu_t(s)\sum_{a\in\mathcal{A}}\pi_t(s,\mu_t)(a)P(s,\mu_t,a,\nu_t)(x'),\end{aligned}$$

which concludes that $s_{t+1}\sim\Phi(\mu_t,\pi_t(\cdot,\mu_t))$. Here $\mathbb{E}^\pi$ denotes the expectation under policy π. Therefore, under $\pi=\{\pi_t\}_{t=0}^\infty$, $\mu_{t+1}=\Phi(\mu_t,\pi_t(\cdot,\mu_t))$ defines a deterministic flow $\{\mu_t\}_{t=0}^\infty$ in $\mathcal{P}(\mathcal{S})$, and $s_t\sim\mu_t$. Moreover, by Fubini's theorem

$$\begin{aligned}v^\pi(\mu) &= \mathbb{E}^\pi\Big[\sum_{t=0}^\infty\gamma^t\tilde{r}(s_t,\mu_t,a_t,\nu_t)\Big|s_0\sim\mu\Big] = \sum_{t=0}^\infty\gamma^t\mathbb{E}^\pi\Big[\tilde{r}(s_t,\mu_t,a_t,\nu_t)\Big|s_0\sim\mu\Big]\\ &= \sum_{t=0}^\infty\gamma^t\mathbb{E}\Big[\tilde{r}(s_t,\mu_t,a_t,\nu_t)\Big|s_t\sim\mu_t,a_t\sim\pi_t(s_t,\mu_t)\Big]\\ &= \sum_{t=0}^\infty\gamma^t\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\tilde{r}(s,\mu_t,a,\nu(\mu_t,\pi_t(\cdot,\mu_t)))\mu_t(s)\pi_t(s,\mu_t)(a)\\ &= \sum_{t=0}^\infty\gamma^t r(\mu_t,\pi_t(\cdot,\mu_t)).\end{aligned}$$

This proves (3.2.5). □

Proof of Lemma 3.2.6

$$\begin{aligned}
\|\nu(\mu,h)-\nu(\mu',h')\|_1 &\le \|\nu(\mu,h)-\nu(\mu,h')\|_1+\|\nu(\mu,h')-\nu(\mu',h')\|_1\\
&\le \Big|\Big|\sum_{s\in\mathcal{S}}(h(s)-h'(s))\mu(s)\Big|\Big|_1+\Big|\Big|\sum_{s\in\mathcal{S}}(\mu(s)-\mu'(s))h'(s)\Big|\Big|_1\\
&\le \sum_{s\in\mathcal{S}}\mu(s)\Big|\Big|h(s)-h'(s)\Big|\Big|_1+\Big|\Big|\sum_{s\in\mathcal{S}}(\mu(s)-\mu'(s))h'(s)\Big|\Big|_1\\
&\le \max_{s\in\mathcal{S}}\Big|\Big|h(s)-h'(s)\Big|\Big|_1+\sum_{a\in\mathcal{A}}\sum_{s\in\mathcal{S}}|\mu(s)-\mu'(s)|h'(s)(a)\\
&= d_{\mathcal{H}}(h,h')+\|\mu-\mu'\|_1=d_{\mathcal{C}}((\mu,h),(\mu',h')).
\end{aligned}$$

□

Proof of Lemma 3.2.7

$$\begin{aligned}
&|r(\mu,h)-r(\mu',h')|\\
=\ &\Big|\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\tilde{r}(s,\mu,a,\nu(\mu,h))\mu(s)h(s)(a)-\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\tilde{r}(s,\mu',a,\nu(\mu',h'))\mu'(s)h'(s)(a)\Big|\\
&\text{(For simplicity, denote } \tilde{r}_{s,a}=\tilde{r}(s,\mu,a,\nu(\mu,h)),\tilde{r}'_{s,a}=\tilde{r}(s,\mu',a,\nu(\mu',h')).)\\
\le\ &\Big|\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}(\tilde{r}_{s,a}-\tilde{r}'_{s,a})\mu(s)h(s)(a)\Big|+\Big|\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\tilde{r}'_{s,a}(\mu(s)h(s)(a)-\mu'(s)h'(s)(a))\Big|.
\end{aligned}$$

By Assumption 3.2.4 and Lemma 3.2.6, for any $s\in\mathcal{S},a\in\mathcal{A}$,

$$\begin{aligned}
|\tilde{r}_{s,a}-\tilde{r}'_{s,a}| &\le L_{\tilde{r}}(\|\mu-\mu'\|_1+\|\nu(\mu,h),\nu(\mu',h')\|_1)\\
&\le L_{\tilde{r}}\cdot(\|\mu-\mu'\|_1+d_{\mathcal{C}}((\mu,h),(\mu',h')))\le 2L_{\tilde{r}}d_{\mathcal{C}}((\mu,h),(\mu',h')).
\end{aligned}$$

Meanwhile,

$$\begin{aligned}
&\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}|\mu(s)h(s)(a)-\mu'(s)h'(s)(a)|\\
\le\ &\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}|\mu(s)-\mu'(s)|h(s)(a)+\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\mu'(s)|h(s)(a)-h'(s)(a)|\\
=\ &\sum_{s\in\mathcal{S}}|\mu(s)-\mu'(s)|+\sum_{s\in\mathcal{S}}\mu'(s)\|h(s)-h'(s)\|_1\\
\le\ &\|\mu-\mu'\|_1+\max_{s\in\mathcal{S}}\|h_1(s)-h_2(s)\|_1=d_{\mathcal{C}}((\mu,h),(\mu',h')).
\end{aligned}$$

Combining all these results, we have

$$\begin{aligned}|r(\mu,h)-r(\mu',h')| &\leq \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}|\tilde{r}_{s,a}-\tilde{r}'_{s,a}|\mu(s)h(s)(a)\\ &\quad+\tilde{R}\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}|\mu(s)h(s)(a)-\mu'(s)h'(s)(a)|\\ &\leq (\tilde{R}+2L_{\tilde{r}})d_{\mathcal{C}}((\mu,h),(\mu',h')).\end{aligned}$$

□

Proof of Lemma 3.2.8

$$\begin{aligned}&\|\Phi(\mu,h)-\Phi(\mu',h')\|_1\\ =\ &\Big|\Big|\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}P(s,\mu,a,\nu(\mu,h))\mu(s)h(s)(a)-\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}P(s,\mu',a,\nu(\mu',h'))\mu'(s)h'(s)(a)\Big|\Big|_1\\ &\text{(For simplicity, denote } P_{s,a}=P(s,\mu,a,\nu(\mu,h)), P'_{s,a}=P(s,\mu',a,\nu(\mu',h')).)\\ \leq\ &\Big|\Big|\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}(P_{s,a}-P'_{s,a})\mu(s)h(s)(a)\Big|\Big|_1+\Big|\Big|\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}P'_{s,a}(\mu(s)h(s)(a)-\mu'(s)h'(s)(a))\Big|\Big|_1.\end{aligned}$$

By Assumption 3.2.5 and Lemma 3.2.6, for any x and u,

$$\begin{aligned}||P_{s,a}-P'_{s,a}||_1 &\leq L_P\cdot(\|\mu-\mu'\|_1+\|\nu(\mu,h)-\nu(\mu',h')\|_1)\\ &\leq L_P\cdot(\|\mu-\mu'\|_1+d_{\mathcal{C}}((\mu,h),(\mu',h')))\leq 2L_P\cdot d_{\mathcal{C}}((\mu,h),(\mu',h')).\end{aligned}$$

Meanwhile, from the proof of Lemma 3.2.7, we know

$$\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}|\mu(s)h(s)(a)-\mu'(s)h'(s)(a)|\leq d_{\mathcal{C}}((\mu,h),(\mu',h')).$$

Combining all these results, we have

$$\begin{aligned}\|\Phi(\mu,h)-\Phi(\mu',h')\|_1 &\leq \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}||P_{s,a}-P'_{s,a}||_1\mu(s)h(s,a)\\ &\quad+\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}||P'_{s,a}||_1|\mu(s)h(s)(a)-\mu'(s)h'(s)(a))|\\ &\leq (2L_P+1)d_{\mathcal{C}}((\mu,h),(\mu',h')).\end{aligned}$$

□

Proof of Lemma 3.3.1 To prove the continuity of Q, first fix c and $c' \in \mathcal{C}$. Then there exists some policy π such that $Q(c) - Q^\pi(c) < \frac{\epsilon}{2}$. Let $c = (\mu_0, h_0), (\mu_1, h_1), (\mu_2, h_2), \ldots, (\mu_t, h_t), \ldots$ be the trajectory of the system starting from c and then taking the policy π. Then $Q^\pi(c) = \sum_{t=0}^{\infty} \gamma^t r(\mu_t, h_t)$.

Now consider the trajectory of the system starting from c' and then taking $h_1, \ldots, h_t, \ldots$, denoted by $c' = (\mu'_0, h'_0), (\mu'_1, h_1), (\mu'_2, h_2), \ldots, (\mu'_t, h_t), \ldots$. Note that this trajectory starting from c' may not be the optimal trajectory, therefore, $Q(c') \geq \sum_{t=0}^{\infty} \gamma^t r(\mu'_t, h_t)$. By Lemma 3.2.7 and Lemma 3.2.8,

$$\begin{aligned} |r(\mu'_t, h_t) - r(\mu_t, h_t)| &\leq L_r \cdot d_{\mathcal{P}(\mathcal{S})}(\mu'_t, \mu_t) = L_r \cdot d_{\mathcal{P}(\mathcal{S})}(\Phi(\mu'_{t-1}, h_{t-1}), \Phi(\mu_{t-1}, h_{t-1})) \\ &\leq L_r \cdot L_\Phi \cdot d_{\mathcal{P}(\mathcal{S})}(\mu'_{t-1}, \mu_{t-1}) \leq \cdots \leq L_r \cdot L_\Phi^t \cdot d_{\mathcal{C}}(c, c'), \end{aligned}$$

implying that

$$\begin{aligned} Q(c) - Q(c') &\leq \frac{\epsilon}{2} + Q^\pi(c) - Q(c') \leq \frac{\epsilon}{2} + (r(c) - r(c')) + \sum_{t=1}^{\infty} \gamma^t (r(\mu_t, h_t) - r(\mu'_t, h_t)) \\ &\leq \frac{\epsilon}{2} + \sum_{t=0}^{\infty} \gamma^t \cdot L_\Phi^t \cdot L_r \cdot d_{\mathcal{C}}(c, c') = \frac{\epsilon}{2} + \frac{L_r}{1 - \gamma \cdot L_\Phi} \cdot d_{\mathcal{C}}(c, c'). \end{aligned}$$

Similarly, one can show $Q(c') - Q(c) \leq \frac{\epsilon}{2} + \frac{L_r}{1-\gamma \cdot L_\Phi} \cdot d_{\mathcal{C}}(c, c')$. Therefore, as long as $d_{\mathcal{C}}(c, c') \leq \frac{\epsilon \cdot (1-\gamma \cdot L_\Phi)}{2L_r}$, $|Q(c') - Q(c)| \leq \epsilon$. This proves that Q is continuous. □

Proof of Lemma 3.5.3 By definition, it is easy to show that B and $B_{\mathcal{H}_\epsilon}$ map $\{f \in \mathbb{R}^{\mathcal{C}} : \|f\|_\infty \leq V_{\max}\}$ to itself, B_ϵ and $\widehat{B}_\epsilon$ map $\{f \in \mathbb{R}^{\mathcal{C}_\epsilon} : \|f\|_\infty \leq V_{\max}\}$ to itself, and T maps $\{f \in \mathbb{R}^{\mathcal{P}(\mathcal{S})} : \|f\|_\infty \leq V_{\max}\}$ to itself.

For B_ϵ, we have

$$\begin{aligned} \|B_\epsilon q_1 - B_\epsilon q_2\|_\infty &\leq \gamma \max_{c \in \mathcal{C}_\epsilon} \max_{\tilde{h} \in \mathcal{H}_\epsilon} |\Gamma_K q_1(\Phi(c), \tilde{h}) - \Gamma_K q_2(\Phi(c), \tilde{h})| \\ &\leq \gamma \max_{c \in \mathcal{C}_\epsilon} \max_{\tilde{h} \in \mathcal{H}_\epsilon} \sum_{i=1}^{N_\epsilon} K(c^i, (\Phi(c), \tilde{h})) |q_1(c^i) - q_2(c^i)| \leq \gamma \|q_1 - q_2\|_\infty, \end{aligned}$$

where we use (3.4.2) for the property of kernel function $K(c^i, c)$.

Therefore, B_ϵ is a contraction mapping with modulus $\gamma < 1$ under the sup norm on $\{f \in \mathbb{R}^{\mathcal{C}_\epsilon} : \|f\|_\infty \leq V_{\max}\}$. By Banach Fixed Point Theorem, the statement for B_ϵ holds. Similar arguments prove the statements for the other four operators. □

Proof of Lemma 3.5.4 Using the same DPP argument as in Theorem 3.3.2, we can show the value function for (3.5.9)-(3.5.10) is a fixed point for T (3.5.3) in $\{f \in \mathbb{R}^{\mathcal{P}(\mathcal{S})} : \|f\|_\infty \leq V_{\max}\}$. By Lemma 3.5.3, it coincides with $V_{\mathcal{H}_\epsilon}$.

To prove (3.5.11), recall from Lemma 3.5.3 that T is a contraction mapping with modulus γ with the supremum norm on $\{f \in \mathbb{R}^{\mathcal{P}(\mathcal{S})} : \|f\|_\infty \leq V_{\max}\}$, with a fixed point $V_{\mathcal{H}_\epsilon}$ which is the value function of the MFC (3.5.9)-(3.5.10), i.e., (MDP) with the action space restricted to $\mathcal{H}_\epsilon$. Moreover, define $\tilde{Q}(\mu, h) := r(\mu, h) + \gamma V_{\mathcal{H}_\epsilon}(\Phi(\mu, h))$. Then

$$\begin{aligned}\tilde{Q}(\mu, h) &= r(\mu, h) + \gamma V_{\mathcal{H}_\epsilon}(\Phi(\mu, h)) \\ &= r(\mu, h) + \gamma \max_{\tilde{h}\in\mathcal{H}_\epsilon}(r(\Phi(\mu, h), \tilde{h}) + \gamma V_{\mathcal{H}_\epsilon}(\Phi(\Phi(\mu, h), \tilde{h}))) \\ &= r(\mu, h) + \gamma \max_{\tilde{h}\in\mathcal{H}_\epsilon} \tilde{Q}(\Phi(\mu, h), \tilde{h}).\end{aligned}$$

So $\tilde{Q} \in \{f \in \mathbb{R}^{\mathcal{C}} : \|f\|_\infty \leq V_{\max}\}$ is a fixed point of $B_{\mathcal{H}_\epsilon}$. By Lemma 3.5.3, $\tilde{Q} = Q_{\mathcal{H}_\epsilon}$.

Now, since $Q_{\mathcal{H}_\epsilon}$ is the value function of the MFC problem (3.5.9), replacing Q with $Q_{\mathcal{H}_\epsilon}$ in the argument of Lemma 3.3.4 and then taking $\epsilon \to 0$ yield the Lipschitz continuity of $Q_{\mathcal{H}_\epsilon}$. □

Proof of Lemma 3.5.7 By Markov's inequality,

$$\mathbb{P}(T_{\mathcal{C},\pi} > eT) \leq \frac{\mathbb{E}[T_{\mathcal{C},\pi}]}{eT} \leq \frac{1}{e}.$$

Since $T_{\mathcal{C},\pi}$ is independent of the initial state and the dynamics are Markovian, the probability that $\mathcal{C}_\epsilon$ has not been covered during any time period with length eT is less or equal to $\frac{1}{e}$. Therefore, for any positive integer k, $\mathbb{P}(T_{\mathcal{C},\pi} > ekT) \leq \frac{1}{e^k}$. Take $k = \log(1/\delta)$ and we get the desired result. □

Proof of Corollary 3.6.4 From (3.6.1), we have for any $\mu^N = \frac{\sum_{j=1}^N 1(s^{j,N}=s)}{N}$,

$$v^{\pi^*}(\mu^N) - \frac{C}{\sqrt{N}} \leq a_N^{\pi^*}(\mu^N) \leq v^{\pi^*}(\mu^N) + \frac{C}{\sqrt{N}},$$

$$v^{\tilde{\pi}}(\mu^N) - \frac{C}{\sqrt{N}} \leq a_N^{\tilde{\pi}}(\mu^N) \leq v^{\tilde{\pi}}(\mu^N) + \frac{C}{\sqrt{N}}.$$

By the optimality condition, we have $v^{\tilde{\pi}}(\mu^N) \leq v^{\pi^*}(\mu^N)$. Hence

$$a_N^{\tilde{\pi}}(\mu^N) \leq v^{\tilde{\pi}}(\mu^N) + \frac{C}{\sqrt{N}} \leq v^{\pi^*}(\mu^N) + \frac{C}{\sqrt{N}}. \tag{3.8.1}$$

Similarly since $a^{\tilde{\pi}}(\mu^N) \geq a^{\pi^*}(\mu^N)$, we have

$$v_N^{\pi^*}(\mu^N) \leq a^{\pi^*}(\mu^N) + \frac{C}{\sqrt{N}} \leq a^{\tilde{\pi}}(\mu^N) + \frac{C}{\sqrt{N}}. \tag{3.8.2}$$

Combining (3.8.1) and (3.8.2) leads to the desired result. □

3.9 Discussions and Future Works

Related works on kernel-based reinforcement learning. Kernel method is a popular dimension reduction technique to map high-dimensional features into a low dimension space that best represents the original features. This technique was first introduced for RL by [135, 134], in which a kernel-based reinforcement learning algorithm (KBRL) was proposed to handle the continuity of the state space. Subsequent works demonstrated the applicability of KBRL to large-scale problems and for various types of RL algorithms [12, 170, 193]. However, there is no prior work on convergence rate or sample complexity analysis.

Our kernel regression idea is closely related to [153], which combined Q-learning with kernel-based nearest neighbor regression to study continuous-state stochastic MDPs with sample complexity guarantee. However, our problem setting and technique for error bound analysis are different from theirs. In particular, Theorem 3.5.5 has both action space approximation and state space approximation; whereas [153] has only state space approximation and their action space is finite. The error control in [153] was obtained via martingale concentration inequalities whereas ours is by the regularity property of the underlying dynamics. Other than the kernel regression method, one could also consider the empirical (or approximate) dynamic programming approach to handle the infinite dimensional problem [41, 74].

Stochastic vs deterministic dynamics. We reiterate that unlike learning algorithms for stochastic dynamics where the choice of learning rate η_t is to guarantee the convergence of the Q-function (see e.g. [185]), MFC-K-Q directly conducts the fixed point iteration for the approximated Bellman operator B_ϵ on the sampled data set, and sets the learning rate as 1 to fully utilize the deterministic nature of the dynamics. Consequently, complexity analysis of this algorithm is reduced significantly. By comparison, for stochastic systems each component in the ϵ-net has to be visited sufficiently many times for a decent estimate in Q-learning.

Sample complexity comparison. Theorem 3.5.6 shows that sample complexity for MFC with learning is $\Omega(\text{poly}((1/\epsilon)\cdot\log(1/\delta)))$, instead of the exponential rate in N by existing algorithms for cooperative MARL in Proposition 3.2.1. Careful readings reveal that this complexity analysis holds for other exploration schemes, including the Gaussian exploration and the Boltzmann exploration, as long as Lemma 3.5.7 holds.

Convergence under different norms. Our main assumptions and results adopt the infinity norm ($\|\cdot\|_\infty$) for ease of exposition. Under appropriate assumptions on the mixing behavior of the mean-field dynamic, and applying techniques in [131], the convergence results can also be established under the L_p ($\|\cdot\|_p$) norm to allow for the function approximation of Q-learning. In addition, by properly controlling the Lipschtiz constant, the empirical performance of the neural network approximation may be further improved ([9]).

Extensions to other settings. For future research, we are interested in extending our framework and learning algorithm to other variations of mean-field controls including risk-sensitive mean-field controls [17, 48, 49], robust mean-field controls [179], mean-field controls on polish space [146], and partially observed mean-field controls [48, 148].

If the state space of each individual player is a Polish space [146], one can adopt, instead of the Q learning framework in this chapter, Proximal Policy Optimization (PPO) type of algorithms [152, 151]. In this framework, the mean-field information on the lifted probability measure may be incorporated via a mean embedding technique, which embeds the mean-field states into a reproducing kernel Hilbert space (RKHS) [159, 69].

Given the connection between the Q-function and the Hamiltonian of nonlinear control problem with single-agent [120], one may also extend the kernel-based Q learning algorithm to more general nonlinear mean-field control problems.

Chapter 4

Decentralized Cooperative Mean-Field MARL

One of the challenges for multi-agent reinforcement learning (MARL) is designing efficient learning algorithms for a large system in which each agent has only limited or partial information of the entire system. While exciting progress has been made to analyze decentralized MARL with the *network of agents* for social networks and team video games, little is known theoretically for decentralized MARL with the *network of states* for modeling self-driving vehicles, ride-sharing, and data and traffic routing.

This chapter proposes a framework of *localized training and decentralized execution* to study MARL with *network of states*. Localized training means that agents only need to collect local information in their neighboring states during the training phase; decentralized execution implies that agents can execute afterwards the learned decentralized policies, which depend only on agents' current states.

The theoretical analysis consists of three key components: the first is the reformulation of the MARL system as a networked Markov decision process with teams of agents, enabling updating the associated team Q-function in a localized fashion; the second is the Bellman equation for the value function and the appropriate Q-function on the probability measure space; and the third is the exponential decay property of the team Q-function, facilitating its approximation with efficient sample efficiency and controllable error.

The theoretical analysis paves the way for a new algorithm LTDE-Neural-AC, where the actor-critic approach with over-parameterized neural networks is proposed. The convergence and sample complexity is established and shown to be scalable with respect to the sizes of both agents and states. To the best of our knowledge, this is the first neural network based MARL algorithm with network structure and provably convergence guarantee.

4.1 Motivation and Related Works

Multi-agent reinforcement learning (MARL) has achieved substantial successes in a broad range of cooperative games and their applications, including coordination of robot swarms [81], self-driving vehicles [154, 28], real-time bidding games [87], ride-sharing [100], power management [209] and traffic routing [54].

One of the challenges for the development of MARL is designing efficient learning algorithms for a large system, in which each individual agent has only limited or partial information of the entire system. In such a system, it is necessary to design algorithms to learn policies of the decentralized type, i.e., policies that depend only on the *local* information of each agent.

In a simulated or laboratory setting, decentralized policies may be learned in a centralized fashion. It is to train a central controller to dictate the actions of all agents. Such paradigm of *centralized training with decentralized execution* has achieved significant empirical successes, especially with the computational power of deep neural networks [112, 58, 40, 145, 197, 173]. Such a training approach, however, suffers from the curse of dimensionality as the computational complexity grows exponentially with the number of agents [205]; it also requires extensive and costly communications between the central controller and all agents [143]. Moreover, policies derived from the centralized training stage may not be robust in the execution phase [203]. Most importantly, this approach has not been supported or analyzed theoretically.

An alternative and promising paradigm is to take into consideration the network structure of the system to train decentralized policies. Compared with the centralized training approach, exploiting network structures makes the training procedure more efficient as it allows the algorithm to be updated with parallel computing and reduces communication cost.

There are two distinct types of network structures. The first is the *network of agents*, often found in social networks such as Facebook and Twitter, as well as team video games including StarCraft II. This network describes *interactions and relations among heterogeneous agents*. For MARL systems with such network of agents, [206] establishes the asymptotic convergence of decentralized-actor-critic algorithms which are scalable in agent actions. Similar ideas are extended to the continuous space where deterministic policy gradient method (DPG) is used [204], with finite-sample analysis for such framework established in the batch setting [207]. [142] studies a network of agents where state and action interact in a local manner; by exploiting the network structure and the exponential decay property of the Q-function, it proposes an actor-critic framework scalable in both actions and states. Similar framework is considered for the linear quadratic case with local policy gradients conducted with zero order optimization and parallel updating [101].

The second type of network, *the network of states*, has been frequently used for modeling self-driving vehicles, ride-sharing, and data and traffic routing. It focuses on the *state of agents*. Compared with the network of agents which is *static* from agent's perspective [162], the network of states is *stochastic*: neighboring agents of any given agent may change

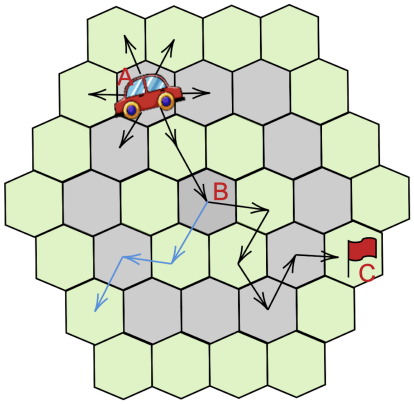


Figure 4.1: Illustration of the Hexagon grid system studied in the transportation networks.

dynamically. This type of network has been empirically studied in various applications, including packet routing [200], traffic routing [30, 71], resource allocations [31] and social economic systems [208]. However, there is no existing theoretical analysis for this type of decentralized MARL. Moreover, the dynamic nature of agents' relationship makes it difficult to adopt existing methodology from the static network of agents. The goal of this chapter is, therefore, to fill the gap.

Motivating example. To get the essence of the network of states, let us consider the following ride-hailing dispatch problem, studied empirically in [100] via the multi-agent RL approach. In this problem, the rides/demands are exogenous and drivers/supplies are distributed at different locations on a (transportation) network, where the state includes the location of drivers within the graph and her status of being idle or occupied. Driver's action is state-dependent: she can only take a new order when the her status is "idle" *and* when the pick-up location is reachable within k steps, i.e., within the k-hop neighborhood of her current location on the graph. If she is occupied, her only allowable action is to continue with the current order till the destination. The reward function has two main components. The first one is the usual payment the driver receives upon completing a trip, which is proportional to the distance traveled. In addition to this standard payment, there are rebates which take into account the supply-demand imbalance in both *the origin* and *the destination* of any impending trip: one rebate for the driver when she accepts orders in locations where the demand is higher than the supply; another rebated for her from the supply-demand imbalance in the k-hop neighborhood of the destination. This last one is known as "order destination potential" in the literature which measures the potential of the origin for the next ride.

The above example highlights a couple of features common in transportation networks: 1) the reward function relies on the aggregated information of drivers and riders, with additional rebates for imbalance between the supply and the demand; and 2), the network is a Hexagon

grid system [141], shown in Figure 4.1. This network is sparse in the sense that drivers travel only to neighboring states within a single time step. These two stylized yet critical features are the basis of our mathematical formulation in order to develop a scalable and efficient learning framework.

Contributions. Motivated by this transportation network, this chapter proposes and studies multi-agent systems with *network of states*. In this network, homogeneous agents can move from one state to any connecting state, and observe only partial information of the entire system in an aggregated fashion. To analyze this system, we propose a framework of *localized training and decentralized execution* (LTDE). Localized training means that agents only need to collect local information in their neighboring states during the training phase; decentralized execution implies that, agents can execute afterwards the learned decentralized policies which only require knowledge of agents' current states.

The theoretical analysis consists of three key elements. The first is the regrouping of homogeneous agents according to their states and reformulation of the MARL system as a networked Markov decision process with teams of agents. This part leads to the decomposition of the Q-function and the value function according to the states, enabling the update of the consequent team Q-function in a localized fashion. The second is the establishment of the Bellman equation for the value function and the appropriate Q-function on the probability measure space, by utilizing the homogeneity of agents. These functions are invariant with respect to the number of agents. The third is the exploration of the exponential decay property of the team Q-function, enabling its approximation with a truncated version of a much smaller dimension and yet with a controllable approximation error.

This last piece is inspired by earlier studies of exponential decay in random graphs (e.g., [61, 62]) and extensive analysis of network among heterogeneous agents (e.g., [142, 105]).

To design an efficient and scalable reinforcement learning algorithm for such framework, the actor-critic approach with over-parameterized neural networks is adopted. The neural networks, representing decentralized policies and localized Q-functions, are much smaller compared with the global one. The convergence and the sample complexity of the proposed algorithm is established and shown to be scalable with respect to the size of both agents and states. The techniques to prove the convergence of the neural actor-critic algorithm are adapted from the single-agent case in [181] to the multi-agents setting.

To the best of our knowledge, this work is the first neural- network-based MARL algorithm with network structures and with provably convergence guarantee. In particular, this work contributes to two lines of research: MARL and CTDE.

First, we build a theoretical framework that incorporates network structures in the MARL framework, and provide computationally efficient algorithms where each agent only needs local information of neighborhood states to learn and to execute the policy. In contrast, existing works for mean-field control with reinforcement learning, including the Q-learning algorithm proposed in Chapter 3, require that each agent have the full information of the population distribution [35, 36, 128], although in most applications agents only have access

to partial or limited information [194].

Secondly, this work builds the theoretical foundation for the practically popular scheme of centralized-training-decentralized-execution (CTDE) [112, 145, 173, 197]. The CTDE framework is first proposed in [112] to learn optimal policies in cooperative games with two steps: the first step is to train a global policy for the central controller, and the second one is to decompose the central policy (i.e., a large Q-table) into individual policies so that individual agent can apply the decomposed/decentralized policy after training. Despite the popularity of CTDE, however, there has been no theoretical study as to when the Q-table can be decomposed and when the truncation error can be controlled, except for a heuristic argument by [112] for large N with local observations. This work analyzes for the first time with theoretical guarantee that applying our algorithm to this CTDE paradigm yields a near-optimal sample complexity, when there is a network structure among agent states. Moreover, our algorithm, which is easier to scale-up, improves the centralized training step with a localized training. To differentiate our approach from the CTDE scheme, we call it localized-training-decentralized-execution (LTDE).

Notation. For a set $\mathcal{X}$, denote $\mathbb{R}^{\mathcal{X}} = \{f : \mathcal{X} \to \mathbb{R}\}$ as the set of all real-valued functions on $\mathcal{X}$. For each $f \in \mathbb{R}^{\mathcal{X}}$, define $\|f\|_\infty = \sup_{x\in\mathcal{X}} |f(x)|$ as the sup norm of f. In addition, when $\mathcal{X}$ is finite, denote $|\mathcal{X}|$ as the size of $\mathcal{X}$, and $\mathcal{P}(\mathcal{X})$ as the set of all probability measures on $\mathcal{X}$: $\mathcal{P}(\mathcal{X}) = \{p : p(x) \geq 0, \sum_{x\in\mathcal{X}} p(x) = 1\}$, which is equivalent to the probability simplex in $\mathbb{R}^{|\mathcal{X}|}$. $[N] := \{1, 2, \cdots, N\}$. For any $\mu \in \mathcal{P}(\mathcal{X})$ and a subset $\mathcal{Y} \subset \mathcal{X}$, let $\mu(\mathcal{Y})$ denote the restriction of the vector μ on $\mathcal{Y}$, and let $\mathcal{P}(\mathcal{Y})$ denote the set $\{\mu(\mathcal{Y}) : \mu \in \mathcal{P}(\mathcal{X})\}$. For $x \in \mathbb{R}^d$, $d \in \mathbb{N}$, denote $\|x\|_2$ as the L^2-norm of x and $\|x\|_\infty$ as the L^∞-norm of x.

4.2 Mean-Field MARL with Local Dependency

The focus of this chapter is to study a cooperative multi-agent system with a network of agent states, which consists of nodes representing states of the agents and edges by which states are connected. In this system, every agent is only allowed to move from her present state to its connecting states. Moreover, she is assumed to only observe (realistically) *partial information* of the system on an aggregated level. Mean-field theory provides efficient approximations when agents only observe aggregated information, and has been applied in stochastic systems with large homogeneous agents such as financial markets [34, 95, 78, 37], energy markets [66, 5], and auction systems [82, 72].

4.2.1 Review of MARL

Let us first recall the cooperative MARL in an infinite time horizon, where there are N agents whose policies are coordinated by a central controller. We assume that both the state space $\mathcal{S}$ and the action space $\mathcal{A}$ are finite.

At each step $t = 0, 1, \cdots$, the state of agent i $(= 1, 2, \cdots, N)$ is $s_t^i \in \mathcal{S}$ and she takes an action $a_t^i \in \mathcal{A}$. Given the current state profile $\boldsymbol{s}_t = (s_t^1, \cdots, s_t^N) \in \mathcal{S}^N$ and the current action profile $\boldsymbol{a}_t = (a_t^1, \cdots, a_t^N) \in \mathcal{A}^N$ of N agents, agent i will receive a reward $r^i(\boldsymbol{s}_t, \boldsymbol{a}_t)$ and her state will change to s_{t+1}^i according to a transition probability function $P^i(\boldsymbol{s}_t, \boldsymbol{a}_t)$. A Markovian game further restricts the admissible policy for agent i to be of the form $a_t^i \sim \pi_t^i(\boldsymbol{s}_t)$. That is, $\pi_t^i : \mathcal{S}^N \to \mathcal{P}(\mathcal{A})$ maps each state profile $\boldsymbol{s} \in \mathcal{S}^N$ to a randomized action, with $\mathcal{P}(\mathcal{A})$ the space of all probability measures on space $\mathcal{A}$.

In this cooperative MARL framework, the central controller is to maximize the expected discounted accumulated reward averaged over all agents. That is to find

$$V(\boldsymbol{s}) = \sup_{\boldsymbol{\pi}} \frac{1}{N} \sum_{i=1}^{N} v^i(\boldsymbol{s}, \boldsymbol{\pi}), \tag{4.2.1}$$

where

$$v^i(\boldsymbol{s}, \boldsymbol{\pi}) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r^i(\boldsymbol{s}_t, \boldsymbol{a}_t) \big| \boldsymbol{s}_0 = \boldsymbol{s} \right] \tag{4.2.2}$$

is the accumulated reward for agent i, given the initial state profile $\boldsymbol{s}_0 = \boldsymbol{s}$ and policy $\boldsymbol{\pi} = \{\boldsymbol{\pi}_t\}_{t=0}^{\infty}$ with $\boldsymbol{\pi}_t = (\pi_t^1, \ldots, \pi_t^N)$. Here $\gamma \in (0, 1)$ is a discount factor, $a_t^i \sim \pi_t^i(\boldsymbol{s}_t)$, and $s_{t+1}^i \sim P^i(\boldsymbol{s}_t, \boldsymbol{a}_t)$.

The corresponding Bellman equation for the value function (4.2.1) is

$$V(\boldsymbol{s}) = \max_{\boldsymbol{a} \in \mathcal{A}^N} \left\{ \mathbb{E}\left[\frac{1}{N} \sum_{i=1}^{N} r^i(\boldsymbol{s}, \boldsymbol{a}) \right] + \gamma \mathbb{E}_{\boldsymbol{s}' \sim \boldsymbol{P}(\boldsymbol{s}, \boldsymbol{a})} \left[V(\boldsymbol{s}') \right] \right\}, \tag{4.2.3}$$

with the population transition kernel $\boldsymbol{P} = (P^1, \cdots, P^N)$. The value function can be written as

$$V(\boldsymbol{s}) = \max_{\boldsymbol{a} \in \mathcal{A}^N} Q(\boldsymbol{s}, \boldsymbol{a}),$$

in which the Q-function is defined as

$$Q(\boldsymbol{s}, \boldsymbol{a}) = \mathbb{E}\left[\frac{1}{N} \sum_{i=1}^{N} r^i(\boldsymbol{s}, \boldsymbol{a}) \right] + \gamma \mathbb{E}_{\boldsymbol{s}' \sim \boldsymbol{P}(\boldsymbol{s}, \boldsymbol{a})} [V(\boldsymbol{s}')], \tag{4.2.4}$$

consisting of the expected reward from taking action $\boldsymbol{a}$ at state $\boldsymbol{s}$ and then following the optimal policy thereafter. The Bellman equation for the Q-function, defined from $\mathcal{S}^N \times \mathcal{A}^N$ to $\mathbb{R}$, is given by

$$Q(\boldsymbol{s}, \boldsymbol{a}) = \mathbb{E}\left[\frac{1}{N} \sum_{i=1}^{N} r^i(\boldsymbol{s}, \boldsymbol{a}) \right] + \gamma \mathbb{E}_{\boldsymbol{s}' \sim \boldsymbol{P}(\boldsymbol{s}, \boldsymbol{a})} \left[\max_{\boldsymbol{a}' \in \mathcal{A}^N} Q(\boldsymbol{s}', \boldsymbol{a}') \right]. \tag{4.2.5}$$

One can thus retrieve the optimal (stationary) control $\pi^*(\boldsymbol{s}, \boldsymbol{a})$ (if it exists) from $Q(\boldsymbol{s}, \boldsymbol{a})$, with $\pi^*(\boldsymbol{s}) \in \arg\max_{\boldsymbol{a} \in \mathcal{A}^N} Q(\boldsymbol{s}, \boldsymbol{a})$.

4.2.2 Mean-field MARL with Local Dependency

In this system, there are N agents who share a finite state space $\mathcal{S}$ and take actions from a finite action space $\mathcal{A}$. Moreover, there is a network on the state space $\mathcal{S}$ associated with an underlying undirected graph $(\mathcal{S}, \mathcal{E})$, where $\mathcal{E} \subset \mathcal{S} \times \mathcal{S}$ is the set of edges. The distance between two nodes is defined as the number of edges in a shortest path. For a given $s \in \mathcal{S}$, $\mathcal{N}_s^1$ denotes the nearest neighbor of s, which consists of all nodes connected to s by an edge and includes s itself; and $\mathcal{N}_s^k$ denotes the k-hop neighborhood of s, which consists of all nodes whose distance to s is less than or equal to k, including s itself. For simplicity, we use $\mathcal{N}_s := \mathcal{N}_s^1$. From agent i's perspective, agents in her neighborhood $\mathcal{N}_{s_t^i}$ change stochastically over time.

To facilitate mean-field approximation to this system, assume throughout the chapter that the agents are homogeneous and indistinguishable. In particular, at each step $t = 0, 1, \cdots$, if agent i at state $s_t^i \in \mathcal{S}$ takes an action $a_t^i \in \mathcal{A}$, then she will receive a *localized* stochastic reward which is uniformly upper bounded by $r_{\max}$ such that

$$r^i(\boldsymbol{s}_t, \boldsymbol{a}_t) := r\Big(s_t^i,\ \mu_t(\mathcal{N}_{s_t^i}),\ a_t^i\Big) \le r_{\max}, \quad i \in [N]; \tag{4.2.6}$$

and her state will change to a neighboring state $s_{t+1}^i \in \mathcal{N}_{s_t^i}$ according to a *localized* transition probability such that

$$s_{t+1}^i \sim P^i(\boldsymbol{s}_t, \boldsymbol{a}_t) := P\left(\cdot \,\middle|\, s_t^i,\ \mu_t(\mathcal{N}_{s_t^i}),\ a_t^i\right), \quad i \in [N], \tag{4.2.7}$$

where

$$\begin{aligned}\mu_t(\cdot) &= \frac{\sum_{i=1}^N \mathbf{1}(s_t^i = \cdot)}{N} \in \mathcal{P}^N(\mathcal{S}) \\ &:= \left\{\mu \in \mathcal{P}(\mathcal{S}) : \mu(s) \in \left\{0, \frac{1}{N}, \frac{2}{N}, \cdots, \frac{N-1}{N}, 1\right\} \text{ for all } s \in \mathcal{S}\right\}\end{aligned}$$

is the empirical state distribution of N agents at time t, with $N \cdot \mu_t(s)$ the number of agents in state s at time t, and $\mu_t(\mathcal{N}_{s_t^i})$ denotes the truncation of the μ_t vector with indices in $\mathcal{N}_{s_t^i}$, i.e., $\mu_t(\mathcal{N}_{s_t^i}) := \{\mu_t(s)\}_{s \in \mathcal{N}_{s_t^i}}$.

(4.2.6)-(4.2.7) indicate that the reward and the transition probability of agent i at time t depend on both her individual information (a_t^i, s_t^i) and the mean-field of her 1-hop neighborhood $\mu_t(\mathcal{N}_{s_t^i})$, in an aggregated yet localized format: *aggregated* or *mean-field* meaning that agent i depends on other agents only through the empirical state distribution; *localized* meaning that agent i depends on the mean-field information of her 1-hop neighborhood. Intuitive examples of such a setting include traffic-routing, package delivery, data routing, resource allocations, distributed control of autonomous vehicles and social economic systems.

Policies with partial information. To incorporate the element of *partial or limited information* into this mean-field MARL system, consider the following *individual-decentralized*

policies

$$a_t^i \sim \pi^i(\boldsymbol{s}_t) := \pi\left(s_t^i, \mu_t(s_t^i)\right) \in \mathcal{P}(\mathcal{A}), \quad i \in [N], \tag{4.2.8}$$

and denote $\mathfrak{u}$ as the admissible policy set of all such policies.

Note that for a given mean-field information μ_t, $\pi(\cdot, \mu_t(\cdot)) : \mathcal{S} \to \mathcal{P}(\mathcal{A})$ maps the agent state to a randomized action. That is, the policy of each agent is executed in a decentralized manner and assumes that each agent only has access to the population information in her own state. This is more realistic than centralized policies which assume full access to the state information of all agents.

Value function and Q-function. The goal for this mean-field MARL is to maximize the expected discounted accumulated reward averaged over all agents, i.e.,

$$V(\mu) := \sup_{\pi\in\mathfrak{u}} V^\pi(\mu) = \sup_{\pi\in\mathfrak{u}} \frac{1}{N}\sum_{i=1}^N \mathbb{E}\left[\sum_{t=0}^\infty \gamma^t r\big(s_t^i,\ \mu_t(\mathcal{N}_{s_t^i}),\ a_t^i\big) \,\middle|\, \mu_0 = \mu\right], \tag{MF-MARL}$$

subject to (4.2.6)-(4.2.8) with a discount factor $\gamma \in (0,1)$.

The mean-field assumption leads to the following definition of the corresponding Q-function for (MF-MARL) on the measure space:

$$\begin{aligned} Q(\mu, h) : \ = \ & \underbrace{\mathbb{E}\left[\sum_{i=1}^N \frac{1}{N} r(s_0^i, \mu(\mathcal{N}_{s_0^i}), a_0^i) \,\middle|\, \boldsymbol{s}_0, \boldsymbol{a}_0\right]}_{\text{Expected reward of taking } \boldsymbol{a}_0 = (a_0^1, \cdots, a_0^N)} \\ & + \underbrace{\mathbb{E}_{s_1^i \sim P\left(\cdot \,\middle|\, s_0^i,\ \mu(\mathcal{N}_{s_0^i}),\ a_0^i\right)}\left[\sum_{t=1}^\infty \gamma^t \sum_{i=1}^N \frac{1}{N} r(s_t^i, \mu_t(\mathcal{N}_{s_t^i}), a_t^i) \,\middle|\, a_t^i \sim \pi_t^\star\right]}_{\text{Expected reward of playing optimally thereafter } a_t^i \sim \pi_t^\star}, \end{aligned} \tag{4.2.9}$$

where

$$\mu(\cdot) = \frac{\sum_{i=1}^N \mathbf{1}(s_0^i = \cdot)}{N}$$

is the initial empirical state distribution and

$$h(s)(a) = \frac{\sum_{i=1}^N \mathbf{1}(s_0^i = s, a_0^i = a)}{\sum_{i=1}^N \mathbf{1}(s_0^i = s)}$$

is a "decentralized" policy representing the proportion of agents in state s that takes action a. Specifically, given $\mu \in \mathcal{P}^N(\mathcal{S})$, $s \in \mathcal{S}$, and the $N \cdot \mu(s)$ agents in state s,

$$h(s) \in \mathcal{P}^{N\cdot\mu(s)}(\mathcal{A}) := \left\{\varsigma \in \mathcal{P}(\mathcal{A}) : \varsigma(a) \in \left\{0, \frac{1}{N\cdot\mu(s)}, \cdots, \frac{N\cdot\mu(s)-1}{N\cdot\mu(s)}\right\} \text{ for all } a \in \mathcal{A}\right\},$$

where ς in $\mathcal{P}^{N\cdot\mu(s)}(\mathcal{A})$ is an empirical action distribution of $N\cdot\mu(s)$ agents in state s, and $\varsigma(a)$ is the proportion of agents taking action $a\in\mathcal{A}$ among all $N\cdot\mu(s)$ agents in state s. Furthermore, for a given $s\in\mathcal{S}$, denote $\mathcal{P}^{N\cdot\mu(s)}(\mathcal{A})$ the set of all admissible "decentralized" policies $h(s)(\cdot)$; and for a given $\mu\in\mathcal{P}^N(\mathcal{S})$, denote the product of $\mathcal{P}^{N\cdot\mu(s)}(\mathcal{A})$ over all states by $\mathcal{H}^N(\mu):=\{h: h(s)\in\mathcal{P}^{N\cdot\mu(s)}(\mathcal{A})\ \forall\ s\in\mathcal{S}\}$. Here $\mathcal{H}^N(\mu)$ depends on μ and is a subset of $\mathcal{H}=\{h:\mathcal{S}\to\mathcal{P}(\mathcal{A})\}$.

Remark 4.2.1 *Before further analysis, let us recall some important properties for the value function in* (MF-MARL) *and the Q-function in* (4.2.9).

First is the dynamics programming principle for the mean-field Q function. Take an N-player game, the value function for any $\boldsymbol{s}:=(s_1,s_2,\cdots,s_N)\in\mathcal{S}^N$ is defined as

$$V(\boldsymbol{s}):=\frac{1}{N}\mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t r(\boldsymbol{s}_t,a_t^i)\middle|\boldsymbol{s}_0=\boldsymbol{s}\right].$$

In the mean-field formulation, agents are assumed to be identical and interchangeable, and the empirical state distribution $\mu(\cdot)=\frac{\sum_{i=1}^N\mathbf{1}(s_0^i=\cdot)}{N}$ is the sufficient statistics for the dynamic programming principle (DPP) of the corresponding value function. Analogously, for the mean-field Q-function, it is shown in Chapter 2 that the empirical state distribution $\mu(\cdot)=\frac{\sum_{i=1}^N\mathbf{1}(s_0^i=\cdot)}{N}$ and the empirical action distribution $h:\mathcal{S}\times\mathcal{A}\to\mathbb{R}$, $h(s)(a)=\frac{\sum_{i=1}^N\mathbf{1}(s_0^i=s,a_0^i=a)}{\sum_{i=1}^N\mathbf{1}(s_0^i=s)}$ are the sufficient statistics to establish the associated DPP for the mean-field Q-function, with $h(s)(a)$ representing the proportion of agents in state s who take action a.

Secondly, $Q(\mu,h)$ defined in (4.2.9) *is invariant with respect to the order of the elements in $\boldsymbol{s}_0$ and $\boldsymbol{a}_0$. More critically, the input dimension of the Q-function defined in* (4.2.9) *is independent of the number of agents N in the system, which renders it to be more scalable in the large population regime. This differs from the Q-function defined in* (4.2.4)*, in which the input dimension grows exponentially with respect to the number of agents, the main culprit of the curse of dimensionality for MARL algorithms. (More detailed analysis of the mean-field Q-function can be found in Chapter 2.)*

4.3 Analysis of Mean-Field MARL with Local Dependency

The theoretical study of this mean-field MARL with local dependency (Section 4.2.2) consists of three key components, which are crucial for subsequent algorithm design and convergence analysis: the first is the reformulation of the MARL system as a networked Markov decision process with teams of agents. This reformulation leads to the decomposition of the Q-function and the value function according to states, facilitating updating the consequent team Q-function in a localized fashion (Section 4.3.1); the second is the Bellman equation for the value function and the Q-function on the probability measure space (Section 4.3.2); the third

is the exponential decay property of the team Q-function, enabling its approximation with a truncated version of a much smaller dimension and yet with a controllable approximation error (Section 4.3.3).

4.3.1 Markov Decision Process (MDP) on Network of States

This section shows that the mean-field MARL (4.2.6)-(4.2.8) can be reformulated in an MDP framework by exploiting the network structure of states. This reformulation leads to the decomposition of the Q-function, facilitating more computationally efficient updates.

The key idea is to utilize the homogeneity of the agents in the problem set-up and to regroup these N agents according to their states. This regrouping translates (MF-MARL) with N agents into a networked MDP with $|\mathcal{S}|$ agents teams, indexed by their states.

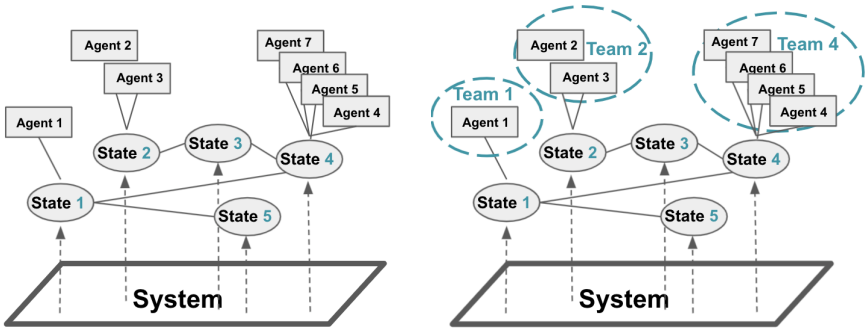


Figure 4.2: **Left:** Illustration of the MF-MARL problem (4.2.6)-(4.2.8) defined on a state network. **Right:** Reformulation of the MF-MARL problem as a team game (4.3.2)-(4.3.6).

To see how the policy, the reward function, and the dynamics in this networked Markov decision process are induced by the regrouping approach, recall that there are $N \cdot \mu(s)$ agents in state s, each agent i in state s will independently choose action $a_i \sim \pi(s, \mu(s))$ according to the individual-decentralized policy $\pi(s, \mu(s)) \in \mathcal{P}(\mathcal{A})$ in (4.2.8). Therefore the empirical action distribution of $\{a_1, \cdots, a_{N\cdot\mu(s)}\}$ is a random variable taking values from $\mathcal{P}^{N\cdot\mu(s)}(\mathcal{A})$, the set of empirical action distributions with $N \cdot \mu(s)$ agents. Moreover, for any $h(s) \in \mathcal{P}^{N\cdot\mu(s)}(\mathcal{A})$, we have

$$\begin{aligned} & \mathbb{P}\left(h(s) \text{ is the empirical action distribution of } \{a_1, \cdots, a_{N\cdot\mu(s)}\}, a_i \overset{i.i.d}{\sim} \pi(s, \mu(s))\right) \\ = \; & \mathbb{P}\left(\text{for each } a \in \mathcal{A}, a \text{ appears } N \cdot \mu(s) h(s)(a) \text{ times in } \{a_1, \cdots, a_{N\cdot\mu(s)}\}, a_i \overset{i.i.d}{\sim} \pi(s, \mu(s))\right) \\ = \; & \frac{(N \cdot \mu(s))!}{\prod_{a\in\mathcal{A}} (N \cdot \mu(s) h(s)(a))!} \prod_{a\in\mathcal{A}} \Big(\pi(s, \mu(s))(a)\Big)^{N\cdot\mu(s)h(s)(a)}. \end{aligned} \tag{4.3.1}$$

Here $h(s)(a)$ denotes the proportion of agents taking action a among all agents in state s, with last equality derived from the multinomial distribution with parameters $N \cdot \mu(s)$ and $\pi(s, \mu(s))$.

Now, clearly each individual-decentralized policy $\pi(s, \mu(s)) \in \mathcal{P}(\mathcal{A})$ in (4.2.8) induces a *team-decentralized policy* of the following form:

$$\Pi_s(h(s) \mid \mu(s)) = \frac{(N \cdot \mu(s))!}{\prod_{a \in \mathcal{A}} (N \cdot \mu(s)h(s)(a))!} \prod_{a \in \mathcal{A}} \Big(\pi(s, \mu(s))(a)\Big)^{N \cdot \mu(s)h(s)(a)}, \tag{4.3.2}$$

where $h(s) \in \mathcal{P}^{N \cdot \mu(s)}(\mathcal{A})$. Conversely, given a team-decentralized policy $\Pi_s(\cdot \mid \mu(s))$, one can recover the individual-decentralized policy $\pi(s, \mu(s))$ by choosing appropriate $h(s) \in \mathcal{P}^{N \cdot \mu(s)}(\mathcal{A})$ and querying the value of $\Pi_s(h(s) \mid \mu(s))$: let $h_i(s) = \delta_{a_i}$ be the Dirac measure with $a_i \in \mathcal{A}$, which is an action distribution such that all agents in state s take action a_i. By (4.3.2), $\Pi_s(h_i(s) \mid \mu(s)) = (\pi(s, \mu(s))(a_i))^{N \cdot \mu(s)}$, implying $\pi(s, \mu(s))(a_i) = (\Pi(h_i(s)|\mu(s))^{\frac{1}{N \cdot \mu(s)}}$.

Next, given $\mu \in \mathcal{P}^N(\mathcal{S})$ and $h \in \mathcal{H}^N(\mu) = \{h : h(s) \in \mathcal{P}^{N \cdot \mu(s)}(\mathcal{A}), \forall s \in \mathcal{S}\}$, the set of empirical action distributions on every state, if we define

$$\Pi(h \mid \mu) := \prod_{s \in \mathcal{S}} \Pi_s(h(s) \mid \mu(s)), \tag{4.3.3}$$

then $\mathfrak{u}$, the admissible policy set of individual-decentralized policies in the form of (4.2.8), is now replaced by $\mathfrak{U}$, the set of all team-decentralized policies Π induced from $\pi \in \mathfrak{u}$ through (4.3.2) and (4.3.3). In addition, denote the set of all state-action distribution pairs as

$$\Xi := \cup_{\mu \in \mathcal{P}^N(\mathcal{S})} \{\zeta = (\mu, h) : h \in \mathcal{H}^N(\mu)\}, \tag{4.3.4}$$

Moreover, from the team perspective, the transition probability in (4.2.7) can be viewed as a Markov process of μ_t and $h_t \in \mathcal{H}^N(\mu_t)$ with an induced transition probability $\mathbf{P}^N$ from (4.2.7) such that

$$\mu_{t+1} \sim \mathbf{P}^N(\cdot \,|\, \mu_t, h_t). \tag{4.3.5}$$

It is easy to verify that for a given state $s \in \mathcal{S}$, $\mu_{t+1}(s)$ only depends on $\mu_t(\mathcal{N}_s^2)$, the empirical distribution in the 2-hop neighborhood of s, and $h_t(\mathcal{N}_s)$. More specifically, each agent can only move from its current state s to a neighboring state in $\mathcal{N}_s$ in each time step. Therefore, the change of population in state s consists of two sources: (1) the out-flow of agents from state s to its neighboring states in $\mathcal{N}_s$; (2) the in-flow of agents from states in $\mathcal{N}_s$ to state s. The out-flow of agents depends on the actions of the agents in state s as well as the transition kernel. Since both the policy and the transition kernel only depend on information $\mu(\mathcal{N}_s)$, the out-flow has a 1-hop neighbor dependence. Similarly, the in-flow from any state $s' \in \mathcal{N}_s$ depends on the information $\mu(\mathcal{N}_{s'})$, which is contained in $\mu(\mathcal{N}_s^2)$ since $\mathcal{N}_{s'} \subset \mathcal{N}_s^2$ for any $s' \in \mathcal{N}_s$. Therefore, the in-flow to s has a 2-hop neighbor dependence. Consequently, the transition of $\mu_{t+1}(s)$ depends only locally on μ_t and h_t through $\mu_t(\mathcal{N}_s^2)$ and $h_t(\mathcal{N}_s)$.

Finally, given $\mu(\mathcal{N}_s) \in \mathcal{P}^N(\mathcal{N}_s)$, an empirical distribution restricted to the 1-hop neighborhood of s, one can define a *localized team reward function for team* s from $\mathcal{P}^{N\cdot\mu(s)}(\mathcal{A})$ to $\mathbb{R}$ as

$$r_s(\mu(\mathcal{N}_s), h(s)) = \sum_{a\in\mathcal{A}} r(s, \mu(\mathcal{N}_s), a)h(s)(a), \tag{4.3.6}$$

which depends on the state s and its 1-hop neighborhood; and define the maximal expected discounted accumulative localized team rewards over all *teams* as

$$\widetilde{V}(\mu) := \sup_{\Pi\in\mathfrak{U}} \widetilde{V}^{\Pi}(\mu) = \sup_{\Pi\in\mathfrak{U}} \mathbb{E}\left[\sum_{t=0}^{\infty}\sum_{s\in\mathcal{S}} \gamma^t\, r_s(\mu_t(\mathcal{N}_s), h_t(s)) \,\middle|\, \mu_0 = \mu\right]. \tag{4.3.7}$$

With all these key elements, one can establish the equivalence between maximizing the reward averaged over all *agents* in (MF-MARL) and maximizing the localized team reward summed over all *teams* in (4.3.7), and can thus reformulate the (MF-MARL) problem as an equivalent MDP of (4.3.2)-(4.3.7) with $|\mathcal{S}|$ teams, the latter denoted as (MF-DEC-MARL). That is,

Lemma 4.3.1 *(Value function and Q-function decomposition)*

$$V(\mu) = \widetilde{V}(\mu) = \sup_{\Pi\in\mathfrak{U}} \sum_{s\in\mathcal{S}} \widetilde{V}_s^{\Pi}(\mu), \tag{4.3.8}$$

where $h_t \sim \Pi(\cdot \,|\, \mu_t)$, $\mu_{t+1} \sim \mathbf{P}^N(\cdot \,|\, \mu_t, h_t)$, *and*

$$\widetilde{V}_s^{\Pi}(\mu) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t\, r_s(\mu_t(\mathcal{N}_s), h_t(s)) \,\middle|\, \mu_0 = \mu\right] \tag{4.3.9}$$

is called the value function under policy Π *for team* s. *Similarly,*

$$Q^{\Pi}(\mu, h) := \mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t \sum_{s\in\mathcal{S}} r_s(\mu_t(\mathcal{N}_s), h_t(s)) \,\middle|\, \mu_0 = \mu, h_0 = h\right] = \sum_{s\in\mathcal{S}} Q_s^{\Pi}(\mu, h), \tag{4.3.10}$$

where

$$Q_s^{\Pi}(\mu, h) = \mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t r_s(\mu_t(\mathcal{N}_s), h_t(s)) \,\middle|\, \mu_0 = \mu, h_0 = h\right], \tag{4.3.11}$$

is the Q-function under policy Π *for team* s, *called team-decentralized Q-function.*

Proof of Lemma 4.3.1 The goal is to show that $V(\mu) = \widetilde{V}(\mu)$, with the former the value function of (MF-MARL) subject to the transition probability P defined in (4.2.7) under a given individual policy $\pi \in \mathfrak{u}$, and the latter the value function of (4.3.7) subject to the joint transition probability $\mathbf{P}^N$ defined in (4.3.5) under the policy $\Pi \in \mathfrak{U}$. The proof consists of two steps. Step 1 shows that $V(\mu)$ can be reformulated as a *measured-valued* Markov decision

problem. Step 2 shows that the measured-valued Markov decision problem from Step 1 is equivalent to $\widetilde{V}(\mu)$ in (4.3.7).

Step 1: Recall that $\mu_{t+1} := \frac{1}{N}\sum_{i=1}^{N}\delta_{s_{t+1}^i}$ with s_{t+1}^i subject to (4.2.7). First, one can show that μ_t is a measure-valued Markov decision process under π. To see this, denote $\mathcal{F}_t^s = \sigma(s_t^1, \cdots, s_t^N)$ as the σ-algebra generated by $s_t^1, \cdots, s_t^N$. Then it suffices to show

$$\mathbb{P}(\mu_{t+1} \mid \sigma(\mu_t) \vee \mathcal{F}_t^s) = \mathbb{P}(\mu_{t+1} \mid \sigma(\mu_t)), \quad \mathbb{P}-a.s.. \tag{4.3.12}$$

Following similar arguments for Lemma 2.3.1 and Proposition 2.3.3 in [46], (4.3.12) holds due to the exchangeability of the individual transition dynamics (4.2.7) under π. (4.3.12) implies that there exists a joint transition probability induced from (4.2.7) under π, denoted as $\widetilde{\mathbf{P}}^N$ such that

$$\mu_{t+1} \sim \widetilde{\mathbf{P}}^N(\cdot \mid \mu_t, \pi). \tag{4.3.13}$$

Meanwhile, rewrite $V^\pi(\mu)$ in (MF-MARL) by regrouping the agents according to their states

$$\begin{aligned} V^\pi(\mu) &:= \mathbb{E}\Big[\sum_{t=0}^{\infty}\gamma^t \sum_{i=1}^{N}\frac{1}{N} r(s_t^i, \mu_t(\mathcal{N}_{s_t^i}), a_t^i)\,\Big|\,\mu_0 = \mu\Big], \\ &= \mathbb{E}\Big[\sum_{t=0}^{\infty}\gamma^t \sum_{s\in\mathcal{S}}\mu_t(s)\sum_{a\in\mathcal{A}} r(s, \mu_t(\mathcal{N}_s), a)\pi(s, \mu_t(s))(a)\,\Big|\,\mu_0 = \mu\Big]. \end{aligned} \tag{4.3.14}$$

We see (4.2.7)-(MF-MARL) is reformulated in an equivalent form of (4.3.13)-(4.3.14).

Step 2: It suffices to show that (4.3.13) under π is the same as (4.3.5) under Π and that V^π in (4.3.14) equals to $\widetilde{V}^\Pi$ in (4.3.7). To see this, denote $\langle g, \mu\rangle = \sum_{s\in\mathcal{S}} g(s)\mu(s)$ for any measurable bounded function $g : \mathcal{S} \to \mathbb{R}$, then

$$\begin{aligned} &\mathbb{E}\big[\langle g, \mu_{t+1}\rangle \mid \sigma(\mu_t)\big] \\ &= \frac{1}{N}\mathbb{E}\Big[\sum_{i=1}^{N}\mathbb{E}\big[g(s_{t+1}^j) \mid \sigma(\mu_t) \vee \mathcal{F}_t^s\big]\Big] \\ &= \frac{1}{N}\sum_{s'\in\mathcal{S}}\sum_{i=1}^{N}\sum_{a\in\mathcal{A}} g(s')P(s' \mid s_t^i, \mu_t(\mathcal{N}(s_t^i)), a)\pi(s_t^i, \mu_t(s_t^i))(a) \\ &= \frac{1}{N}\sum_{s'\in\mathcal{S}} g(s')\sum_{s\in\mathcal{S}}\sum_{i=1}^{N}\mathbb{1}(s_t^i = s)\sum_{a\in\mathcal{A}} P(s' \mid s_t^i, \mu_t(\mathcal{N}(s_t^i)), a)\pi(s_t^i, \mu_t(s_t^i))(a) \\ &= \sum_{s'\in\mathcal{S}} g(s')\sum_{s\in\mathcal{S}}\mu_t(s)\sum_{a\in\mathcal{A}} P(s' \mid s, \mu_t(\mathcal{N}(s)), a)\pi(s, \mu_t(s))(a) \\ &= \sum_{s'\in\mathcal{S}} g(s')\sum_{s\in\mathcal{S}}\mu_t(s)\sum_{h\in\mathcal{P}^{N\cdot\mu_t(s)}(\mathcal{A})}\Pi(h \mid \mu_t(s))\sum_{a\in\mathcal{A}} P(s' \mid s, \mu_t(\mathcal{N}(s)), a)h(s)(a), \end{aligned} \tag{4.3.15}$$

where in the last step, the expectation of random variable $h(s)(a)$ with respect to distribution $\Pi(h \mid \mu)$ is $\pi(s, \mu_t(s))$. And from the last equality, clearly μ_{t+1} evolves according to transition

dynamics $\mathbf{P}^N(\cdot|\mu_t, h_t)$ under $\Pi(h_t \mid \mu_t)$. This implies the equivalence of (4.3.13) and (4.3.5). As a byproduct, when taking $g(s') = \mathbb{1}(s' = s^o)$ for any fixed $s^o \in \mathcal{S}$, (4.3.15) becomes

$$\mathbb{E}\big[\mu_{t+1}(s^o)|\sigma(\mu_t)\big] = \sum_{s\in\mathcal{N}(s^o)} \mu_t(s) \sum_{h\in\mathcal{P}^{N\cdot\mu_t(s)}(\mathcal{A})} \Pi(h \mid \mu_t(s)) \sum_{a\in\mathcal{A}} P(s^o \mid s, \mu_t(\mathcal{N}(s)), a)h(s)(a),$$

where the local structure (4.2.7) is used. This suggests that $\mu_{t+1}(s^o)$ only depends on $\mu_t(\mathcal{N}^2_{s^o})$ and $h_t(\mathcal{N}_{s^o})$ since $\mathcal{N}(s) = \mathcal{N}^2(s^o)$ for $s \in \mathcal{N}(s^o)$.
Now we show that $V^\pi(\mu)$ in (4.3.14) and $\widetilde{V}^\Pi(\mu)$ in (4.3.7) are equal. Take $\widetilde{V}^\Pi$ defined in (4.3.7),

$$\begin{aligned}
&\widetilde{V}^\Pi(\mu)\\
=\ &\mathbb{E}_{h_t\sim\Pi(\cdot\,|\,\mu_t),\,\mu_{t+1}\sim\mathbf{P}^N(\cdot\,|\,\mu_t,h_t)}\Big[\sum_{t=0}^{\infty}\sum_{s\in\mathcal{S}}\gamma^t\, r_s(\mu_t(\mathcal{N}_s), h_t)\,\Big|\mu_0=\mu\Big]\\
=\ &\mathbb{E}_{\mu_{t+1}\sim\mathbf{P}^N(\cdot\,|\,\mu_t,h_t)}\Big[\sum_{t=0}^{\infty}\gamma^t\sum_{s\in\mathcal{S}}\mathbb{E}_{h_t\sim\Pi(\cdot\,|\,\mu_t)}\big[r_s(\mu_t(\mathcal{N}_s), h_t)|\mu_t\big]\,\Big|\mu_0=\mu\Big]\\
=\ &\mathbb{E}_{\mu_{t+1}\sim\mathbf{P}^N(\cdot\,|\,\mu_t,h_t)}\Big[\sum_{t=0}^{\infty}\gamma^t\sum_{s\in\mathcal{S}}\sum_{h_t\in\mathcal{P}^{N\cdot\mu_t(s)}(\mathcal{A})} r_s(\mu_t(\mathcal{N}_s), h_t(s))\Pi(h;\pi)\,\Big|\mu_0=\mu\Big]\\
=\ &\mathbb{E}_{\mu_{t+1}\sim\mathbf{P}^N(\cdot\,|\,\mu_t,h_t)}\Big[\sum_{t=0}^{\infty}\gamma^t\sum_{s\in\mathcal{S}}\mu_t(s)\sum_{h_t\in\mathcal{P}^{N\cdot\mu_t(s)}(\mathcal{A})}\Pi(h_t \mid \mu_t)\sum_{a\in\mathcal{A}} r(s,\mu_t(\mathcal{N}_s),a)h(a)\,\Big|\mu_0=\mu\Big]\\
=\ &\mathbb{E}_{\mu_{t+1}\sim\widetilde{\mathbf{P}}^N(\cdot\,|\,\mu_t,\pi)}\Big[\sum_{t=0}^{\infty}\gamma^t\sum_{s\in\mathcal{S}}\mu_t(s)\sum_{a\in\mathcal{A}} r(s,\mu_t(\mathcal{N}_s),a)\pi_t(s,\mu_t(s))(a)\,\Big|\mu_0=\mu\Big]\\
=\ &V^\pi(\mu),
\end{aligned}$$

where in the last second step, $\mathbf{P}^N$ under π is equivalent to $\widetilde{\mathbf{P}}^N$ under Π, and the expectation of $h_t(s)(a)$ with distribution $\Pi(h_t \mid \mu_t)$ is $\pi(s, \mu_t(s))(a)$ such that

$$\begin{aligned}
\sum_{h\in\mathcal{P}^{N\cdot\mu_t(s)}(\mathcal{A})}\Pi(h_t \mid \mu_t)\sum_{a\in\mathcal{A}} r(s,\mu_t(\mathcal{N}_s),a)h(a) &= \mathbb{E}_{h\sim\Pi(\cdot|\mu_t)}\Big[\sum_{a\in\mathcal{A}} r(s,\mu_t(\mathcal{N}_s,a)h(a)\Big]\\
&= \sum_{a\in\mathcal{A}} r(s,\mu_t(\mathcal{N}_s),a)\pi_t(s,\mu_t(s))(a).
\end{aligned}$$

Finally, the decomposition of $\widetilde{V}(\mu)$ and $Q^{\Pi^\theta}(\mu, h)$ according to the states is straightforward. □

The decomposition for the Q-function in (4.3.10) is one of the key elements to allow for approximation of $Q^\Pi_s(\mu, h)$ by a truncated Q-function defined on a smaller space and updated in a localized fashion; it is useful for designing sample-efficient learning algorithms and for parallel computing, as will be clear in the next Section 4.3.3.

4.3.2 Bellman equation for Q-function.

This section builds the second block for reinforcement learning algorithms, the Bellman equation for Q-function. Indeed, the Bellman equation for $Q(\mu, h)$ can be derived following a similar argument in Chapter 2, after establishing the dynamic programming principle on an appropriate probability measure space.

Lemma 4.3.2 *(Bellman Equation for Q-function) The Q-function defined in (4.2.9) satisfies:*

$$\begin{aligned} Q(\mu, h) &= \mathbb{E}\left[\sum_{i=1}^{N} \frac{1}{N} r(s_0^i, \mu(\mathcal{N}_{s_0^i}), a_0^i) \middle| \boldsymbol{s}_0, \boldsymbol{a}_0\right] \\ &\quad + \gamma \mathbb{E}_{s_1^i \sim P\left(\cdot \middle| s_0^i,\, \mu(\mathcal{N}_{s_0^i}),\, a_0^i\right)} \left[\sup_{h' \in \mathcal{H}^N(\mu_1)} Q\left(\mu_1, h'\right)\right]. \end{aligned} \tag{4.3.16}$$

with $\mu_1(\cdot) = \frac{\sum_{i=1}^N \mathbf{1}(s_1^i = \cdot)}{N}$ *the empirical state distribution at time* 1.

Note that the Bellman equation (4.3.16) is for the Q-function defined in (4.2.9) for general mean-field MARL. In order to enable the *localized-training-decentralized-execution* for computational efficiency, one needs to consider the decomposition of Q-function (4.3.10) and the updating rule based on the team-decentralized Q-function (4.3.11). The corresponding Bellman equation for the team-decentralized Q-function (4.3.11) is:

Lemma 4.3.3 *Given a policy* $\Pi \in \mathfrak{U}$, Q_s^Π *defined in (4.3.11) is the unique solution to the Bellman equation* $Q_s^\Pi = \mathcal{T}_s^\Pi Q_s^\Pi$, *with* $\mathcal{T}_s^\Pi$ *the Bellman operator taking the form of*

$$\mathcal{T}_s^\Pi Q_s^\Pi(\mu, h) = \mathbb{E}_{\mu' \sim \mathbf{P}^N(\cdot \,|\, \mu, h),\, h' \sim \Pi(\cdot \,|\, \mu)} \left[r_s(\mu, h) + \gamma \cdot Q_s^\Pi(\mu', h')\right], \forall (\mu, h) \in \Xi. \tag{4.3.17}$$

These Bellman equations are the basis for general Q-function-based algorithms in mean-field MARL.

4.3.3 Exponential Decay of Q-function

This section will show that the team-decentralized Q-function $Q_s^\Pi(\mu, h)$ has an *exponential decay* property. This is another key element to enable an approximation to Q_s^Π by a *localized Q-function* $\widehat{Q}_s^\Pi(\mu(\mathcal{N}_s^k), h(\mathcal{N}_s^k))$, and to guarantee the scalability and sample efficiency of subsequent algorithm design.

To establish the exponential decay property of the Q-function (4.3.11), first recall that $\mathcal{N}_s^k$ is the set of k-hop neighborhood of state s, and define $\mathcal{N}_s^{-k} = \mathcal{S}/\mathcal{N}_s^k$ as the set of states that are outside of s'th k-hop neighborhood. Next, rewrite any given empirical state distribution $\mu \in \mathcal{P}^N(\mathcal{S})$ as $\left(\mu(\mathcal{N}_s^k), \mu(\mathcal{N}_s^{-k})\right)$, and similarly, $h \in \mathcal{H}^N(\mu)$ as $\left(h(\mathcal{N}_s^k), h(\mathcal{N}_s^{-k})\right)$.

Definition 4.3.4 *The Q^{Π} is said to have (c,ρ)-exponential decay property, if for any $s \in \mathcal{S}$ and any $\Pi \in \mathfrak{U}$, $(\mu,h),(\mu',h') \in \Xi$ with $\mu(\mathcal{N}_s^k) = \mu'(\mathcal{N}_s^k)$ and $h(\mathcal{N}_s^k) = h'(\mathcal{N}_s^k)$*

$$\left| Q_s^{\Pi}\left(\mu(\mathcal{N}_s^k), \mu(\mathcal{N}_s^{-k}), h(\mathcal{N}_s^k), h(\mathcal{N}_s^{-k})\right) - Q_s^{\Pi}\left(\mu(\mathcal{N}_s^k), \mu'(\mathcal{N}_s^{-k}), h(\mathcal{N}_s^k), h'(\mathcal{N}_s^{-k})\right) \right| \leq c\rho^{k+1}.$$

Note that the exponential decay property is defined for the team-decentralized Q-function Q_s^{Π}, instead of the centralized Q-function Q^{Π}. The following Lemma provides a sufficient condition for the exponential decay property.

Lemma 4.3.5 *When the reward r_s in (4.3.6) is uniformly upper bounded by $r_{max} > 0$, for any $s \in \mathcal{S}$, Q_s^{Π} satisfies the $\left(\frac{r_{max}}{1-\gamma}, \sqrt{\gamma}\right)$-exponential decay property.*

Proof of Lemma 4.3.5 Let $\mathfrak{P}_{t,s}$ and $\mathfrak{P}'_{t,s}$ be, respectively, distribution of $(\mu_t(\mathcal{N}_s), h_t(s))$ and $(\mu'_t(\mathcal{N}_s), h'_t(s))$ under policy Π^{θ}. By localized transition kernel (4.2.7), it is easy to see that for any given $s \in \mathcal{S}$, $\mu_{t+1}(s)$ only depends on $\mu_t(\mathcal{N}_s^2)$ and $h_t(\mathcal{N}_s)$. Then by the local dependency, (4.3.5) can be rewritten as

$$\mu_{t+1}(s) \sim \mathbf{P}_s^N(\cdot \,|\, \mu_t(\mathcal{N}_s^2), h_t(\mathcal{N}_s)). \tag{4.3.18}$$

Due to the local structure of dynamics (4.3.18) and local dependence of Π^{θ}, the distribution $\mathfrak{P}_{t,s}$, $t \leq \lfloor \frac{k}{2} \rfloor$ only depends on the initial value $(\mu(\mathcal{N}_s^k), h(\mathcal{N}_s^k))$. Therefore, $\mathfrak{P}_{t,s} = \mathfrak{P}'_{t,s}$, $t \leq \lfloor \frac{k}{2} \rfloor$,

$$\begin{aligned}
&\left| Q_s^{\Pi^{\theta}}\left(\mu(\mathcal{N}_s^k), \mu(\mathcal{N}_s^{-k}), h(\mathcal{N}_s^k), h(\mathcal{N}_s^{-k})\right) - Q_s^{\Pi^{\theta}}\left(\mu(\mathcal{N}_s^k), \mu'(\mathcal{N}_s^{-k}), h(\mathcal{N}_s^k), h'(\mathcal{N}_s^{-k})\right) \right| \\
= &\sum_{t=\lfloor \frac{k}{2} \rfloor + 1}^{\infty} \mathbb{E}_{(\mu_t(\mathcal{N}_s), h_t(s)) \sim \mathfrak{P}_{t,s}}\left[r_s(\mu_t(\mathcal{N}_s), h_t(s))\right] - \mathbb{E}_{(\mu'_t(\mathcal{N}_s), h'_t(s)) \sim \mathfrak{P}'_{t,s}}\left[r_s(\mu'_t(\mathcal{N}_s), h'_t(s))\right] \\
\leq &\sum_{t=\lfloor \frac{k}{2} \rfloor + 1}^{\infty} \gamma^t r_{\max} \mathrm{TV}(\mathfrak{P}_{t,s}, \mathfrak{P}'_{t,s}) \leq \frac{r_{\max}}{1-\gamma} \gamma^{\lfloor \frac{k}{2} \rfloor + 1},
\end{aligned}$$

where $\mathrm{TV}(\mathfrak{P}_{t,s}, \mathfrak{P}'_{t,s})$ is total variation between $\mathfrak{P}_{t,s}$ and $\mathfrak{P}'_{t,s}$ that is upper bounded by 1. □

The exponential decay property implies that for a given state $s \in \mathcal{S}$, the dependence of Q_s^{Π} on other states decays quickly with respect to its distance from state s. It motivates and enables the approximation of $Q_s^{\Pi}(\mu,h)$ by a truncated function which only depends on $\mu(\mathcal{N}_s^k)$ and $h(\mathcal{N}_s^k)$, especially when k is large and ρ is small. Specifically, consider the following class

of *localized* Q-functions,

$$\widehat{Q}_s^{\Pi}\Big(\mu(\mathcal{N}_s^k), h(\mathcal{N}_s^k)\Big) = \sum_{\mu(\mathcal{N}_s^{-k}), h(\mathcal{N}_s^{-k})} \Bigg[w_s\Big(\mu(\mathcal{N}_s^{-k}), h(\mathcal{N}_s^{-k}); \mu(\mathcal{N}_s^k), h(\mathcal{N}_s^k)\Big) \cdot Q_s^{\Pi}\Big(\mu(\mathcal{N}_s^k), \mu(\mathcal{N}_s^{-k}), h(\mathcal{N}_s^k), h(\mathcal{N}_s^{-k})\Big)\Bigg], \tag{Local Q-function}$$

where $w_s\left(\mu(\mathcal{N}_s^{-k}), h(\mathcal{N}_s^{-k}); \mu(\mathcal{N}_s^k), h(\mathcal{N}_s^{-k})\right)$ are any non-negative weights of

$$\sum_{\mu(\mathcal{N}_s^{-k}), h(\mathcal{N}_s^{-k})} w_s\Big(\mu(\mathcal{N}_s^{-k}), h(\mathcal{N}_s^{-k}); \mu(\mathcal{N}_s^k), h(\mathcal{N}_s^k)\Big) = 1$$

for any $\mu(\mathcal{N}_s^k)$ and $h(\mathcal{N}_s^k)$.

Then, direct computation yields the following proposition.

Proposition 4.3.6 *Let $\widehat{Q}_s^{\Pi}$ be any localized Q-function in the form of* (Local Q-function)*. Assume the (c, ρ)-exponential decay property in Definition 4.3.4 holds, then for any $\mu \in \mathcal{P}^N(\mathcal{S})$ and $h \in \mathcal{H}^N(\mu)$,*

$$\left|\widehat{Q}_s^{\Pi}\Big(\mu(\mathcal{N}_s^k), h(\mathcal{N}_s^k)\Big) - Q_s^{\Pi}(\mu, h)\right| \leq c\rho^{k+1}. \tag{4.3.19}$$

Moreover, (4.3.19) *holds independent of the weights in* (Local Q-function)*.*

Note that given a team-decentralized Q-function Q_s^{Π}, its localized version $\widehat{Q}_s^{\Pi}$ only takes $\mu(\mathcal{N}_s^k), h(\mathcal{N}_s^k)$ as inputs, and $\widehat{Q}_s^{\Pi}\Big(\mu(\mathcal{N}_s^k), h(\mathcal{N}_s^k)\Big)$ is defined as a weighted average of Q_s^{Π} over all (μ, h)-pairs which agree with $\Big(\mu(\mathcal{N}_s^k), h(\mathcal{N}_s^k)\Big)$ in the k-hop neighborhood of s. Although the localized Q-function $\widehat{Q}_s^{\Pi}$ may vary according to different choices of the weights, by the exponential decay property, every $\widehat{Q}_s^{\Pi}$ approximates Q_s^{Π} with uniform error and requires a smaller dimension of input.

Remark 4.3.7 (Exponential Decay Property) *In a discounted reward setting* (4.2.1)*, the exponential decay property follows directly from the fact that the discount factor $\gamma \in (0, 1)$ and the local dependency structure in* (4.3.2)*-*(4.3.7)*. For problems of finite-time or infinite horizons with ergodic reward functions, this property can be established by imposing additional Lipschitz condition on the transition kernel. (See [142], Theorem 1 for network of heterogeneous agents and $\gamma = 1$).*

4.4 Algorithm Design

The three key analytical components for problem (MF-DEC-MARL) in previous sections pave the way for designing efficient learning algorithms. In this section, we propose and analyze a decentralized neural actor-critic algorithm, called LTDE-NEURAL-AC.

Our focus is the localized Q-function $\widehat{Q}_s^{\Pi}(\mu(\mathcal{N}_s^k), h(\mathcal{N}_s^k))$, the approximation to Q_s^{Π} with a smaller input dimension. First, this localized Q-function $\widehat{Q}_s^{\Pi}$ and the team-decentralized policy Π_s will be parameterized by two-layer neural networks with parameters ω_s and θ_s respectively (Section 4.4.2). Next, these neural network parameters $\theta = \{\theta_s\}_{s\in\mathcal{S}}$ and $\omega = \{\omega_s\}_{s\in\mathcal{S}}$ are updated via an actor-critic algorithm in a *localized fashion* (Section 4.4.3): the critic aims to find a proper estimate for the localized Q-function under a fixed policy (parameterized by θ), while the actor computes the policy gradient based on the localized Q-function, and updates θ by a gradient step.

These networks are updated locally requiring only information of the neighborhood states during the training phase; afterwards agents in the system will execute these learned *decentralized policies* which requires only information of the agent's current state. This *localized training and decentralized execution* enables efficient parallel computing especially for a large shared state space.

Moreover, over-parameterization of neural networks avoids issues of nonconvexity and divergence associated with the neural network approach, and ensures the global convergence of our proposed LTDE-NEURAL-AC algorithm.

4.4.1 Basic Set-up

Policy parameterization. To start, let us assume that at state s the *team-decentralized policy* $\Pi_s^{\theta_s}$ is parameterized by $\theta_s \in \Theta_s$. Further denote $\theta := \{\theta_s\}_{s\in\mathcal{S}}$, $\Theta := \prod_{s\in\mathcal{S}} \Theta_s$, $\Pi^\theta := \prod_{s\in\mathcal{S}} \Pi_s^{\theta_s}$, and $\mathbf{\Pi} := \{\Pi^\theta : \theta \in \Theta\}$ as the class of admissible policies parameterized by the parameter space $\{\theta : \theta \in \Theta\}$.

Initialization. Let us also assume that the initial state distribution μ_0 of N agents is sampled from a given distribution P_0 over $\mathcal{P}^N(\mathcal{S})$, i.e., $\mu_0 \sim P_0$; and define the expected total reward function $J(\theta)$ under policy Π^θ by

$$J(\theta) = \mathbb{E}_{\mu_0\sim P_0}[\widetilde{V}^{\Pi^\theta}(\mu_0)]. \tag{4.4.1}$$

Visitation measure. Denote ν_θ as the stationary distribution on Ξ of the Markov process (4.3.5) induced by Π^θ.

Similar to the single-agent RL problem [2, 60], each admissible policy Π^θ induces a visitation measure $\sigma_\theta(\mu, h)$ on Ξ describing the frequency that policy Π^θ visits (μ, h), with

$$\sigma_\theta(\mu, h) := (1-\gamma)\cdot\sum_{t=0}^{\infty}\gamma^t\cdot\mathbb{P}\left(\mu_t = \mu, h_t = h \mid \Pi^\theta\right), \tag{4.4.2}$$

where $\mu_0 \sim P_0$, $h_t \sim \Pi^\theta(\cdot \,|\, \mu_t)$, and $\mu_{t+1} \sim \mathbf{P}^N(\cdot \,|\, \mu_t, h_t)$.

Policy gradient theorem. In order to find the optimal parameterized policy Π^θ which maximizes the expected total reward function $J(\theta)$, the policy optimization step will search for $\theta \in \Theta$ along the gradient direction $\nabla J(\theta)$. Note that computing the gradient $\nabla J(\theta)$ depends on both the action selection, which is directly determined by Π^θ, and the visitation measure σ_θ in (4.4.2), which is indirectly determined by Π^θ.

A simple and elegant result called the policy gradient theorem (Lemma 4.4.1) proposed in [166], reformulates the gradient $\nabla J(\theta)$ in terms of Q^{Π_θ} in (4.3.10) and $\nabla \log \Pi^\theta(h \,|\, \mu)$ under the visitation measure σ_θ. This result simplifies the gradient computation significantly, and is fundamental for actor-critic algorithms.

Lemma 4.4.1 *[166]* $\nabla J(\theta) = \frac{1}{1-\gamma} \mathbb{E}_{\sigma_\theta} \left[Q^{\Pi^\theta}(\mu, h) \nabla \log \Pi^\theta(h \,|\, \mu) \right]$.

Now, direct implementation of the actor-critic algorithm with the *centralized* policy gradient theorem in Lemma 4.4.1 suffers from high sample complexity due to the dimension of the Q-function. Instead, we will show that the exponential decay property of Q-function allows efficient approximation of the policy gradient via *localization* and hence a *scalable* algorithm to solve (MF-MARL).

4.4.2 Neural Policy and Neural Q-function

We now turn to the localized Q-function $\widehat{Q}_s^\Pi(\mu(\mathcal{N}_s^k), h(\mathcal{N}_s^k))$ (i.e., the approximation of Q_s^Π) and the team-decentralized policy Π_s, and their parameterization by two-layer neural networks. We emphasize that the parameterization framework in this section can be extended to any neural-based single-agent algorithms with convergence guarantee.

Two-layer neural network. For any input space $\mathcal{X} \subset \mathbb{R}^{d_x}$ with dimension $d_x \in \mathbb{N}$, a two-layer neural network $\widetilde{f}(x; W, b)$ with input $x \in \mathcal{X}$ and width $M \in \mathbb{N}$ takes the form of

$$\widetilde{f}(x; W, b) = \frac{1}{\sqrt{M}} \sum_{m=1}^{M} b_m \cdot \mathrm{ReLU}\left(x \cdot [W]_m\right). \tag{4.4.3}$$

Here the scaling factor $\frac{1}{\sqrt{M}}$ called the *Xavier initialization* [68] ensures the same input variance and the same gradient variance for all layers; the activation function $\mathrm{ReLU} : \mathbb{R} \to \mathbb{R}$, defined as $\mathrm{ReLU}(u) = \mathbb{1}\{u > 0\} \cdot u$; $\mathrm{b}=\{b_m\}_{m\in[M]}$ and $W = \left([W]_1^\top, \ldots, [W]_M^\top\right)^\top \in \mathbb{R}^{M \times d_x}$ in (4.4.3) are parameters of the neural network.

Taking advantage of the homogeneity of ReLU (i.e., $\mathrm{ReLU}(c \cdot u) = c \cdot \mathrm{ReLU}(u)$ for all $c > 0$ and $u \in \mathbb{R}$), we adopt the usual trick [29, 181, 7] to fix b throughout the training and only to update W in the sequel. Consequently, denote $\widetilde{f}(x; W, b)$ as $f(x; W)$ when $b_m = 1$ is fixed. $[W]_m$ is initialized according to a multivariate normal distribution $N(0, I_{d_x}/d_x)$, where I_{d_x} is the identity matrix of size d_x.

Neural policy. For each $s \in \mathcal{S}$, denote the tuple $\zeta_s = (\mu(s), h(s)) \in \mathbb{R}^{d_{\zeta_s}}$ for notational simplicity, where $d_{\zeta_s} := 1 + |\mathcal{A}|$ is the dimension of ζ_s. Given the input $\zeta_s = (\mu(s), h(s))$ and parameter $W = \theta_s$ in the two-layer neural network $f(\cdot; \theta_s)$ in (4.4.3), the team-decentralized policy $\Pi_s^{\theta_s}$, called the *actor*, is parameterized in the form of an *energy-based policy* ,

$$\Pi_s^{\theta_s}(h(s) \mid \mu(s)) = \frac{\exp[\tau \cdot f((\mu(s), h(s)); \theta_s)]}{\sum_{h'(s) \in \mathcal{P}^{N \cdot \mu(s)}(\mathcal{A})} \exp\left[\tau \cdot f\left((\mu(s), h'(s)); \theta_s\right)\right]}, \tag{4.4.4}$$

where τ is the temperature parameter and f is the energy function.

To study the policy gradient for (4.4.4), let us first define a class of feature mappings that is consistent with the representation of two-layer neural networks. This connection between the gradient of a two-layer ReLU neural network and the feature mapping defined in (4.4.6) is crucial in the convergence analysis of Theorems 4.5.4 and 4.5.11. Specifically, rewrite the two-layer neural network in (4.4.3) as

$$f(\zeta_s; \theta_s) = \frac{1}{\sqrt{M}} \sum_{m=1}^{M} \mathrm{ReLU}\left(\zeta_s^\top [\theta_s]_m\right) = \frac{1}{\sqrt{M}} \sum_{m=1}^{M} \mathbb{1}\left\{\zeta_s^\top [\theta_s]_m > 0\right\} \cdot \zeta_s^\top [\theta_s]_m. := \phi_{\theta_s}(\zeta_s)^\top \theta_s. \tag{4.4.5}$$

Then the feature mapping $\phi_{\theta_s} = \left([\phi_{\theta_s}]_1^\top, \ldots, [\phi_{\theta_s}]_M^\top\right)^\top : \mathbb{R}^{d_{\zeta_s}} \to \mathbb{R}^{M \times d_{\zeta_s}}$ may take the following form:

$$[\phi_{\theta_s}]_m(\zeta_s) = \frac{1}{\sqrt{M}} \cdot \mathbb{1}\left\{\zeta_s^\top [\theta_s]_m > 0\right\} \cdot \zeta_s. \tag{4.4.6}$$

That is, the two-layer neural network $f(\zeta_s; \theta_s)$ may be viewed as the inner product between the feature $\phi_{\theta_s}(\zeta_s)$, and the neural network parameters θ_s. Since $f(\zeta_s; \theta_s)$ is almost everywhere differentiable with respect to θ_s, we see $\nabla_{\theta_s} f(\zeta_s; \theta_s) = \phi_{\theta_s}(\zeta_s)$. It is worth noting that the neural feature setting considered in our framework (4.4.6) is different from the linear feature literature [65, 86]. This is because the feature mapping ϕ_{θ_s} in (4.4.6) depends on θ_s in a nonlinear fashion through the indicator function whereas the linear feature mapping does not depend on the parameter θ.

Furthermore, define a "centered" version of the feature ϕ_{θ_s} such that

$$\Phi(\theta, s, \mu, h) := \phi_{\theta_s}(\mu(s), h(s)) - \mathbb{E}_{h(s)' \sim \Pi_s^{\theta_s}(\cdot \mid \mu(s))}\left[\phi_{\theta_s}(\mu(s), h'(s))\right]. \tag{4.4.7}$$

Note that when policy Π^θ takes the energy-based form (4.4.4), $\Phi = \frac{1}{\tau} \nabla_\theta \log \Pi^\theta$. Therefore,

Lemma 4.4.2 *For any $\theta \in \Theta$, $s \in \mathcal{S}$, $\mu \in \mathcal{P}^N(\mathcal{S})$ and $h \in \mathcal{H}^N(\mu)$, $\|\Phi(\theta, s, \mu, h)\|_2 \leq 2$, and*

$$\nabla_{\theta_s} J(\theta) = \frac{\tau}{1-\gamma} \cdot \mathbb{E}_{\sigma_\theta}\left[Q^{\Pi^\theta}(\mu, h) \cdot \Phi(\theta, s, \mu, h)\right]. \tag{4.4.8}$$

Moreover, for each $s \in \mathcal{S}$, define the following localized policy gradient

$$g_s(\theta) = \frac{\tau}{1-\gamma} \mathbb{E}_{\sigma_\theta}\left[\left[\sum_{y \in \mathcal{N}_s^k} \widehat{Q}_y^{\Pi^\theta}(\mu(\mathcal{N}_y^k), h(\mathcal{N}_y^k)\right] \cdot \Phi(\theta, s, \mu, h)\right], \tag{4.4.9}$$

with $\widehat{Q}_s^{\Pi^\theta}$ *in* (Local Q-function) *satisfying the* (c,ρ)*-exponential decay property, then there exists a universal constant* $c_0 > 0$ *such that*

$$\|g_s(\theta) - \nabla_{\theta_s} J(\theta)\| \leq \frac{c_0 \tau |\mathcal{S}|}{1-\gamma} \rho^{k+1}. \tag{4.4.10}$$

Proof of Lemma 4.4.2 For any $\theta \in \Theta$, $s \in \mathcal{S}$, $\mu \in \mathcal{P}^N(\mathcal{S})$ and $h \in \mathcal{H}^N(\mu)$, it is easy to verify that $\|\Phi(\theta, s, \mu, h)\|_2 \leq \|\zeta_s\|_2 \leq 2$, by the definitions of the feature mapping ϕ in (4.4.6) and the center feature mapping Φ in (4.4.7).

To prove (4.4.8), note that by Lemma 4.4.1 & the definition of energy-based policy $\Pi_s^{\theta_s}$ (4.4.4),

$$\begin{aligned}
\nabla_{\theta_s} \log \Pi_s^{\theta_s}(h(s) \mid \mu(s)) &= \tau \cdot \nabla_{\theta_s} f((\mu(s), h(s)); \theta_s) - \tau \cdot \mathbb{E}_{h(s)' \sim \Pi^{\theta_s}(\cdot|\mu(s))}[\nabla_{\theta_s} f(\mu(s), h'(s))] \\
&= \tau \cdot \phi_{\theta_s}(\mu(s), h(s)) - \tau \cdot \mathbb{E}_{h(s)' \sim \Pi^{\theta_s}(\cdot|\mu(s))}[\phi_{\theta_s}(\mu(s), h(s))] \\
&= \tau \cdot \Phi(\theta, s, \mu, h).
\end{aligned}$$

The second equality follows from the fact that $\nabla_{\theta_s} f((\mu(s), h(s)); \theta_s) = \phi_{\theta_s}(\mu(s), h(s))$. Therefore,

$$\nabla_{\theta_s} J(\theta) = \frac{\tau}{1-\gamma} \mathbb{E}_{\sigma_\theta}\left[Q^{\Pi^\theta}(\mu, h) \cdot \Phi(\theta, s, \mu, h)\right] = \frac{\tau}{1-\gamma} \mathbb{E}_{\sigma_\theta}\left[\sum_{y \in \mathcal{S}} Q_y^{\Pi^\theta}(\mu, h) \cdot \Phi(\theta, s, \mu, h)\right],$$

where the second equality is by the decomposition of Q-function in Lemma 4.3.1.

The proof of (4.4.9) is based on the exponential decay property in Definition 4.3.4. Notice that

$$\begin{aligned}
g_s(\theta) &= \frac{1}{1-\gamma} \mathbb{E}_{\sigma_\theta}\left[\left[\sum_{y \in \mathcal{N}_s^k} \widehat{Q}_y^{\Pi^\theta}(\mu(\mathcal{N}_y^k), h(\mathcal{N}_y^k)\right] \nabla_{\theta_s} \log \Pi^{\theta_s}(h(s) \mid \mu(s))\right] \\
&= \frac{1}{1-\gamma} \mathbb{E}_{\sigma_\theta}\left[\left[\sum_{y \in \mathcal{S}} \widehat{Q}_y^{\Pi^\theta}(\mu(\mathcal{N}_y^k), h(\mathcal{N}_y^k)\right] \nabla_{\theta_s} \log \Pi^{\theta_s}(h(s) \mid \mu(s))\right].
\end{aligned} \tag{4.4.11}$$

This is because for all $y \notin \mathcal{N}_s^k$, $\widehat{Q}_y^{\Pi^\theta}(\mu(\mathcal{N}_y^k), h(\mathcal{N}_y^k)$ is independent of s. Consequently,

$$\mathbb{E}_{\sigma_\theta}\left[\left[\sum_{y \notin \mathcal{N}_s^k} \widehat{Q}_y^{\Pi^\theta}(\mu(\mathcal{N}_y^k), h(\mathcal{N}_y^k)\right] \nabla_{\theta_s} \log \Pi^{\theta_s}(h(s) \mid \mu(s))\right] = 0.$$

Given Lemma 4.4.1 and (4.4.11), we have the following bound:

$$\begin{aligned}
&\|g_s(\theta) - \nabla_{\theta_s} J(\theta)\|_2 \\
\leq\ & \frac{1}{1-\gamma} \sum_{y \in \mathcal{S}} \sup_{\substack{\mu \in \mathcal{P}^N(\mathcal{S}), \\ h \in \mathcal{H}^N(\mu)}} \left[\left|\widehat{Q}_y^{\Pi^\theta}\left(\mu(\mathcal{N}_y^k), h(\mathcal{N}_y^k)\right) - Q_y^{\Pi^\theta}(\mu, h)\right| \cdot \|\nabla_{\theta_s} \log \Pi^{\theta_s}(h(s) \mid \mu(s))\|_2\right] \\
\leq\ & \frac{c_0 \tau |\mathcal{S}|}{1-\gamma} \rho^{k+1}.
\end{aligned}$$

The last inequality follows from (4.3.19) and $\|\log \Pi^{\theta_s}(h(s) \mid \mu(s))\|_2 = \|\Phi(\theta, s, \mu, h)\|_2 \leq 2$ for any $\mu \in \mathcal{P}^N(\mathcal{S}), h \in \mathcal{H}^N(\mu)$. □

Neural Q-function. Note $\widehat{Q}_s^{\Pi^\theta}$ in (Local Q-function) is unknown *a priori*. To obtain the localized policy gradient (4.4.9), the neural network (4.4.3) to parameterize $\widehat{Q}_s^{\Pi^\theta}$ is taken as:

$$Q_s(\mu(\mathcal{N}_s^k), h(\mathcal{N}_s^k); \omega_s) = f((\mu(\mathcal{N}_s^k), h(\mathcal{N}_s^k)); \omega_s).$$

This Q_s is called the *critic*. For simplicity, denote $\zeta_s^k = (\mu(\mathcal{N}_s^k), h(\mathcal{N}_s^k))$, with $d_{\zeta_s^k}$ the dimension of ζ_s^k.

4.4.3 Actor-Critic

Critic update. For a fixed policy Π^θ, it is to estimate $\widehat{Q}_s^{\Pi^\theta}$ of (Local Q-function) by a two-layer neural network $Q_s(\cdot\,; \omega_s)$, where $\widehat{Q}_s^{\Pi^\theta}$ serves as an approximation to the team-decentralized Q-function $Q_s^{\Pi^\theta}$.

To design the update rule for $\widehat{Q}_s^{\Pi^\theta}$, note that the Bellman equation (4.3.17) is for $Q_s^{\Pi^\theta}$ instead of $\widehat{Q}_s^{\Pi^\theta}$. Indeed, $Q_s^{\Pi^\theta}$ takes (μ, h) as the input while $\widehat{Q}_s^{\Pi^\theta}$ takes the partial information $(\mu(\mathcal{N}_s^k), h(\mathcal{N}_s^k))$ as the input.

In order to update parameter ω_s, we substitute $(\mu(\mathcal{N}_s^k), h(\mathcal{N}_s^k))$ for the state-action pair in the Bellman equation (4.3.17). It is therefore necessary to study the error of using $(\mu(\mathcal{N}_s^k), h(\mathcal{N}_s^k))$ as the input. Specifically, given a tuple $(\mu_t, h_t, r_s(\mu_t(\mathcal{N}_s), h_t(s)), \mu_{t+1}, h_{t+1})$ sampled from the stationary distribution ν_θ of adopting policy Π^θ, the parameter ω_s will be updated to minimize the error:

$$(\delta_{s,t})^2 = \left[Q_s(\mu_t(\mathcal{N}_s^k), h_t(\mathcal{N}_s^k); \omega_s) - r_s(\mu_t(\mathcal{N}_s), h_t(s)) - \gamma \cdot Q_s(\mu_{t+1}(\mathcal{N}_s^k), h_{t+1}(\mathcal{N}_s^k); \omega_s)\right]^2.$$

Estimating $\delta_{s,t}$ depends only on $\mu_t(\mathcal{N}_s^k), h_t(\mathcal{N}_s^k)$ and can be *collected locally*. (See Theorem 4.5.4).

The neural critic update takes the iterative forms of

$$\omega_s(t+1/2) \leftarrow \omega_s(t) - \eta_{\text{critic}} \cdot \delta_{s,t} \cdot \nabla_{\omega_s} Q_s(\mu_t(\mathcal{N}_s^k), h_t(\mathcal{N}_s^k); \omega_s), \tag{4.4.12}$$

$$\omega_s(t+1) \leftarrow \arg\min_{\omega \in \mathcal{B}_s^{\text{critic}}} \|\omega - \omega_s(t+1/2)\|_2, \tag{4.4.13}$$

$$\bar{\omega}_s \leftarrow (t+1)/(t+2) \cdot \bar{\omega}_s + 1/(t+2) \cdot \omega_s(t+1), \tag{4.4.14}$$

in which η_{critic} is the learning rate. Here (4.4.12) is the stochastic semigradient step, (4.4.13) is a projection to the parameter space $\mathcal{B}_s^{\text{critic}} := \left\{\omega_s \in \mathbb{R}^{M \times d_{\zeta_s^k}} : \|\omega_s - \omega_s(0)\|_\infty \leq R/\sqrt{M}\right\}$ for some $R > 0$, and (4.4.14) is the averaging step. This critic update is summarized in Algorithm 4.

Algorithm 4 Localized-Training-Decentralized-Execution Neural Temporal Difference

Input: Width of the neural network M, radius of the constraint set R, number of iterations T_{critic}, policy $\Pi^\theta = \{\Pi_s^{\theta_s}\}_{s\in\mathcal{S}}$, learning rate η_{critic}, localization parameter k.
Initialize: For all $m \in [M]$ and $s \in \mathcal{S}$, sample $b_m \sim \text{Unif}(\{-1,1\})$, $[\omega_s(0)]_m \sim N\left(0, I_{d_{\zeta_s^k}}/d_{\zeta_s^k}\right)$, $\bar{\omega}_s = \omega_s(0)$.
for $t = 0$ to $T_{\text{critic}} - 2$ **do**
  Sample $(\mu_t, h_t, \{r_s(\mu_t(\mathcal{N}_s), h_t(s))\}_{s\in\mathcal{S}}, \mu_t', h_t')$ from the stationary distribution ν_θ of Π^θ.
  for $s \in \mathcal{S}$ **do**
    Denote $\zeta_{s,t}^k = (\mu_t(\mathcal{N}_s^k), h_t(\mathcal{N}_s^k))$, $\zeta_{s,t}^{k\,\prime} = (\mu_t'(\mathcal{N}_s^k), h_t'(\mathcal{N}_s^k))$.
    Residual calculation: $\delta_{s,t} \leftarrow Q_s(\zeta_{s,t}^k; \omega_s(t)) - r_s(\mu_t(\mathcal{N}_s), h_t(s)) - \gamma \cdot Q_s(\zeta_{s,t}^{k\,\prime}; \omega_s(t))$.
    Temporal difference update:
    $\omega_s(t+1/2) \leftarrow \omega_s(t) - \eta_{\text{critic}} \cdot \delta_{s,t} \cdot \nabla_{\omega_s} Q_s(\zeta_{s,t}^k; \omega_s(t))$.
    Projection onto the parameter space: $\omega_s(t+1) \leftarrow \underset{\omega\in\mathcal{B}_s^{\text{critic}}}{\arg\min} \|\omega - \omega_s(t+1/2)\|_2$.
    Averaging the output: $\bar{\omega}_s \leftarrow \frac{t+1}{t+2} \cdot \bar{\omega}_s + \frac{1}{t+2} \cdot \omega_s(t+1)$.
  end for
end for
Output: $Q_s(\,\cdot\,; \bar{\omega}_s), \forall s \in \mathcal{S}$.

Actor update. At the iteration step t, a neural network estimation $Q_s(\,\cdot\,; \bar{\omega}_s)$ is given for the localized Q-function $\widehat{Q}_s^{\Pi^{\theta(t)}}$ under the current policy $\Pi^{\theta(t)}$. Let $\{(\mu_l, h_l)\}_{l\in[B]}$ be samples from the state-action visitation measure $\sigma_{\theta(t)}$ of (4.4.2), and define an estimator $\widehat{\Phi}(\theta, s, \mu_l, h_l)$ of $\Phi(\theta, s, \mu_l, h_l)$ in (4.4.7):

$$\widehat{\Phi}(\theta, s, \mu_l, h_l) = \phi_{\theta_s}(\mu_l(s), h_l(s)) - \mathbb{E}_{\Pi_s^{\theta_s}}\left[\phi_{\theta_s}\left(\mu_l(s), h'(s)\right)\right].$$

By Lemma 4.4.2, one can compute the following estimator of $g_s(\theta(t))$ defined in (4.4.9),

$$\widehat{g}_s(\theta(t)) = \frac{\tau}{(1-\gamma)B} \sum_{l\in[B]} \left[\left[\sum_{y\in\mathcal{N}_s^k} Q_y\left(\mu_l(\mathcal{N}_y^k), h_l(\mathcal{N}_y^k); \bar{\omega}_y\right) \right] \cdot \widehat{\Phi}(\theta(t), s, \mu_l, h_l) \right]. \tag{4.4.15}$$

This estimators $\widehat{g}_s$ in (4.4.15) only *depends locally* on $\{(\mu_l, h_l)\}_{l\in[B]}$. Hence $\widehat{g}$ and $\widehat{\Phi}$ can be computed in a *localized fashion* after the samples are collected. Similar to the critic update, $\theta_s(t)$ is updated by performing a gradient step with $\widehat{g}_s$, and then projected onto the parameter space $\mathcal{B}_s^{\text{actor}} := \left\{\theta_s \in \mathbb{R}^{M\times d_{\zeta_s}} : \|\theta_s - \theta_s(0)\|_\infty \le R/\sqrt{M}\right\}$.

This actor update is summarized in Algorithm 5.

Sampling from ν_θ and the visitation measure σ_θ. In Algorithms 4 and 5, it is assumed that one can sample independently from the stationary distribution ν_θ and the visitation

measure σ_θ, respectively. Such an assumption of sampling from ν_θ can be relaxed by either sampling from a rapidly-mixing Markov chain, with weakly-dependent sequence of samples [23], or by randomly picking samples from replay buffers consisting of long trajectories, with reduced correlation between samples.

To sample from the visitation measure σ_θ and computing the unbiased policy gradient estimator, [92] suggests introducing a new MDP such that the next state is sampled from the transition probability with probability γ, and from the initial distribution with probability $1-\gamma$. Then the stationary distribution of this new MDP is exactly the visitation measure. Alternatively, [111] proposes an importance-sampling-based algorithm which enables off-policy evaluation with low variance.

Algorithm 5 Localized-Training-Decentralized-Execution Neural Actor-Critic

1: **Input**: Width of the neural network M, radius of the constraint set R, number of iterations T_{actor} and T_{critic}, learning rate η_{actor} and η_{critic}, temperature parameter τ, batch size B, localization parameter k.
2: **Initialize**: For all $m \in [M]$ and $s \in \mathcal{S}$, sample $b_m \sim \text{Unif}(\{-1,1\})$, $[\theta_s(0)]_m \sim N\left(0, I_{d_{\zeta_s}}/d_{\zeta_s}\right)$.
3: **for** $t = 1$ to T_{actor} **do**
4: Define the decentralized policy $\Pi_s^{\theta_s}$ for each state $s \in \mathcal{S}$,

$$\Pi_s^{\theta_s}(h(s) \mid \mu(s)) = \frac{\exp[\tau \cdot f((\mu(s), h(s)); \theta_s)]}{\sum_{h'(s)\in\mathcal{H}^N} \exp\left[\tau \cdot f\left((\mu(s), h'(s)); \theta_s\right)\right]}.$$

5: Output $Q_s(\,\cdot\,; \bar{\omega}_s)$ using Algorithm 4 with the inputs: policy $\Pi^\theta = \left\{\Pi_s^{\theta_s}\right\}_{s\in\mathcal{S}}$, width of the neural network M, radius of the constraint set R, number of iterations T_{critic}, learning rate η_{critic} and localization parameter k.
6: Sample $\{\mu_l, h_l\}_{l\in[B]}$ from the state-action visitation measure σ_θ (4.4.2) of Π^θ.
7: **for** $s \in \mathcal{S}$ **do**
8: Compute the local gradient estimator $\widehat{g}_s(\theta(t))$ using (4.4.15).
9: Policy update: $\theta_s(t+1/2) \leftarrow \theta_s(t) + \eta_{\text{actor}} \cdot \widehat{g}_s(\theta(t))$
10: Projection onto the parameter space: $\theta_s(t+1) \leftarrow \underset{\theta\in\mathcal{B}_s^{\text{actor}}}{\arg\min} \|\theta - \theta_s(t+1/2)\|_2$.
11: **end for**
12: **end for**
13: **Output**: $\{\Pi^{\theta(t)}\}_{t\in[T_{\text{actor}}]}$.

4.5 Convergence of the Critic and Actor Updates

We now establish the global convergence for LTDE-NEURAL-AC proposed in Section 4.4. Our analysis of convergence relies on the use of an over-parameterization technique, which involves a two-layer neural network with a large width M. This technique is critical to our analysis, as it allows to address the non-convexity issue in neural network optimization and

to prove the convergence result. Indeed, some commonly used loss functions, such as the mean-square error and the cross-entropy loss, are often neither convex nor concave with respect to neural network parameters. In addition, gradient-based method or other first order algorithms may be trapped at some undesired stationary points due to the non-convex optimization landscape. Meanwhile, it has shown that the training problem in the over-parameterization regime is almost equivalent to a regression problem in a reproducing kernel Hilbert space [6, 7, 211, 38]. In addition, the optimization landscape can also be improved by over-parameterization in the sense that all stationary points are nearly optimal. These key properties of the over-parameterized neural network facilitate our convergence analysis.

Convergence of the critic update. The convergence of the decentralized neural critic update in Algorithm 4 relies on the following assumptions.

Assumption 4.5.1 *(Action-Value Function Class) For each $s \in \mathcal{S}$, $k \in \mathbb{N}$, define*

$$\mathcal{F}_{R,\infty}^{s,k} = \left\{ f(\zeta_s^k) = Q_s(\zeta_s^k; \omega_s(0)) + \int \mathbb{1}\left\{v^\top \zeta_s^k > 0\right\} \cdot (\zeta_s^k)^\top \iota(v)\, d\mu(v) : \|\iota(v)\|_\infty \le R \right\}, \tag{4.5.1}$$

with $\mu : \mathbb{R}^{d_{\zeta_s^k}} \to \mathbb{R}$ the density function of Gaussian distribution $N(0, I_{d_{\zeta_s^k}}/d_{\zeta_s^k})$ and denote $Q_s(\zeta_s^k; \omega_s(0))$ as the two-layer neural network under the initial parameter $\omega_s(0)$. We assume that $\widehat{Q}_s^{\Pi^\theta} \in \mathcal{F}_{R,\infty}^{s,k}$.

Assumption 4.5.2 *(Regularity of ν_θ and σ_θ) There exists a universal constant $c_0 > 0$ such that for any policy Π^θ, any $\alpha \ge 0$, and any $v \in \mathbb{R}^{d_\zeta}$ with $\|v\|_2 = 1$, the stationary distribution ν_θ and the state visitation measure σ_θ satisfy*

$$\mathbb{P}_{\zeta \sim \nu_\theta}\left(\left|v^\top \zeta\right| \le \alpha\right) \le c_0 \cdot \alpha, \quad \mathbb{P}_{\zeta \sim \sigma_\theta}\left(\left|v^\top \zeta\right| \le \alpha\right) \le c_0 \cdot \alpha.$$

Remark 4.5.3 *Both Assumption 4.5.1 and Assumption 4.5.2 are similar to the standard assumptions in the analysis of single-agent neural actor-critic algorithms [29, 110, 181, 38].*

In particular, Assumption 4.5.1 is a regularity condition for $\widehat{Q}_s^{\Pi^\theta}$ in (Local Q-function). Here $\mathcal{F}_{R,\infty}^{s,k}$ is a subset of the reproducing kernel Hilbert space (RKHS) induced by the random feature $\mathbb{1}\left\{v^\top \zeta_s^k > 0\right\} \cdot (\zeta_s^k)$ with $v \sim N(0, I_{d_{\zeta_s^k}}/d_{\zeta_s^k})$ up to the shift of $Q_s(\zeta_s^k; \omega_s(0))$ [144].

This RKHS is dense in the space of continuous functions on any compact set [124, 84]. (See also Section 4.6.1.1 for details of the connection between $\mathcal{F}_{R,\infty}^{s,k}$ and the linearizations of two-layer neural networks (4.6.4)).

Assumption 4.5.2 holds when σ_θ and ν_θ have uniformly upper bounded probability densities [29].

Theorem 4.5.4 *(Convergence of Critic Update) Assume Assumptions 4.5.1 and 4.5.2. Set $T_{\text{critic}} = \Omega(M)$ and $\eta_{\text{critic}} = \min\{(1-\gamma)/8, (T_{\text{critic}})^{-1/2}\}$ in Algorithm 4. Then $Q_s(\cdot\,; \bar{\omega}_s)$*

generated by Algorithm 4 satisfies

$$\mathbb{E}_{\text{init}}\left[\left\|Q_s(\cdot\,;\bar{\omega}_s)-Q_s^{\Pi^\theta}(\cdot)\right\|^2_{L^2(\nu_\theta)}\right]\leq\mathcal{O}\left(\frac{R^3 d_{\zeta_s^k}^{3/2}}{M^{1/2}}+\frac{R^{5/2}d_{\zeta_s^k}^{5/4}}{M^{1/4}}+\frac{r_{max}^2\gamma^{k+1}}{(1-\gamma)^2}\right),\tag{4.5.2}$$

where $\|f\|_{L^2(\nu_\theta)}:=(\mathbb{E}_{\zeta\sim\nu_\theta}[f(\zeta)^2])^{1/2}$*, and the expectation* (4.5.2) *is taken with respect to the random initialization.*

Theorem 4.5.4 indicates the trade-off between the approximation-optimization error and the localization error. The first two terms in (4.5.2) correspond to the neural network approximation-optimization error, similar to the single-agent case [29, 38]. This approximation-optimization error decreases when the width of the hidden layer M increases. Meanwhile, the last term in (4.5.2) represents the additional error from using the localized information in (4.4.12), unique for the mean-field MARL case. This localization error and γ^k decrease as the number of truncated neighborhood k increases, with more information from a larger neighborhood used in the update. However, the input dimension $d_{\zeta_s^k}$ and the approximation-optimization error will increase if the dimension of the problem increases.

In particular, for a relatively sparse network on $\mathcal{S}$, one can choose $k\ll|\mathcal{S}|$ hence $d_{\zeta_s^k}\ll d_\zeta$, and Theorem 4.5.4 indicates the superior performance of the localized training scheme in efficiency over directly approximating the centralized Q-function.

Proof of Theorem 4.5.4 is presented in Section 4.6.1.

Convergence of the actor update. This section establishes the global convergence of the actor update. The convergence analysis consists of two steps. The first step proves the convergence to a stationary point $\widetilde{\theta}$; the second step controls the gap between the stationary point $\widetilde{\theta}$ and the optimality θ^* in the over-parameterization regime. The convergence is built under the following assumptions and definition.

Assumption 4.5.5 *(Variance Upper Bound) For every* $t\in[T_{actor}]$ *and* $s\in\mathcal{S}$*, denote* $\xi_s(t)=\widehat{g}_s(\theta(t))-\mathbb{E}\left[\widehat{g}_s(\theta(t))\right]$ *with* $\widehat{g}_s(\theta(t))$ *defined in* (4.4.15)*. Assume there exists* $\Sigma>0$ *such that* $\mathbb{E}\left[\|\xi_s(t)\|_2^2\right]\leq\tau^2\Sigma^2/B$*. Here the expectations are taken over* $\sigma_{\theta(t)}$ *given* $\{\bar{\omega}_s\}_{s\in\mathcal{S}}$*.*

Assumption 4.5.6 *(Regularity of* $\mathrm{d}\sigma_\theta/\mathrm{d}\nu_\theta$*) There exists an absolute constant* $D>0$ *such that for every* Π^θ*, the stationary distribution* ν_θ *and the state-action visitation measure* σ_θ *satisfy*

$$\left\{\mathbb{E}_{\nu_\theta}\left[(\mathrm{d}\sigma_\theta/\mathrm{d}\nu_\theta(\mu,h))^2\right]\right\}\leq D^2,$$

where $\mathrm{d}\sigma_\theta/\mathrm{d}\nu_\theta$ *is the Radon-Nikodym derivative of* σ_θ *with respect to* ν_θ*.*

Assumption 4.5.7 *(Lipschitz Continuous Policy Gradient) There exists an absolute constant* $L>0$*, such that* $\nabla_\theta J(\theta)$ *is* L*-Lipschitz continuous with respect to* θ*, i.e., for all* θ_1*,* θ_2*,*

$$\|\nabla_\theta J(\theta_1)-\nabla_\theta J(\theta_2)\|_2\leq L\cdot\|\theta_1-\theta_2\|_2.$$

Definition 4.5.8 *$\widetilde{\theta} \in \mathcal{B}^{actor}$ is called a stationary point of $J(\theta)$ if for all $\theta \in \mathcal{B}^{actor}$,*

$$\nabla_\theta J(\widetilde{\theta})^\top (\theta - \widetilde{\theta}) \leq 0. \tag{4.5.3}$$

Meanwhile, $\theta^* \in \mathcal{B}^{actor}$ is called an optimal point of $J(\theta)$ if

$$\theta^* \in \arg\max_{\theta \in \mathcal{B}^{actor}} J(\theta). \tag{4.5.4}$$

Assumption 4.5.9 *(Policy Function Class) Define a function class*

$$\mathcal{F}_{R,\infty} = \left\{ f(\zeta) = \sum_{s\in\mathcal{S}} \left[\phi_{\theta_s(0)}(\zeta_s)^\top \theta_s(0) + \int \mathbb{1}\left\{ v^\top \zeta_s > 0 \right\} \cdot (\zeta_s)^\top \iota(v)\, d\mu(v) \right] : \|\iota(v)\|_\infty \leq R \right\}$$

where $\mu : \mathbb{R}^{d_{\zeta_s}} \to \mathbb{R}$ is the density function of the Gaussian distribution $N\left(0, I_{d_{\zeta_s}}/d_{\zeta_s}\right)$ and $\theta(0)$ is the initial parameter. For any stationary point $\widetilde{\theta}$, define the function

$$u_{\widetilde{\theta}}(\mu, h) := \frac{d\sigma_{\theta^*}}{d\sigma_{\widetilde{\theta}}}(\zeta) - \frac{d\bar{\sigma}_{\theta^*}}{d\bar{\sigma}_{\widetilde{\theta}}}(\mu) + \sum_{s\in\mathcal{S}} \phi_{\widetilde{\theta}_s}(\zeta_s)^\top \widetilde{\theta}_s,$$

with $\bar{\sigma}_\theta$ the state visitation measure under policy Π^θ, and $\frac{\mathrm{d}\sigma_{\theta^*}}{\mathrm{d}\sigma_{\widetilde{\theta}}}, \frac{\mathrm{d}\bar{\sigma}_{\theta^*}}{\mathrm{d}\bar{\sigma}_{\widetilde{\theta}}}$ the Radon-Nikodym derivatives between corresponding measures. We assume that $u_{\widetilde{\theta}} \in \mathcal{F}_{R,\infty}$ for any stationary point $\widetilde{\theta}$.

A few remarks are in place for these Assumption 4.5.5, 4.5.6, 4.5.7 and 4.5.9.

Remark 4.5.10 *All these assumptions are counterparts of standard assumption in the analysis of single-agent policy gradient method [139, 191, 192, 202, 181].*

In particular, Assumption 4.5.5 and Assumption 4.5.6 hold if the Markov chain (4.3.5) mixes sufficiently fast, and the critic $Q_s(\cdot\,; \omega_s)$ has an upper-bounded second moment under $\sigma_{\theta(t)}$ [181]. Note that different from Assumption 4.5.2, where regularity conditions are imposed separately on ν_θ and σ_θ, Assumption 4.5.6 imposes the regularity condition directly on the Radon-Nikodym derivative of σ_θ with respect to ν_θ. This allows the change of measures in the analysis of Theorem 4.5.11. In general, Assumption 4.5.2 does not necessarily imply Assumption 4.5.6.

Assumption 4.5.7 holds when the transition probability and the reward function are both Lipschitz continuous with respect to their inputs [139], or when the reward is uniformly bounded and the score function $\nabla_\theta \Pi^\theta$ is uniformly bounded and Lipschitz continuous with respect to θ [202].

As for Assumption 4.5.9, we first emphasize that $u_{\widetilde{\theta}}(\mu, h)$ is a key element in the proof of Theorem 4.5.11. More specifically, this assumption is motivated by the well-known Performance Difference Lemma [89] in order to characterize the optimality gap of a stationary point $\widetilde{\theta}$. In particular, it guarantees that $u_{\widetilde{\theta}}$ can be decomposed into a sum of local functions depending on ζ_s, and that each local function lies in a rich RKHS (see the discussion after Assumption 4.5.1). Section 4.7 provides a concrete network example that satisfies all Assumptions 4.5.1, 4.5.2, 4.5.5, 4.5.6, 4.5.7 and 4.5.9 (or their mild relaxations).

With all these assumptions, we now establish the rate of convergence for Algorithm 5.

Theorem 4.5.11 *Assume Assumptions 4.5.1, 4.5.2, 4.5.5, 4.5.6, 4.5.7 and 4.5.9. Set* $T_{\text{critic}} = \Omega(M)$, $\eta_{\text{critic}} = \min\{(1-\gamma)/8, (T_{\text{critic}})^{-1/2}\}$, $\eta_{\text{actor}} = (T_{\text{actor}})^{-1/2}, R = \tau = 1$, $M = \Omega\left((f(k)|\mathcal{A}|)^5 (T_{\text{actor}})^8\right)$, $\gamma \leq (T_{\text{actor}})^{-2/k}$, *with* $f(k) := \max_{s\in\mathcal{S}} |\mathcal{N}_s^k|$ *the size of the largest* k*-neighborhood in the graph* $(\mathcal{S}, \mathcal{E})$. *Then, the output* $\{\theta(t)\}_{t\in[T_{\text{actor}}]}$ *of Algorithm 5 satisfies*

$$\min_{t\in[T_{\text{actor}}]} \mathbb{E}\left[J(\theta^*) - J(\theta(t))\right] \leq \mathcal{O}\left(|\mathcal{S}|^{1/2} B^{-1/2} + |\mathcal{S}||\mathcal{A}|^{1/4}\left(\gamma^{k/8} + (T_{\text{actor}})^{-1/4}\right)\right). \tag{4.5.5}$$

Note that the error $\mathcal{O}(\gamma^{k/8}|\mathcal{S}||\mathcal{A}|^{1/4})$ in Theorem 4.5.11, coming from the localized training, decays exponentially fast as k increases and is negligible with a careful choice of k. According to Theorem 4.5.11, Algorithm 5 converges at rate $T_{\text{actor}}^{-1/4}$ with sufficiently large width M and batch size B.

Indeed, Theorem 4.5.11 manages to incorporate the neural network optimization error, which has been analyzed in [29] and [181], with the errors arising from the decentralized and parallel updates of $\{\theta_s(t)\}_{s\in\mathcal{S}}$ and from the truncated Q-functions. It is established by generalizing the techniques for the single-agent setting studied by [29] and [181]. Detailed proof of Theorem 4.5.11 is provided in Section 4.6.2.

Remark 4.5.12 (Convergence to Optimal Decentralized Neural Policy) *By Definition 4.5.8, the policy* Π^{θ^*} *is the optimal decentralized policy within the policy class parameterized by two-layer neural networks, which is a policy class subject to the specific parameterization defined in (4.4.4) and a subset of all possible decentralized policies. The convergence in Theorem 4.5.11 relies on the neural network parameterization and may not necessarily imply the convergence under a different policy class.*

Remark 4.5.13 (Choice of k) *The particular form* $\gamma < (T_{actor})^{-2/k}$ *in Theorem 4.5.11 is not essential and is mainly chosen to highlight the error bound in (4.5.5): if* k *is chosen to be small, the error from estimating the truncated Q-function may become the dominant term in the error bound and hence the leading order of the bound may change accordingly. The detailed error bound without such an inequality can be found in the proof of Theorem 4.5.11 See (4.6.43) in Section 4.6.2.*

4.6 Proof of Convergence Results

4.6.1 Proof of Theorem 4.5.4: Convergence of Critic Update

This section presents the proof of convergence of the decentralized neural critic update. It consists of several steps. Section 4.6.1.1 introduces necessary notations and definitions. Section 4.6.1.2 proves that the critic update minimizes the projected mean-square Bellman error given a two-layer neural network. Section 4.6.1.3 shows that the global minimizer of the

projected mean-square Bellman error converges to the true team-decentralized Q-function as the width of hidden layer $M \to \infty$.

4.6.1.1 Notations

Recall that the set of all state-action (distribution) pairs is denoted as $\Xi := \cup_{\mu \in \mathcal{P}^N(\mathcal{S})} \{\zeta = (\mu, h) : h \in \mathcal{H}^N(\mu)\}$. For any $\zeta = (\mu, h) \in \Xi$, denote the localized state-action (distribution) pair as $\zeta_s^k = (\mu(\mathcal{N}_s^k), h(\mathcal{N}_s^k))$. Meanwhile, denote $\Xi_s^k = \{\zeta_s^k : \zeta \in \Xi\}$ as the set of all possible localized state-action (distribution) pairs. Without loss of generality, assume $\|\zeta_s^k\|_2 \leq 1$ for any $\zeta_s^k \in \Xi_s^k$.

Let d_ζ denote the dimension of the space Ξ. Since $\mathcal{P}^N(\mathcal{S})$ has dimension $(|\mathcal{S}| - 1)$ and $\mathcal{H}^N(\mu)$ has dimension $|\mathcal{S}|(|\mathcal{A}| - 1)$ for any $\mu \in \mathcal{P}^N(\mathcal{S})$, the product space Ξ has dimension $d_\zeta = |\mathcal{S}||\mathcal{A}| - 1$. Similarly, one can see that the dimension of the space Ξ_s^k, denoted by $d_{\zeta_s^k}$, is at most $f(k)|\mathcal{A}|$, where $f(k) := \max_{s \in \mathcal{X}} |\mathcal{N}_s^k|$ is the size of the largest k-neighborhood in the graph $(\mathcal{S}, \mathcal{E})$.

Let $\mathbb{R}^\Xi$ and $\mathbb{R}^{\Xi_s^k}$ be the sets of real-valued square-integrable functions (with respect to ν_θ) on Ξ and Ξ_s^k, respectively. Define the norm $\|\cdot\|_{L^2(\nu_\theta)}$ on $\mathbb{R}^\Xi$ by

$$\| f \|_{L^2(\nu_\theta)} := \left(\mathbb{E}_{\zeta \sim \nu_\theta}[f(\zeta)^2]\right)^{1/2}, \quad \forall f \in \mathbb{R}^\Xi. \tag{4.6.1}$$

Note that for any function $f \in \mathbb{R}^{\Xi_s^k}$, a function $\tilde{f} \in \mathbb{R}^\Xi$ is called a *natural extension* of f if $\tilde{f}(\zeta) = f(\zeta_s^k)$ for all $\zeta \in \Xi$. Since the natural extension is an injective mapping from $\mathbb{R}^{\Xi_s^k}$ to $\mathbb{R}^\Xi$, one can view $\mathbb{R}^{\Xi_s^k}$ as a subset of $\mathbb{R}^\Xi$. In addition for a function $f \in \mathbb{R}^{\Xi_s^k}$, we use the same notation $f \in \mathbb{R}^\Xi$ to denote the natural extension of f.

For any closed and convex function class $\mathcal{F} \subset \mathbb{R}^\Xi$, define the project operator $\text{Proj}_\mathcal{F}$ from $\mathbb{R}^\Xi$ onto $\mathcal{F}$ by

$$\text{Proj}_\mathcal{F}(g) := \arg\min_{f \in \mathcal{F}} \|f - g\|_{L^2(\nu_\theta)}. \tag{4.6.2}$$

This projection operator $\text{Proj}_\mathcal{F}$ is non-expansive in the sense that

$$\|\text{Proj}_\mathcal{F}(f) - \text{Proj}_\mathcal{F}(g)\|_{L^2(\nu_\theta)} \leq \|f - g\|_{L^2(\nu_\theta)}. \tag{4.6.3}$$

Recall that for each state $s \in \mathcal{S}$, the critic parameter ω_s is updated in a localized fashion using information from the k-hop neighborhood of s. Without loss of generality, let us omit the subscript s of ω_s in the following presentation, and the result holds for all $s \in \mathcal{S}$ simultaneously.

Given an initialization $\omega(0) \in \mathbb{R}^{M \times d_{\zeta_s^k}}$, define the following function class

$$\mathcal{F}_{R,M} = \Bigg\{ Q_0(\zeta_s^k; \omega) := \frac{1}{\sqrt{M}} \sum_{m=1}^{M} \mathbb{1}\left\{[\omega(0)]_m^\top \zeta_s^k > 0\right\} \omega_m^\top \zeta_s^k : \omega \in \mathbb{R}^{M \times d_{\zeta_s^k}}, \|\omega - \omega(0)\|_\infty \leq R/\sqrt{M} \Bigg\}. \tag{4.6.4}$$

$Q_0(\,\cdot\,;\omega)$ locally linearizes the neural network $Q(\,\cdot\,;\omega)$ (with respect to ω) at $\omega(0)$. Any function $Q_0(\,\cdot\,;\omega) \in \mathcal{F}_{R,M}$ can be viewed as an inner product between the feature mapping $\phi_{\omega(0)}(\cdot)$ defined in (4.4.6) and the parameter ω, i.e. $Q_0(\,\cdot\,;\omega) = \phi_{\omega(0)}(\cdot)^\top\omega$. In addition it holds that $\nabla_\omega Q_0(\,\cdot\,;\omega) = \phi_{\omega(0)}(\cdot)$. All functions in $\mathcal{F}_{R,M}$ share the same feature mapping $\phi_{\omega(0)}(\cdot)$ which only depends on the initialization $\omega(0)$.

Recall the Bellman operator $\mathcal{T}_s^\theta : \mathbb{R}^\Xi \to \mathbb{R}^\Xi$ defined in (4.3.17),

$$\mathcal{T}_s^\theta Q_s^{\Pi^\theta}(\mu,h) = \mathbb{E}_{\mu'\sim\mathbf{P}^N(\cdot\,|\,\mu,h),\,h'\sim\Pi^\theta(\cdot\,|\,\mu)}\left[r_s(\mu,h) + \gamma\cdot Q_s^{\Pi^\theta}(\mu',h')\right], \forall(\mu,h)\in\Xi.$$

The team-decentralized Q-function $Q_s^{\Pi^\theta}$ in (4.3.10) is the unique fixed point of $\mathcal{T}_s^\theta$: $Q_s^{\Pi^\theta} = \mathcal{T}_s^\theta Q_s^{\Pi^\theta}$. Now given a general parameterized function class $\mathcal{F}$, we aim to learn a $Q_s(\,\cdot\,;\omega)\in\mathcal{F}$ to approximate $Q_s^{\Pi^\theta}$ by minimizing the following projected mean-squared Bellman error (PMSBE):

$$\min_\omega \mathrm{PMSBE}(\omega) = \mathbb{E}_{\zeta\sim\nu_\theta}\left[\left(Q_s(\,\zeta_s^k\,;\omega) - \mathrm{Proj}_{\mathcal{F}}\mathcal{T}_s^\theta Q_s(\,\zeta_s^k\,;\omega)\right)^2\right]. \tag{4.6.5}$$

In the first step of the convergence analysis, we take $\mathcal{F} = \mathcal{F}_{R,M}$ (the locally linearized two-layer neural network defined in (4.6.4)) and consider the following PMSBE:

$$\min_\omega \mathbb{E}_{\zeta\sim\nu_\theta}\left[\left(Q_0(\,\zeta_s^k\,;\omega) - \mathrm{Proj}_{\mathcal{F}_{R,M}}\mathcal{T}_s^\theta Q_0(\,\zeta_s^k\,;\omega)\right)^2\right]. \tag{4.6.6}$$

We will show in Section 4.6.1.2 that the output of Algorithm 4 converges to the global minimizer of (4.6.6).

4.6.1.2 Convergence to the Global Minimizer in $\mathcal{F}_{R,M}$

The following lemma guarantees the existence and the uniqueness of the global minimizer of MSPBE that corresponds to the projection onto $\mathcal{F}_{R,M}$ in (4.6.6).

Lemma 4.6.1 *(Existence and Uniqueness of the Global Minimizer in $\mathcal{F}_{R,M}$) For any $b\in\mathbb{R}^M$ and $\omega(0)\in\mathbb{R}^{M\times d_{\zeta_s^k}}$, there exists an ω^* such that $Q_0(\,\cdot\,;\omega^*)\in\mathcal{F}_{R,M}$ is unique almost everywhere in $\mathcal{F}_{R,M}$ and is the global minimizer of MSPBE that corresponds to the projection onto $\mathcal{F}_{R,M}$ in (4.6.6).*

Proof of Lemma 4.6.1 We first show that the operator $\mathcal{T}_s^\theta : \mathbb{R}^\Xi \to \mathbb{R}^\Xi$ (4.3.17) is a γ-contraction in the $L^2(\nu_\theta)$-norm.

$$\begin{aligned}
&\|\mathcal{T}_s^\theta Q_1 - \mathcal{T}_s^\theta Q_2\|_{L^2(\nu_\theta)}^2 = \mathbb{E}_{\zeta\sim\nu_\theta}\left[\left(\mathcal{T}_s^\theta Q_1(\zeta) - \mathcal{T}_s^\theta Q_2(\zeta)\right)^2\right]\\
=&\,\gamma^2\mathbb{E}_{\zeta\sim\nu_\theta}\left[\left(\mathbb{E}\left[Q_1(\zeta') - Q_2(\zeta')\,\middle|\,\zeta' = (\mu',h'), \mu'\sim P^N(\cdot\,|\,\zeta), h'\sim\Pi^\theta(\cdot\,|\,\mu')\right]\right)^2\right]\\
\le&\,\gamma^2\mathbb{E}_{\zeta\sim\nu_\theta}\left[\mathbb{E}\left[\left(Q_1(\zeta') - Q_2(\zeta')\right)^2\,\middle|\,\zeta' = (\mu',h'), \mu'\sim P^N(\cdot\,|\,\zeta), h'\sim\Pi^\theta(\cdot\,|\,\mu')\right]\right]\\
=&\,\gamma^2\mathbb{E}_{\zeta'\sim\nu_\theta}[(Q_1(\zeta') - Q_2(\zeta'))^2] = \gamma^2\|Q_1 - Q_2\|_{L^2(\nu_\theta)}^2,
\end{aligned}$$

where the first inequality follows from Hölder's inequality for the conditional expectation and the third equality stems from the fact that ζ' and ζ have the same stationary distribution ν_θ.

Meanwhile, the projection operator $\mathrm{Proj}_{\mathcal{F}_{R,M}} : \mathbb{R}^{\Xi} \to \mathcal{F}_{R,M}$ is non-expansive. Therefore, the operator $\mathrm{Proj}_{\mathcal{F}_{R,M}} \mathcal{T}_s^\theta : \mathcal{F}_{R,M} \to \mathcal{F}_{R,M}$ is γ-contraction in the $L^2(\nu_\theta)$-norm. Hence $\mathrm{Proj}_{\mathcal{F}_{R,M}}$ admits a unique fixed point $Q_0(\,\cdot\,;\omega^*) \in \mathcal{F}_{R,M}$. By definition, $Q_0(\,\cdot\,;\omega^*)$ is the global minimizer of MSPBE that corresponds to the projection onto $\mathcal{F}_{R,M}$ in (4.6.6). □

We will show that the function class $\mathcal{F}_{R,M}$ will approximately become $\mathcal{F}_{R,\infty}^{s,k}$ (defined in Assumption 4.5.1) as $M \to \infty$, where $\mathcal{F}_{R,\infty}^{s,k}$ is a rich reproducing kernel Hilbert space (RKHS). Consequently, $Q_0(\,\cdot\,;\omega^*)$ will become the global minimum of the MSPBE (4.6.6) on $\mathcal{F}_{R,\infty}^{s,k}$ given Lemma 4.6.1.

Moreover, by using similar argument and technique developed in [29, Theorem 4.6], we can establish the convergence of Algorithm 4 to $Q_0(\,\cdot\,;\omega^*)$ as the following.

Theorem 4.6.2 *(Convergence to $Q_0(\,\cdot\,;\omega^*)$) Set $\eta_{\text{critic}} = \min\{(1-\gamma)/8, 1/\sqrt{T_{critic}}\}$ in Algorithm 4. Then the output $Q_s(\,\cdot\,;\bar{\omega})$ of Algorithm 4 satisfies*

$$\mathbb{E}_{\text{init}}\left[\|Q_s(\,\cdot\,;\bar{\omega}) - Q_0(\,\cdot\,;\omega^*)\|_{L^2(\nu_\theta)}^2\right] \leq \mathcal{O}\left(\frac{R^3 d_{\zeta_s^k}^{3/2}}{\sqrt{M}} + \frac{R^{5/2} d_{\zeta_s^k}^{5/4}}{\sqrt[4]{M}} + \frac{R^2 d_{\zeta_s^k}}{\sqrt{T_{\text{critic}}}}\right),$$

where the expectation is taken with respect to the random initialization.

The proof of Theorem 4.6.2 is straightforward from [29, Theorem 4.6] and hence omitted.

4.6.1.3 Convergence to $Q_s^{\Pi^\theta}$

Next, we analyze the error between the global minimizer of (4.6.6) and the team-decentralized Q-function $Q_s^{\Pi^\theta}$ (defined in (4.3.10)) to complete the convergence analysis. Different from the single-agent case as in [29], we have to bound an additional error from using the localized information in the critic update, in addition to the neural network approximation-optimization error.

Proof of Theorem 4.5.4 First recall that by Lemma 4.3.5, $Q_s^{\Pi^\theta}$ satisfies the (c,ρ)-exponential decay property in Definition 4.3.4, with $c = \frac{r_{\max}}{1-\gamma}$, $\rho = \sqrt{\gamma}$. Now, let $\widehat{Q}_s^{\Pi^\theta}$ be any localized Q-function in (Local Q-function), then

$$\left|Q_s^{\Pi^\theta}(\zeta) - \widehat{Q}_s^{\Pi^\theta}(\zeta_s^k)\right| \leq c\rho^{k+1}, \quad \forall \zeta \in \Xi. \tag{4.6.7}$$

By the triangle inequality and $(a+b)^2 \leq 2(a^2+b^2)$,

$$\begin{aligned}\left\|Q_s(\,\cdot\,;\bar{\omega}) - Q_s^{\Pi^\theta}(\cdot)\right\|_{L^2(\nu_\theta)}^2 &\leq \left(\|Q_s(\,\cdot\,;\bar{\omega}) - Q_0(\,\cdot\,;\omega^*)\|_{L^2(\nu_\theta)} + \left\|Q_s^{\Pi^\theta}(\cdot) - Q_0(\,\cdot\,;\omega^*)\right\|_{L^2(\nu_\theta)}\right)^2 \\ &\leq 2\left(\|Q_s(\,\cdot\,;\bar{\omega}) - Q_0(\,\cdot\,;\omega^*)\|_{L^2(\nu_\theta)}^2 + \left\|Q_s^{\Pi^\theta}(\cdot) - Q_0(\,\cdot\,;\omega^*)\right\|_{L^2(\nu_\theta)}^2\right). \end{aligned} \tag{4.6.8}$$

The first term in (4.6.8) is studied in Theorem 4.6.2 and it suffices to bound the second term. By interpolating two intermediate terms $\widehat{Q}_s^{\Pi^\theta}$ and $\mathrm{Proj}_{\mathcal{F}_{R,M}}\widehat{Q}_s^{\Pi^\theta}$, we have

$$\left\|Q_s^{\Pi^\theta}(\cdot) - Q_0(\,\cdot\,;\omega^*)\right\|_{L^2(\nu_\theta)} \le \underbrace{\left\|Q_s^{\Pi^\theta}(\cdot) - \widehat{Q}_s^{\Pi^\theta}(\cdot)\right\|_{L^2(\nu_\theta)}}_{\text{(I)}} + \underbrace{\left\|\widehat{Q}_s^{\Pi^\theta}(\cdot) - \mathrm{Proj}_{\mathcal{F}_{R,M}}\widehat{Q}_s^{\Pi^\theta}(\cdot)\right\|_{L^2(\nu_\theta)}}_{\text{(II)}} + \underbrace{\left\|Q_0(\,\cdot\,;\omega^*) - \mathrm{Proj}_{\mathcal{F}_{R,M}}\widehat{Q}_s^{\Pi^\theta}(\cdot)\right\|_{L^2(\nu_\theta)}}_{\text{(III)}}. \tag{4.6.9}$$

First, we have (I) $\le c\rho^{k+1}$ according to (4.6.7). To bound (III), we have

$$\begin{aligned}
\text{(III)} &= \left\|\mathrm{Proj}_{\mathcal{F}_{R,M}}\mathcal{T}_s^\theta Q_0(\,\cdot\,;\omega^*) - \mathrm{Proj}_{\mathcal{F}_{R,M}}\widehat{Q}_s^{\Pi^\theta}(\cdot)\right\|_{L^2(\nu_\theta)} \\
&\le \left\|\mathrm{Proj}_{\mathcal{F}_{R,M}}\mathcal{T}_s^\theta Q_0(\,\cdot\,;\omega^*) - \mathrm{Proj}_{\mathcal{F}_{R,M}}\mathcal{T}_s^\theta Q_s^{\Pi^\theta}(\cdot)\right\|_{L^2(\nu_\theta)} \\
&\qquad + \left\|\mathrm{Proj}_{\mathcal{F}_{R,M}}\mathcal{T}_s^\theta Q_s^{\Pi^\theta}(\cdot) - \mathrm{Proj}_{\mathcal{F}_{R,M}}\widehat{Q}_s^{\Pi^\theta}(\cdot)\right\|_{L^2(\nu_\theta)} \\
&\le \gamma\left\|Q_0(\,\cdot\,;\omega^*) - Q_s^{\Pi^\theta}(\cdot)\right\|_{L^2(\nu_\theta)} + \left\|\mathcal{T}_s^\theta Q_s^{\Pi^\theta}(\cdot) - \widehat{Q}_s^{\Pi^\theta}(\cdot)\right\|_{L^2(\nu_\theta)} \\
&= \gamma\left\|Q_0(\,\cdot\,;\omega^*) - Q_s^{\Pi^\theta}(\cdot)\right\|_{L^2(\nu_\theta)} + \underbrace{\left\|Q_s^{\Pi^\theta}(\cdot) - \widehat{Q}_s^{\Pi^\theta}(\cdot)\right\|_{L^2(\nu_\theta)}}_{\text{(I)}} \\
&\le \gamma\left\|Q_0(\,\cdot\,;\omega^*) - Q_s^{\Pi^\theta}(\cdot)\right\|_{L^2(\nu_\theta)} + c\rho^{k+1}.
\end{aligned} \tag{4.6.10}$$

The first line in (4.6.10) is due to the fact that $Q_0(\cdot;\omega^*)$ is the unique fixed point of the operator $\mathrm{Proj}_{\mathcal{F}_{R,M}}\mathcal{T}_s^\theta$, (as proved in Lemma 4.6.1); the third line in (4.6.10) is because the operator $\mathrm{Proj}_{\mathcal{F}_{R,M}}\mathcal{T}_s^\theta$ is a γ-contraction in the $L^2(\nu_\theta)$ norm, and $\mathrm{Proj}_{\mathcal{F}_{R,M}}$ is non-expansive; the fourth line in (4.6.10) uses the fact that $Q_s^{\Pi^\theta}$ is the unique fixed point of $\mathcal{T}_s^\theta$; and the last line comes from the fact that (I) $\le c\rho^{k+1}$. Therefore, combining the self-bounding inequality (4.6.10) with (4.6.9) and the bound on (I) gives us

$$\left\|Q_s^{\Pi^\theta}(\cdot) - Q_0(\,\cdot\,;\omega^*)\right\|_{L^2(\nu_\theta)} \le \frac{1}{1-\gamma}\left(2c\rho^{k+1} + \underbrace{\left\|\widehat{Q}_s^{\Pi^\theta}(\cdot) - \mathrm{Proj}_{\mathcal{F}_{R,M}}\widehat{Q}_s^{\Pi^\theta}(\cdot)\right\|_{L^2(\nu_\theta)}}_{\text{(II)}}\right),$$

and consequently,

$$\left\|Q_s^{\Pi^\theta}(\cdot) - Q_0(\,\cdot\,;\omega^*)\right\|^2_{L^2(\nu_\theta)} \le \frac{1}{(1-\gamma)^2}\left(8c^2\rho^{2k+2} + 2\underbrace{\left\|\widehat{Q}_s^{\Pi^\theta}(\cdot) - \mathrm{Proj}_{\mathcal{F}_{R,M}}\widehat{Q}_s^{\Pi^\theta}(\cdot)\right\|^2_{L^2(\nu_\theta)}}_{\text{(II)}}\right). \tag{4.6.11}$$

Plugging (4.6.11) into (4.6.8) yields

$$\begin{aligned}
&\mathbb{E}_{\text{init}}\left[\left\|Q_s(\cdot;\bar{\omega})-Q_s^{\Pi^\theta}(\cdot)\right\|_{L^2(\nu_\theta)}^2\right]\\
&\leq 2\left(\mathbb{E}_{\text{init}}\left[\left\|Q_s(\cdot;\bar{\omega})-Q_0(\cdot;\omega^*)\right\|_{L^2(\nu_\theta)}^2\right]+\mathbb{E}_{\text{init}}\left[\left\|Q_s^{\Pi^\theta}(\cdot)-Q_0(\cdot;\omega^*)\right\|_{L^2(\nu_\theta)}^2\right]\right)\\
&\leq \mathcal{O}\left(\frac{R^3 d_{\zeta_s^k}^{3/2}}{\sqrt{M}}+\frac{R^{5/2}d_{\zeta_s^k}^{5/4}}{\sqrt[4]{M}}+\frac{R^2 d_{\zeta_s^k}}{\sqrt{T}}+c^2\rho^{2k+2}\right)\\
&\qquad+\frac{4}{(1-\gamma)^2}\mathbb{E}_{\text{init}}\left[\underbrace{\left\|\widehat{Q}_s^{\Pi^\theta}(\cdot)-\text{Proj}_{\mathcal{F}_{R,M}}\widehat{Q}_s^{\Pi^\theta}(\cdot)\right\|_{L^2(\nu_\theta)}^2}_{\text{(II)}}\right].
\end{aligned}\tag{4.6.12}$$

Term (II) measures the distance between $\widehat{Q}_s^{\Pi^\theta}$ and the class $\mathcal{F}_{R,M}$. As discussed in Section 4.6.1.1, the function class $\mathcal{F}_{R,M}$ converges to $\mathcal{F}_{R,\infty}^{s,k}$ (defined in Assumption 4.5.1) as $M\to\infty$. Consequently, term (II) decreases as the neural network gets wider. To quantitatively characterize the approximation error between $\mathcal{F}_{R,M}$ and $\mathcal{F}_{R,\infty}^{s,k}$, one needs the following lemma from [144] and [29, Proposition 4.3]:

Lemma 4.6.3 *Assume Assumption 4.5.1, we have*

$$\mathbb{E}_{\text{init}}\left[\underbrace{\left\|\widehat{Q}_s^{\Pi^\theta}(\cdot)-Proj_{\mathcal{F}_{R,M}}\widehat{Q}_s^{\Pi^\theta}(\cdot)\right\|_{L^2(\nu_\theta)}^2}_{(II)}\right]\leq\mathcal{O}\left(\frac{R^2 d_{\zeta_s^k}}{M}\right).\tag{4.6.13}$$

With this lemma, Theorem 4.5.4 follows immediately by plugging (4.6.13) into (4.6.12), and setting $c=\frac{r_{\max}}{1-\gamma}$, $\rho=\sqrt{\gamma}$, $T_{\text{critic}}=\Omega(M)$ in (4.6.12). □

4.6.2 Proof of Theorem 4.5.11: Convergence of Actor Update

The proof of Theorem 4.5.11 consists of two steps: the first step in Section 4.6.2.1 shows that the actor update converges to a stationary point of J (4.4.1), and the second step in Section 4.6.2.2 bridges the gap between the stationary point and the optimality.

For the rest of this section, we use η to denote η_{actor} and $\mathcal{B}_s$ to denote $\mathcal{B}_s^{\text{actor}}:=\left\{\theta_s\in\mathbb{R}^{M\times d_{\zeta_s}}:\|\theta_s-\theta_s(0)\|_\infty\leq R/\sqrt{M}\right\}$ for ease of notation. Meanwhile, define $\mathcal{B}=\prod_{s\in\mathcal{S}}\mathcal{B}_s$, the product space of $\mathcal{B}_s$'s, which is a convex set in $\mathbb{R}^{M\times d_\zeta}$.

4.6.2.1 Convergence to Stationary Point

Definition 4.6.4 *A point $\widetilde{\theta} \in \mathcal{B}$ is called a stationary point of $J(\cdot)$ if it holds that*

$$\nabla_\theta J(\widetilde{\theta})^\top (\theta - \widetilde{\theta}) \leq 0, \quad \forall \theta \in \mathcal{B}. \tag{4.6.14}$$

Define the following mapping G from $\mathbb{R}^{M \times d_\zeta}$ to itself:

$$G(\theta) := \eta^{-1} \cdot \left[\mathrm{Proj}_{\mathcal{B}} \left(\theta + \eta \cdot \nabla_\theta J(\theta) \right) - \theta \right]. \tag{4.6.15}$$

It is well-known that (4.6.14) holds if and only if $G(\widetilde{\theta}) = 0$ [161]. Now denote $\rho(t) := G(\theta(t))$, where $\theta(t) = \{\theta_s(t)\}_{s \in \mathcal{S}}$ is the actor parameter updated in Algorithm 5 in iteration t.

To show that Algorithm 5 converges to a stationary point, we focus on analyzing $\|\rho(t)\|_2$.

Theorem 4.6.5 *Assume Assumptions 4.5.5 - 4.5.7. Set $\eta = (T_{\text{actor}})^{-1/2}$ and assume $1 - L\eta \geq 1/2$, where L is the Lipschitz constant in Assumotion 4.5.7. Then the output of Algorithm 5 $\{\theta(t)\}_{t \in [T_{\text{actor}}]}$ satisfies*

$$\min_{t \in [T_{actor}]} \mathbb{E}\left[\|\rho(t)\|_2^2 \right] \leq \frac{8\tau^2 \Sigma^2 |\mathcal{S}|}{B} + \frac{4}{\sqrt{T_{actor}}} \mathbb{E}[J(\theta(T_{actor}+1)) - J(\theta(1))] + \epsilon_Q(T_{actor}). \tag{4.6.16}$$

Here ϵ_Q measures the error accumulated from the critic steps which is defined as

$$\begin{aligned} \epsilon_Q(T_{actor}) = & \frac{32\tau D R d_{\zeta_s}^{1/2} |\mathcal{S}|}{(1-\gamma)\eta T_{actor}} \cdot \sum_{t=1}^{T_{actor}} \sum_{s \in \mathcal{S}} \mathbb{E}\left[\left\| Q_s(\,\cdot\,; \bar{\omega}_s, t) - Q_s^{\Pi^{\theta(t)}}(\cdot) \right\|_{L^2(\nu_{\theta(t)})} \right] \\ & + \frac{16\tau^2 D^2 |\mathcal{S}|^2}{(1-\gamma)^2 T_{actor}} \cdot \sum_{t=1}^{T_{actor}} \sum_{s \in \mathcal{S}} \mathbb{E}\left[\left\| Q_s(\,\cdot\,; \bar{\omega}_s, t) - Q_s^{\Pi^{\theta(t)}}(\cdot) \right\|_{L^2(\nu_{\theta(t)})}^2 \right], \end{aligned} \tag{4.6.17}$$

where $\{Q_s(\,\cdot\,; \bar{\omega}_s, t)\}_{s \in \mathcal{S}}$ is the output of the critic update at step t in Algorithm 5. All expectations in (4.6.16) and (4.6.17) are taken over all randomness in Algorithm 4 and Algorithm 5.

Proof of Theorem 4.6.5 Let $t \in [T_{\text{actor}}]$, we first lower bound the difference between the expected total rewards of $\Pi^{\theta(t+1)}$ and $\Pi^{\theta(t)}$. By Assumption 4.5.7, $\nabla_\theta J(\theta)$ is L-Lipschitz continuous. Hence by Taylor's expansion,

$$J(\theta(t+1)) - J(\theta(t)) \geq \eta \cdot \nabla_\theta J(\theta(t))^\top \delta(t) - L/2 \cdot \|\theta(t+1) - \theta(t)\|_2^2, \tag{4.6.18}$$

where $\delta(t) = (\theta(t+1) - \theta(t))/\eta$. Meanwhile denote $\xi_s(t) = \widehat{g}_s(\theta(t)) - \mathbb{E}[\widehat{g}_s(\theta(t))]$, where $\widehat{g}_s(\theta(t))$ is defined in (4.4.15) and the expectation is taken over $\sigma_{\theta(t)}$ given $\{\bar{\omega}_s\}_{s \in \mathcal{S}}$. Then

$$\begin{aligned} \nabla_\theta J(\theta(t))^\top \delta(t) &= \sum_{s \in \mathcal{S}} \nabla_{\theta_s} J(\theta(t))^\top \delta_s(t) \\ &= \sum_{s \in \mathcal{S}} \left[\left(\nabla_{\theta_s} J(\theta(t)) - \mathbb{E}[\widehat{g}_s(\theta(t))] \right)^\top \delta_s(t) - \xi_s(t)^\top \delta_s(t) + \widehat{g}_s(\theta(t))^\top \delta_s(t) \right], \end{aligned} \tag{4.6.19}$$

where $\delta_s(t) := (\theta_s(t+1) - \theta_s(t))/\eta$. The first term in (4.6.19) represents the error of estimating $\nabla_{\theta_s} J(\theta(t))$ using

$$\mathbb{E}\left[\widehat{g}_s(\theta(t))\right] = \frac{1}{1-\gamma}\mathbb{E}_{\sigma_{\theta(t)}}\left[\left[\sum_{y\in\mathcal{N}_s^k} Q_y\left(\mu(\mathcal{N}_y^k), h(\mathcal{N}_y^k); \bar{\omega}_y, t\right)\right]\nabla_{\theta_s}\log\Pi^{\theta_s}(h(s)\mid\mu(s))\right].$$

To bound the first term, first notice that

$$\mathbb{E}\left[\widehat{g}_s(\theta(t))\right] = \frac{1}{1-\gamma}\mathbb{E}_{\sigma_{\theta(t)}}\left[\left[\sum_{y\in\mathcal{S}} Q_y\left(\mu(\mathcal{N}_y^k), h(\mathcal{N}_y^k); \bar{\omega}_y, t\right)\right]\nabla_{\theta_s}\log\Pi^{\theta_s}(h(s)\mid\mu(s))\right].$$

This is because for all $y \notin \mathcal{N}_s^k$, $Q_y\left(\mu(\mathcal{N}_y^k), h(\mathcal{N}_y^k); \bar{\omega}_y\right)$ is independent of s and consequently, we can verify that

$$\mathbb{E}_{\sigma_{\theta(t)}}\left[\left[\sum_{y\notin\mathcal{N}_s^k} Q_y\left(\mu(\mathcal{N}^k(y)), h(\mathcal{N}^k(y)); \bar{\omega}_y, t\right)\right]\nabla_{\theta_s}\log\Pi^{\theta_s}(h(s)\mid\mu(s))\right] = 0.$$

Therefore, following the similar computation in Lemma D.2, [29], we have

$$\left|\left(\nabla_{\theta_s} J(\theta(t)) - \mathbb{E}\left[\widehat{g}_s(\theta(t))\right]\right)^\top \delta_s(t)\right| \le \frac{4\tau D R d_{\zeta_s}^{1/2}}{(1-\gamma)\eta}\sum_{s\in\mathcal{S}}\left\|Q_s(\,\cdot\,;\bar{\omega}_s,t) - Q_s^{\theta(t)}(\cdot)\right\|_{L^2(\nu_{\theta(t)})}. \tag{4.6.20}$$

To bound the second term in (4.6.19), we simply have

$$\xi_s(t)^\top\delta_s(t) \le \|\xi_s(t)\|_2^2 + \|\delta_s(t)\|_2^2. \tag{4.6.21}$$

To handle the last term in (4.6.19), we have

$$\begin{aligned}
&\widehat{g}_s(\theta(t))^\top\delta_s(t) - \|\delta_s(t)\|_2^2 = \eta^{-1}\cdot\left(\eta\widehat{g}_s(\theta(t)) - (\theta_s(t+1)-\theta_s(t))\right)^\top\delta_s\\
=&\eta^{-1}\cdot\left(\theta_s(t+1/2) - \mathrm{Proj}_{\mathcal{B}_s}(\theta_s(t+1/2))\right)^\top\delta_s(t)\\
=&\eta^{-2}\cdot\left(\theta_s(t+1/2) - \mathrm{Proj}_{\mathcal{B}_s}(\theta_s(t+1/2))\right)^\top\left(\mathrm{Proj}_{\mathcal{B}_s}(\theta_s(t+1/2)) - \theta_s(t)\right) \ge 0
\end{aligned} \tag{4.6.22}$$

Here we write $\theta_s(t) + \eta\widehat{g}_s(\theta(t))$ as $\theta_s(t+1/2)$ to simplify the notation. The last inequality comes from the property of the projection onto a convex set.

Therefore, combining (4.6.19), (4.6.20), (4.6.21) and (4.6.22) suggests

$$\begin{aligned}
&\nabla_{\theta_s} J(\theta(t))^\top\delta_s(t) \ge\\
&\quad -\frac{4\tau D R d_{\zeta_s}^{1/2}}{(1-\gamma)\eta}\sum_{s\in\mathcal{S}}\left[\left\|Q_s(\,\cdot\,;\bar{\omega}_s,t) - Q_s^{\theta(t)}(\cdot)\right\|_{L^2(\nu_{\theta(t)})}\right] + \frac{1}{2}\left(\|\delta_s(t)\|_2^2 - \|\xi_s(t)\|_2^2\right).
\end{aligned} \tag{4.6.23}$$

Consequently,

$$\nabla_\theta J\left(\theta(t)\right)^\top \delta(t) \geq \\ -\frac{4\tau D R d_{\zeta_s}^{1/2}}{(1-\gamma)\eta}|\mathcal{S}|\sum_{s\in\mathcal{S}}\left[\left\|Q_s(\,\cdot\,;\bar{\omega}_s,t)-Q_s^{\Pi^{\theta(t)}}(\cdot)\right\|_{L^2(\nu_{\theta(t)})}\right]+\frac{1}{2}\left(\|\delta(t)\|_2^2-\|\xi(t)\|_2^2\right). \tag{4.6.24}$$

Thus, by plugging (4.6.24) into (4.6.18) and by Assumption 4.5.5, we have

$$\frac{1-L\cdot\eta}{2}\mathbb{E}\left[\|\delta(t)\|_2^2\right]\leq\eta^{-1}\cdot\mathbb{E}\left[J(\theta(t+1))-J(\theta(t))\right]+\frac{\tau^2\Sigma^2|\mathcal{S}|}{2B} \\ +\frac{4\tau D R d_{\zeta_s}^{1/2}|\mathcal{S}|}{(1-\gamma)\eta}\sum_{s\in\mathcal{S}}\left\|Q_s(\,\cdot\,;\bar{\omega}_s,t)-Q_s^{\Pi^{\theta(t)}}(\cdot)\right\|_{L^2(\nu_{\theta(t)})}. \tag{4.6.25}$$

Here the expectation is taken over $\sigma_{\theta(t)}$ given $\{\bar{\omega}_s\}_{s\in\mathcal{S}}$.

Now, in order to bridge the gap between $\|\delta(t)\|_2$ in (4.6.25) and $\|\rho(t)\|_2=\|G(\theta(t))\|_2$ in (4.6.15), we next will bound the difference $\|\delta(t)-\rho(t)\|_2$. We start with defining a local gradient mapping G_s from $\mathbb{R}^{M\times d_\zeta}$ to $\mathbb{R}^{M\times d_{\zeta_s}}$:

$$G_s(\theta):=\eta^{-1}\cdot\left[\mathrm{Proj}_{\mathcal{B}_s}\left(\theta_s+\eta\cdot\nabla_{\theta_s}J(\theta)\right)-\theta_s\right]. \tag{4.6.26}$$

Since $\mathcal{B}_s$ is an l_∞-ball around the initialization, it is easy to verify that $G_s(\theta)=(G(\theta))_s$. Therefore, we can further define $\rho_s(t)=G_s(\theta(t))$ and the following decomposition holds:

$$\|\delta(t)-\rho(t)\|_2^2=\sum_{s\in\mathcal{S}}\|\delta_s(t)-\rho_s(t)\|_2^2.$$

From the definitions of $\delta_s(t)$ and $\rho_s(t)$,

$$\begin{aligned}\|\delta_s(t)-\rho_s(t)\|_2&=\eta^{-1}\cdot\left\|\mathrm{Proj}_{\mathcal{B}_s}\left(\theta_s+\eta\cdot\nabla_{\theta_s}J(\theta)\right)-\theta_s-\mathrm{Proj}_{\mathcal{B}_s}\left(\theta_s+\eta\cdot\widehat{g}_s(\theta)\right)+\theta_s\right\|_2\\&=\eta^{-1}\cdot\left\|\mathrm{Proj}_{\mathcal{B}_s}\left(\theta_s+\eta\cdot\nabla_{\theta_s}J(\theta)\right)-\mathrm{Proj}_{\mathcal{B}_s}\left(\theta_s+\eta\cdot\widehat{g}_s(\theta)\right)\right\|_2\\&\leq\eta^{-1}\cdot\|\theta_s+\eta\cdot\nabla_{\theta_s}J(\theta)-\theta_s+\eta\cdot\widehat{g}_s(\theta)\|_2=\|\nabla_{\theta_s}J(\theta)-\widehat{g}_s(\theta)\|_2\end{aligned}$$

Following similar calculations in [29, Lemma D.3],

$$\mathbb{E}\left[\|\nabla_{\theta_s}J(\theta)-\widehat{g}_s(\theta)\|_2^2\right]\leq\frac{2\tau^2\Sigma^2}{B}+\frac{8\tau^2D^2}{(1-\gamma)^2}\left(\sum_{s\in\mathcal{S}}\left\|Q_s(\,\cdot\,;\bar{\omega}_s,t)-Q_s^{\Pi^{\theta(t)}}(\cdot)\right\|_{L^2(\nu_{\theta(t)})}\right)^2 \\ \leq\frac{2\tau^2\Sigma^2}{B}+\frac{8\tau^2D^2|\mathcal{S}|}{(1-\gamma)^2}\left(\sum_{s\in\mathcal{S}}\left\|Q_s(\,\cdot\,;\bar{\omega}_s,t)-Q_s^{\Pi^{\theta(t)}}(\cdot)\right\|_{L^2(\nu_{\theta(t)})}^2\right). \tag{4.6.27}$$

The expectation is taken over $\sigma_{\theta(t)}$ given $\{\bar{\omega}_s\}_{s\in\mathcal{S}}$. Consequently,

$$\mathbb{E}\left[\|\delta(t)-\rho(t)\|_2^2\right] \leq \frac{2\tau^2\Sigma^2|\mathcal{S}|}{B} + \frac{8\tau^2 D^2|\mathcal{S}|^2}{(1-\gamma)^2}\left(\sum_{s\in\mathcal{S}}\left\|Q_s(\cdot\,;\bar{\omega}_s,t) - Q_s^{\Pi^{\theta(t)}}(\cdot)\right\|^2_{L^2(\nu_{\theta(t)})}\right). \tag{4.6.28}$$

Set $\eta = 1/\sqrt{T_{\text{actor}}}$ and take (4.6.25) and (4.6.28), we obtain (4.6.16) from the following estimations:

$$\begin{aligned}
\min_{t\in[T_{\text{actor}}]}\mathbb{E}\left[\|\rho(t)\|_2^2\right] &\leq \frac{1}{T_{\text{actor}}}\cdot\sum_{t=1}^{T_{\text{actor}}}\|\rho(t)\|_2^2\\
&\leq \frac{2}{T_{\text{actor}}}\cdot\sum_{t=1}^{T_{\text{actor}}}\left(\mathbb{E}\left[\|\delta(t)-\rho(t)\|_2^2\right]+\mathbb{E}\left[\|\delta(t)\|_2^2\right]\right)\\
&\leq \frac{2}{T_{\text{actor}}}\cdot\sum_{t=1}^{T_{\text{actor}}}\left(\mathbb{E}\left[\|\delta(t)-\rho(t)\|_2^2\right]+2(1-L\cdot\eta)\mathbb{E}\left[\|\delta(t)\|_2^2\right]\right)\\
&\leq \frac{8\tau^2\Sigma^2|\mathcal{S}|}{B}+\frac{4}{\sqrt{T_{\text{actor}}}}\mathbb{E}[J(\theta(T_{\text{actor}}+1))-J(\theta(1))]+\epsilon_Q(T_{\text{actor}}),
\end{aligned}$$

where ϵ_Q measures the error accumulated from the critic steps which is defined in (4.6.17), i.e.,

$$\begin{aligned}
\epsilon_Q(T_{\text{actor}}) = &\frac{32\tau D R d_{\zeta_s}^{1/2}|\mathcal{S}|}{(1-\gamma)\eta T_{\text{actor}}}\cdot\sum_{t=1}^{T_{\text{actor}}}\sum_{s\in\mathcal{S}}\mathbb{E}\left[\left\|Q_s(\cdot\,;\bar{\omega}_s)-Q_s^{\Pi^{\theta(t)}}(\cdot)\right\|_{L^2(\nu_{\theta(t)})}\right]\\
&+\frac{16\tau^2D^2|\mathcal{S}|^2}{(1-\gamma)^2T_{\text{actor}}}\cdot\sum_{t=1}^{T_{\text{actor}}}\sum_{s\in\mathcal{S}}\mathbb{E}\left[\left\|Q_s(\cdot\,;\bar{\omega}_s)-Q_s^{\Pi^{\theta(t)}}(\cdot)\right\|^2_{L^2(\nu_{\theta(t)})}\right].
\end{aligned}$$

Here the expectations in (4.6.16) and (4.6.17) are taken over all randomness in Algorithm 4 and Algorithm 5. □

4.6.2.2 Bridging the gap between Stationarity and Optimality

Recall that σ_θ in (4.4.2) denotes the state-action visitation measure under policy Π^θ. Denote $\bar{\sigma}_\theta$ as the state visitation measure under policy Π^θ. Consequently,

$$\bar{\sigma}_\theta(\mu)\Pi^\theta(h\mid\mu) = \sigma_\theta(\mu,h).$$

Following similar steps in the proof of [29, Theorem 4.8], one can characterize the global optimality of the obtained stationary point $\widetilde{\theta}\in\mathcal{B}$ as the following.

Lemma 4.6.6 *Let $\widetilde{\theta} \in \mathcal{B}$ be a stationary point of $J(\cdot)$ satisfying condition (4.6.14) and let $\theta^* \in \mathcal{B}$ be the global maximum point of $J(\cdot)$ in $\mathcal{B}$. Then the following inequality holds:*

$$(1-\gamma)\left(J(\theta^*) - J(\widetilde{\theta})\right) \leq \frac{2r_{max}}{1-\gamma} \inf_{\theta \in \mathcal{B}} \left\| u_{\widetilde{\theta}}(\mu, h) - \sum_{s \in \mathcal{S}} \phi_{\widetilde{\theta}_s}(\mu(s), h(s))^\top \theta_s \right\|_{L^2(\sigma_{\widetilde{\theta}})}, \tag{4.6.29}$$

where $u_{\widetilde{\theta}}(\mu, h) := \frac{d\sigma_{\theta^*}}{d\sigma_{\widetilde{\theta}}}(\mu, h) - \frac{d\bar{\sigma}_{\theta^*}}{d\bar{\sigma}_{\widetilde{\theta}}}(\mu) + \sum_{s \in \mathcal{S}} \phi_{\widetilde{\theta}_s}(\mu(s), h(s))^\top \widetilde{\theta}_s$, and $\frac{\mathrm{d}\sigma_{\theta^*}}{\mathrm{d}\sigma_{\widetilde{\theta}}}, \frac{\mathrm{d}\bar{\sigma}_{\theta^*}}{\mathrm{d}\bar{\sigma}_{\widetilde{\theta}}}$ are the Radon-Nikodym derivatives between the corresponding measures.

Proof of Lemma 4.6.6 First recall that by (4.4.8), for any $\theta \in \mathcal{B}$,

$$\nabla_\theta J(\widetilde{\theta})^\top (\theta - \widetilde{\theta}) = \sum_{s \in \mathcal{S}} \nabla_{\theta_s} J(\widetilde{\theta})^\top (\theta_s - \widetilde{\theta}_s) = \frac{\tau}{1-\gamma} \sum_{s \in \mathcal{S}} \mathbb{E}_{\sigma_{\widetilde{\theta}}} \left[Q^{\Pi^{\widetilde{\theta}}}(\mu, h) \cdot \Phi(\widetilde{\theta}, s, \mu, h)^\top (\theta_s - \widetilde{\theta}_s) \right],$$

in which $\Phi(\theta, s, \mu, h) := \phi_{\theta_s}(\mu(s), h(s)) - \mathbb{E}_{h(s)' \sim \Pi_s^{\theta_s}(\cdot|\mu(s))} \left[\phi_{\theta_s}(\mu(s), h'(s)) \right]$ is defined in (4.4.7).

Since $\widetilde{\theta} \in \mathcal{B}$ is a stationary point of $J(\cdot)$,

$$\sum_{s \in \mathcal{S}} \mathbb{E}_{\sigma_{\widetilde{\theta}}} \left[Q^{\Pi^{\widetilde{\theta}}}(\mu, h) \cdot \Phi(\widetilde{\theta}, s, \mu, h)^\top (\theta_s - \widetilde{\theta}_s) \right] \leq 0, \quad \forall \theta \in \mathcal{B}. \tag{4.6.30}$$

Denote $A^{\Pi^{\widetilde{\theta}}}(\mu, h) := Q^{\Pi^{\widetilde{\theta}}}(\mu, h) - V^{\Pi^{\widetilde{\theta}}}(\mu)$ as the advantage function under policy $\Pi^{\widetilde{\theta}}$. It holds from the definition that $\mathbb{E}_{h \sim \Pi^{\widetilde{\theta}}(\cdot|\mu)}[A^{\Pi^{\widetilde{\theta}}}(\mu, h)] = V^{\Pi^{\widetilde{\theta}}}(\mu) - V^{\Pi^{\widetilde{\theta}}}(\mu) = 0$. Meanwhile, $\sup_{(\mu,h) \in \Xi} \left| A^{\Pi^{\widetilde{\theta}}}(\mu, h) \right| \leq 2 \sup_{\mu \in \mathcal{P}^N(\mathcal{S})} \left| V^{\Pi^{\widetilde{\theta}}}(\mu) \right| \leq \frac{2r_{\max}}{1-\gamma}$.

Given that $\mathbb{E}_{h \sim \Pi^{\widetilde{\theta}}(\cdot|\mu)}[A^{\Pi^{\widetilde{\theta}}}(\mu, h)] = 0$ and $\mathbb{E}_{h \sim \Pi^{\widetilde{\theta}}(\cdot|\mu)}[\Phi(\widetilde{\theta}, s, \mu, h)] = 0$, we have for any $s \in \mathcal{S}$,

$$\mathbb{E}_{\sigma_{\widetilde{\theta}}} \left[V^{\Pi^{\widetilde{\theta}}}(\mu) \cdot \Phi(\widetilde{\theta}, s, \mu, h) \right] = 0, \qquad \text{and} \tag{4.6.31}$$

$$\mathbb{E}_{\sigma_{\widetilde{\theta}}} \left[A^{\Pi^{\widetilde{\theta}}}(\mu, h) \cdot \mathbb{E}_{h(s)' \sim \Pi_s^{\widetilde{\theta}_s}(\cdot|\mu(s))} \left[\phi_{\widetilde{\theta}_s}(\mu(s), h'(s)) \right] \right] = 0. \tag{4.6.32}$$

Combining (4.6.30) with (4.6.31) and (4.6.32),

$$\sum_{s \in \mathcal{S}} \mathbb{E}_{\sigma_{\widetilde{\theta}}} \left[A^{\Pi^{\widetilde{\theta}}}(\mu, h) \cdot \phi_{\widetilde{\theta}_s}(\mu(s), h(s))^\top (\theta_s - \widetilde{\theta}_s) \right] \leq 0, \quad \forall \theta \in \mathcal{B}. \tag{4.6.33}$$

Moreover, by the Performance Difference Lemma [89],

$$(1-\gamma) \cdot \left(J(\theta^*) - J(\widehat{\theta})\right) = \mathbb{E}_{\bar{\sigma}_{\theta^*}} \left[\left\langle A^{\Pi^{\widetilde{\theta}}}(\mu, \cdot), \Pi^{\theta^*}(\cdot \mid \mu) - \Pi^{\widetilde{\theta}}(\cdot \mid \mu) \right\rangle \right]. \tag{4.6.34}$$

Combining (4.6.34) with (4.6.33), it holds that for any $\theta \in \mathcal{B}$,

$$\begin{aligned}
&(1-\gamma)\cdot\left(J(\theta^*)-J(\widehat{\theta})\right)\\
\leq&\mathbb{E}_{\bar{\sigma}_{\theta^*}}\left[\left\langle A^{\Pi^{\widetilde{\theta}}}(\mu,\cdot),\Pi^{\theta^*}(\cdot\mid\mu)-\Pi^{\widetilde{\theta}}(\cdot\mid\mu)\right\rangle\right]-\sum_{s\in\mathcal{S}}\mathbb{E}_{\sigma_{\widetilde{\theta}}}\left[A^{\Pi^{\widetilde{\theta}}}(\zeta)\cdot\phi_{\widetilde{\theta}_s}(\zeta_s)^\top(\theta_s-\widetilde{\theta}_s)\right]\\
=&\mathbb{E}_{\sigma_{\widetilde{\theta}}}\left[A^{\Pi^{\widetilde{\theta}}}(\mu,h)\cdot\left(\frac{\mathrm{d}\sigma_{\theta^*}}{\mathrm{d}\sigma_{\widetilde{\theta}}}(\mu,h)-\frac{\mathrm{d}\bar{\sigma}_{\theta^*}}{\mathrm{d}\bar{\sigma}_{\widetilde{\theta}}}(\mu)-\sum_{s\in\mathcal{S}}\phi_{\widetilde{\theta}_s}(\mu(s),h(s))^\top(\theta_s-\widetilde{\theta}_s)\right)\right]. \qquad (4.6.35)
\end{aligned}$$

Therefore,

$$\begin{aligned}
&(1-\gamma)\cdot\left(J(\theta^*)-J(\widehat{\theta})\right)\\
\leq&\frac{2r_{\max}}{1-\gamma}\inf_{\theta\in\mathcal{B}}\left\|\frac{\mathrm{d}\sigma_{\theta^*}}{\mathrm{d}\sigma_{\widetilde{\theta}}}(\mu,h)-\frac{\mathrm{d}\bar{\sigma}_{\theta^*}}{\mathrm{d}\bar{\sigma}_{\widetilde{\theta}}}(\mu)-\sum_{s\in\mathcal{S}}\phi_{\widetilde{\theta}_s}(\mu(s),h(s))^\top(\theta_s-\widetilde{\theta}_s)\right\|_{L^2(\sigma_{\widetilde{\theta}})}\\
=&\frac{2r_{\max}}{1-\gamma}\inf_{\theta\in\mathcal{B}}\left\|u_{\widetilde{\theta}}(\mu,h)-\sum_{s\in\mathcal{S}}\phi_{\widetilde{\theta}_s}(\mu(s),h(s))^\top\theta_s\right\|_{L^2(\sigma_{\widetilde{\theta}})}, \qquad (4.6.36)
\end{aligned}$$

where $u_{\widetilde{\theta}}(\mu,h):=\frac{d\sigma_{\theta^*}}{d\sigma_{\widetilde{\theta}}}(\mu,h)-\frac{d\bar{\sigma}_{\theta^*}}{d\bar{\sigma}_{\widetilde{\theta}}}(\mu)+\sum_{s\in\mathcal{S}}\phi_{\widetilde{\theta}_s}(\mu(s),h(s))^\top\widetilde{\theta}_s$, and $\frac{\mathrm{d}\sigma_{\theta^*}}{\mathrm{d}\sigma_{\widetilde{\theta}}},\frac{\mathrm{d}\bar{\sigma}_{\theta^*}}{\mathrm{d}\bar{\sigma}_{\widetilde{\theta}}}$ are the Radon-Nikodym derivatives between corresponding measures. □

To further bound the right-hand-side of (4.6.29) in Lemma 4.6.6, define the following function class

$$\begin{aligned}
\widetilde{\mathcal{F}}_{R,M}=\Bigg\{f_0(\zeta;\theta):=\sum_{s\in\mathcal{S}}\underbrace{\left[\frac{1}{\sqrt{M}}\sum_{m=1}^{M}\mathbb{1}\left\{[\theta_s(0)]_m^\top\zeta_s>0\right\}[\theta_s]_m^\top\zeta_s\right]}_{(\star)}:\\
\theta_s\in\mathbb{R}^{M\times d_{\zeta_s}},\|\theta_s-\theta_s(0)\|_\infty\leq R/\sqrt{M}\Bigg\}, \qquad (4.6.37)
\end{aligned}$$

given an initialization $\theta_s(0)\in\mathbb{R}^{M\times d_{\zeta_s}}$, $s\in\mathcal{S}$ and $b\in\mathbb{R}^M$.

$\widetilde{\mathcal{F}}_{R,M}$ (4.6.37) is a local linearization of the actor neural network. More specifically, term $(\star)$ in (4.6.37) locally linearizes the decentralized actor neural network $f(\zeta_s;\theta_s)$ (4.4.4) with respect to θ_s. Any $f_0(\zeta;\theta)\in\widetilde{\mathcal{F}}_{R,M}$ is a sum of $|\mathcal{S}|$ inner products between feature mapping $\phi_{\theta_s(0)}(\cdot)$ (4.4.6) and parameter θ_s: $f_0(\zeta;\theta)=\sum_{s\in\mathcal{S}}\phi_{\theta_s(0)}(\zeta_s)\cdot\theta_s$. As the width of the neural network $M\to\infty$, $\widetilde{\mathcal{F}}_{R,M}$ converges to $\mathcal{F}_{R,\infty}$ (defined in Assumption 4.5.9). The approximation error between $\widetilde{\mathcal{F}}_{R,M}$ and $\mathcal{F}_{R,\infty}$ is bounded in the following lemma.

Lemma 4.6.7 *For any function $f(\zeta)\in\mathcal{F}_{R,\infty}$ defined in Assumption 4.5.9, we have*

$$\mathbb{E}_{\text{init}}\left[\left\|f(\cdot)-Proj_{\widetilde{\mathcal{F}}_{R,M}}f(\cdot)\right\|_{L^2(\sigma_{\widetilde{\theta}})}\right]\leq\mathcal{O}\left(\frac{|\mathcal{S}|Rd_{\zeta_s}^{1/2}}{M^{1/2}}\right). \qquad (4.6.38)$$

Lemma 4.6.7 follows from [144] and [29, Proposition 4.3]. The factor $|\mathcal{S}|$ stems from the fact that $\mathcal{F}_{R,\infty}$ can be decomposed into $|\mathcal{S}|$ independent reproducing kernel Hilbert spaces. With Lemma 4.6.7, we are ready to establish an upper bound for the right-hand-side of (4.6.29) in the following proposition.

Proposition 4.6.8 *Under Assumption 4.5.9, let $\widetilde{\theta} \in \mathcal{B}$ be a stationary point of $J(\cdot)$ and let $\theta^* \in \mathcal{B}$ be the global maximum point of $J(\cdot)$ in $\mathcal{B}$. Then the following inequality holds:*

$$(1-\gamma)\left(J(\theta^*) - J(\widetilde{\theta})\right) \leq \mathcal{O}\left(\frac{|\mathcal{S}| R^{3/2} d_{\zeta_s}^{3/4}}{M^{1/4}}\right). \tag{4.6.39}$$

Proof of Proposition 4.6.8 First by the triangle inequality,

$$\begin{aligned}\inf_{\theta\in\mathcal{B}} \left\| u_{\widetilde{\theta}}(\zeta) - \sum_{s\in\mathcal{S}} \phi_{\widetilde{\theta}_s}(\zeta_s)^\top \theta_s \right\|_{L^2(\sigma_{\widetilde{\theta}})} &\leq \left\| u_{\widetilde{\theta}}(\zeta) - \mathrm{Proj}_{\widetilde{\mathcal{F}}_{R,M}} u_{\widetilde{\theta}}(\zeta) \right\|_{L^2(\sigma_{\widetilde{\theta}})} \\ &\quad + \inf_{\theta\in\mathcal{B}} \left\| \mathrm{Proj}_{\widetilde{\mathcal{F}}_{R,M}} u_{\widetilde{\theta}}(\zeta) - \sum_{s\in\mathcal{S}} \phi_{\widetilde{\theta}_s}(\zeta_s)^\top \theta_s \right\|_{L^2(\sigma_{\widetilde{\theta}})},\end{aligned} \tag{4.6.40}$$

where $\widetilde{\mathcal{F}}_{R,M}$ is defined in (4.6.37). We denote $\mathrm{Proj}_{\widetilde{\mathcal{F}}_{R,M}} u_{\widetilde{\theta}}(\zeta) = \sum_{s\in\mathcal{S}} \phi_{\theta_s(0)}(\zeta_s)\cdot \widehat{\theta}_s \in \widetilde{\mathcal{F}}_{R,M}$ for some $\widehat{\theta} \in \mathcal{B}$. Therefore, by Lemma 4.6.7, the first term on the right-hand-side of (4.6.40) is bounded by (4.6.38):

$$\left\| u_{\widetilde{\theta}}(\zeta) - \sum_{s\in\mathcal{S}} \phi_{\theta_s(0)}(\zeta_s)\cdot \widehat{\theta}_s \right\|_{L^2(\sigma_{\widetilde{\theta}})} \leq \mathcal{O}\left(\frac{|\mathcal{S}| R d_{\zeta_s}^{1/2}}{M^{1/2}}\right).$$

The following Lemma 4.6.9 is a direct application of [181, Lemma E.2], which is used to bound the second term on the right-hand-side of (4.6.40).

Lemma 4.6.9 *It holds for any $\theta_s, \theta_s' \in \mathcal{B}_s = \left\{\alpha_s \in \mathbb{R}^{M\times d_{\zeta_s}} : \|\alpha_s - \theta_s(0)\|_\infty \leq R/\sqrt{M}\right\}$ that*

$$\mathbb{E}_{\text{init}}\left[\|\phi_{\theta_s}(\zeta_s)^\top \theta_s' - \phi_{\theta_s(0)}(\zeta_s)^\top \theta_s'\|_{L^2(\sigma_\theta)}\right] \leq \mathcal{O}\left(\frac{R^{3/2} d_{\zeta_s}^{3/4}}{M^{1/4}}\right), \tag{4.6.41}$$

where the expectation is taken over random initialization.

Taking $\theta = \widetilde{\theta}$ and $\theta' = \widehat{\theta}$ in Lemma 4.6.9 gives us

$$\sum_{s\in\mathcal{S}} \left\| \phi_{\theta_s(0)}(\zeta_s)\cdot\widehat{\theta}_s - \phi_{\widetilde{\theta}_s}(\zeta_s)^\top \widehat{\theta}_s \right\|_{L^2(\sigma_{\widetilde{\theta}})} \leq \mathcal{O}\left(\frac{|\mathcal{S}| R^{3/2} d_{\zeta_s}^{3/4}}{M^{1/4}}\right)$$

Therefore, by Lemma 4.6.1,

$$(1-\gamma)\left(J(\theta^*)-J(\widetilde{\theta})\right) \leq \inf_{\theta\in\mathcal{B}} \left\| u_{\widetilde{\theta}}(\zeta) - \sum_{s\in\mathcal{S}} \phi_{\widetilde{\theta}_s}(\zeta_s)^\top \theta_s \right\|_{L^2(\sigma_{\widetilde{\theta}})} \leq \mathcal{O}\left(\frac{|\mathcal{S}|R^{3/2}d_{\zeta_s}^{3/4}}{M^{1/4}}\right).$$

□

Now we are ready to establish Theorem 4.5.11.

Proof of Theorem 4.5.11 Following similar calculations as in [181, Section H.3], we obtain that at iteration $t \in [T_{\text{actor}}]$,

$$\nabla_\theta J(\theta(t))^\top(\theta-\theta(t)) \leq 2(R+\frac{\eta\cdot r_{\max}}{1-\gamma})\cdot\|\rho(t)\|_2, \quad \forall\theta\in\mathcal{B}. \tag{4.6.42}$$

The right-hand-side of (4.6.42) quantifies the deviation of $\theta(t)$ from a stationary point $\widetilde{\theta}$. Having (4.6.42) and following similar arguments for Lemma 4.6.6 and Proposition 4.6.8, we can show that

$$(1-\gamma)\min_{t\in[T_{\text{actor}}]}\mathbb{E}\left[J(\theta^*)-J(\theta(t))\right] \leq \mathcal{O}\left(\frac{|\mathcal{S}|R^{3/2}d_{\zeta_s}^{3/4}}{M^{1/4}}\right)+2\left(R+\frac{\eta\cdot r_{\max}}{1-\gamma}\right)\cdot\min_{t\in[T_{\text{actor}}]}\mathbb{E}[\|\rho(t)\|_2]. \tag{4.6.43}$$

Here the last term $\min_{t\in[T_{\text{actor}}]}\mathbb{E}[\|\rho(t)\|_2]$ is bounded by (4.6.16) in Theorem 4.6.5, while the term $\epsilon_Q(T_{\text{actor}})$ in (4.6.17) can be upper bounded by Theorem 4.5.4. Finally with the parameters stated in Theorem 4.5.11, the following statement holds by straightforward calculation:

$$\min_{t\in[T_{\text{actor}}]}\mathbb{E}\left[J(\theta^*)-J(\theta(t))\right] \leq \mathcal{O}\left(|\mathcal{S}|^{1/2}B^{-1/2}+|\mathcal{S}||\mathcal{A}|^{1/4}\left(\gamma^{k/8}+(T_{\text{actor}})^{-1/4}\right)\right).$$

□

4.7 A Network Example Satisfying Technical Assumptions

In this section, we provide a concrete network example that satisfies all Assumptions 4.5.1, 4.5.2, 4.5.5, 4.5.6, 4.5.7 and 4.5.9 (or their mild relaxations). The structure of this network is shown in Figure 4.3 which consists of five states. Within each time step, an agent can travel from state i to j only if there is a directed link from state i to j. We consider a mean-field MARL problem with ten agents on this five-state network. For an agent at a given state i, the admissible action is to travel to a neighboring state at the next time step. Once the agent selects a neighboring state as its action, it will transit to that state with probability

one in the next time step. The discount parameter of the problem is set to be $\gamma = 0.95$. The team decentralized policy is paramaterized in the form of (4.4.4).

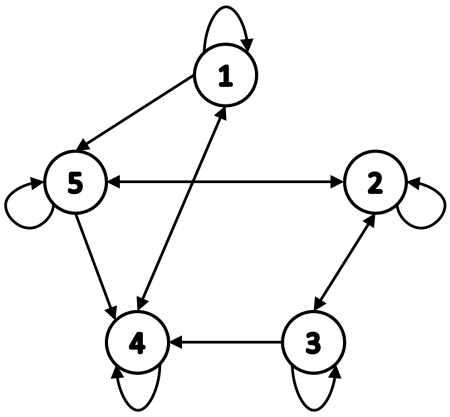


Figure 4.3: The 5-state network structure used to verify Assumptions 4.5.1, 4.5.2, 4.5.5, 4.5.6, 4.5.7 and 4.5.9.

Assumption 4.5.1. In general, it may be difficult to verify whether $\widehat{Q}_s^{\Pi^\theta}$ in (Local Q-function) belongs to $\mathcal{F}_{R,\infty}^{s,k}$ in (4.5.1) by direct computation. However, it can be argued that any continuous function (including any $\widehat{Q}_s^{\Pi^\theta}$ in (Local Q-function)) satisfies Assumption 4.5.1 with some controllable approximation error. More specifically, as pointed out in Remark 4.5.3, $\mathcal{F}_{R,\infty}^{s,k}$ in (4.5.1) is a subset of a reproducing kernel Hilbert space (RKHS) which is dense in the space of continuous functions. In this case, any continuous $\widehat{Q}_s^{\Pi^\theta}$ can be approximated by some function in $\mathcal{F}_{R,\infty}^{s,k}$ up to some approximation error, and the subsequent convergence analysis can also be modified to reflect such error. In short, Assumption 4.5.1 is satisfied by the example in Figure 4.3 up to some approximation error.

Assumption 4.5.2. As mentioned in Remark 4.5.3, Assumption 4.5.2 is satisfied when the stationary distribution ν_θ and the visitation measure σ_θ are both uniformly upper bounded over all policies. It is indeed difficult to verify such assumption by direct computation. Alternatively, we conduct a numerical experiment to show that the upper-boundedness of ν_θ and σ_θ is a reasonable assumption for the example in Figure 4.3.

Given a neural policy Π^θ, the stationary distribution ν_θ and the visitation measure σ_θ are computed by numerical simulations of the system's trajectories. We generate 800 random neural policies $\{\Pi^{\theta_i}\}_{i=1}^{800}$, and for each θ_i, the maximum value of ν_{θ_i} and σ_{θ_i} is recorded. The results are shown in Figure 4.4. It is observed from the histogram that most of the randomly chosen θ's lead to a maximum value smaller than 0.02, while the overall upper bound is smaller than 0.03. Therefore, Assumption 4.5.2 holds numerically under this example.

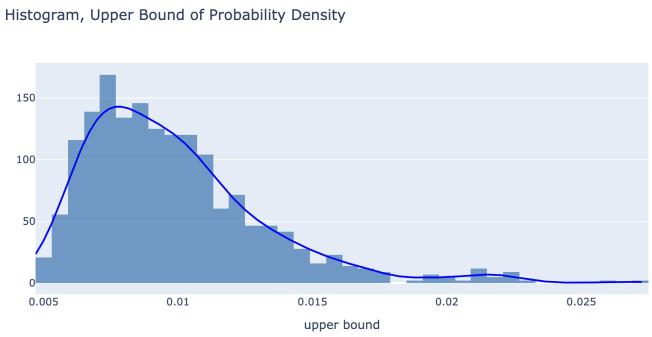


Figure 4.4: Upper bound of the stationary distribution σ_θ and the visitation measure ν_θ over 800 random policies on the 5-state network example.

Assumption 4.5.5. Assumption 4.5.5 also holds under mild conditions. More specifically, when the estimator $\widehat{g}_s$ in (4.4.15) can be viewed as an average of B i.i.d. samples

$$\left[\sum_{y\in\mathcal{N}_s^k} Q_y\left(\mu_l(\mathcal{N}_y^k), h_l(\mathcal{N}_y^k); \bar{\omega}_y\right)\right] \cdot \widehat{\Phi}(\theta(t), s, \mu_l, h_l), \quad l \in [B],$$

Assumption 4.5.5 holds naturally if each sample has uniformly bounded variance over all parameters ω and θ. A sufficient condition to guarantee the uniformly bounded variance is when the neural Q-function $Q_y(\cdot;\bar{\omega}_y)$ is uniformly bounded over all parameters. Indeed, when $Q_y(\cdot;\bar{\omega}_y)$ is a two-layer neural network with bounded parameters $\bar{\omega}_y$ and bounded input, a uniform bound on $Q_y(\cdot;\bar{\omega}_y)$ is guaranteed. Hence, Assumption 4.5.5 holds when the parameters of the critic networks are uniformly bounded.

Assumption 4.5.6. Similar as Assumption 4.5.2, due to the difficulty in directly computing ν_θ and σ_θ, Assumption 4.5.6 is verified numerically under the example in Figure 4.3. Again, 800 random neural policies $\{\Pi^{\theta_i}\}_{i=1}^{800}$ are generated, and $\mathbb{E}_{\nu_\theta}\left[(\mathrm{d}\sigma_\theta/\mathrm{d}\nu_\theta(\mu,h))^2\right]$, the L_2 norm of Radon-Nikodym derivative between σ_θ and ν_θ, is computed for each θ. The results are shown in Figure 4.4. It is observed from the histogram that most of the randomly chosen θ's lead to a bounded L_2 norm smaller than 30, while the overall upper bound is smaller than 45. Therefore, Assumption 4.5.6 holds numerically under this example.

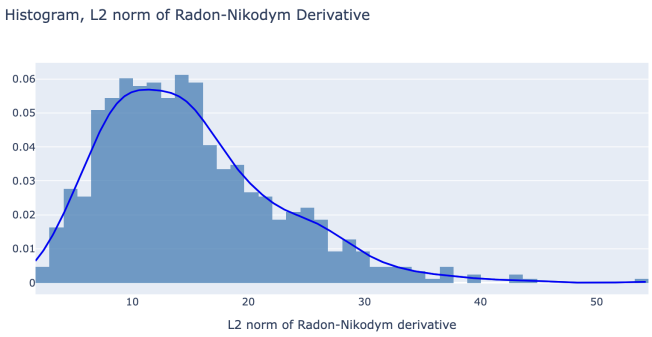


Figure 4.5: L_2 norm of Radon-Nikodym derivative $\mathbb{E}_{\nu_\theta}\left[(\mathrm{d}\sigma_\theta/\mathrm{d}\nu_\theta(\mu,h))^2\right]$ between the stationary distribution σ_θ and the visitation measure ν_θ over 800 random policies on the 5-state network example.

Assumption 4.5.7. In general, Assumption 4.5.7 holds when the transition probability and the reward function are both Lipschitz continuous with respect to their inputs [139], or when the reward is uniformly bounded and the score function $\nabla_\theta \log \Pi^\theta$ is uniformly bounded and Lipschitz continuous with respect to θ [202]. Under the particular example in Figure 4.3, one can set the reward function to be constant, so that the Lipschitz condition in Assumption 4.5.7 holds immediately.

Assumption 4.5.9. Assumption 4.5.9 is similar to Assumption 4.5.1, and such assumption is satisfied by any continuous function up to an approximation error.

Overall, we have shown that Assumptions 4.5.1, 4.5.2, 4.5.5, 4.5.6, 4.5.7 and 4.5.9 in this chapter (or their mild relaxations) are satisfied by the particular example in Figure 4.3.

Bibliography

[1] Alekh Agarwal, Sham M Kakade, Jason D Lee, and Gaurav Mahajan. Optimality and approximation with policy gradient methods in markov decision processes. In Conference on Learning Theory, pages 64–66. PMLR, 2020.

[2] Alekh Agarwal, Sham M Kakade, Jason D Lee, and Gaurav Mahajan. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. Journal of Machine Learning Research, 22(98):1–76, 2021.

[3] Rajeev Agrawal. Sample mean based index policies by O(log(n)) regret for the multi-armed bandit problem. Advances in Applied Probability, 27(4):1054–1078, 1995.

[4] R. Aïd, M. Basei, and H. Pham. A McKean–Vlasov approach to distributed electricity generation development. Mathematical Methods of Operations Research, 91(2):269–310, 2020.

[5] René Aïd, Roxana Dumitrescu, and Peter Tankov. The entry and exit game in the electricity markets: a mean-field game approach. Journal of Dynamics & Games, 8(4):331, 2021.

[6] Zeyuan Allen-Zhu, Yuanzhi Li, and Yingyu Liang. Learning and generalization in overparameterized neural networks, going beyond two layers. In Advances in Neural Information Processing Systems, volume 32, pages 6158–6169, 2019.

[7] Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. In International Conference on Machine Learning, pages 242–252. PMLR, 2019.

[8] Daniel Andersson and Boualem Djehiche. A maximum principle for SDEs of mean-field type. Applied Mathematics & Optimization, 63(3):341–356, 2011.

[9] Kavosh Asadi, Dipendra Misra, and Michael L Littman. Lipschitz continuity in model-based reinforcement learning. arXiv preprint arXiv:1804.07193, 2018.

[10] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multi-armed bandit problem. Machine learning, 47:235–256, 2002.

[11] Yu Bai, Chi Jin, and Tiancheng Yu. Near-optimal reinforcement learning with self-play. In Advances in Neural Information Processing Systems, volume 33, pages 2159–2170, 2020.

[12] André MS Barreto, Doina Precup, and Joelle Pineau. Practical kernel-based reinforcement learning. Journal of Machine Learning Research, 17(1):2372–2441, 2016.

[13] Tamer Başar and Pierre Bernhard. H-infinity Optimal Control and Related Minimax Design Problems: a Dynamic Game Approach. Springer Science & Business Media, 2008.

[14] Tamer Başar and Geert Jan Olsder. Dynamic Noncooperative Game Theory. SIAM, 1998.

[15] Jonathan Baxter and Peter L Bartlett. Infinite-horizon policy-gradient estimation. Journal of Artificial Intelligence Research, 15:319–350, 2001.

[16] Richard Bellman. A Markovian decision process. Journal of Mathematics and Mechanics, pages 679–684, 1957.

[17] Alain Bensoussan, Boualem Djehiche, Hamidou Tembine, and Phillip Yam. Risk-sensitive mean-field-type control. In Conference on Decision and Control, pages 33–38. IEEE, 2017.

[18] Alain Bensoussan, Jens Frehse, and Phillip Yam. Mean Field Games and Mean Field Type Control Theory, volume 101. Springer, 2013.

[19] Dimitri P. Bertsekas. Dynamic Programming and Optimal Control, Vol. I. Athena scientific, 4th edition, 2012.

[20] Dimitri P. Bertsekas. Dynamic Programming and Optimal Control, Vol. II. Athena Scientific, 4th edition, 2012.

[21] Dimitri P Bertsekas and Steven E Shreve. Stochastic optimal control, volume 139 of mathematics in science and engineering, 1978.

[22] Dimitri P Bertsekas and John N Tsitsiklis. Neuro-Dynamic Programming, volume 5. Athena Scientific Belmont, MA, 1996.

[23] Jalaj Bhandari, Daniel Russo, and Raghav Singal. A finite time analysis of temporal difference learning with linear function approximation. In Conference on Learning Theory, pages 1691–1692. PMLR, 2018.

[24] Shalabh Bhatnagar, Richard S Sutton, Mohammad Ghavamzadeh, and Mark Lee. Natural actor–critic algorithms. Automatica, 45(11):2471–2482, 2009.

[25] Noam Brown and Tuomas Sandholm. Libratus: The superhuman AI for no-limit poker. In Proceedings of the 26th International Joint Conference on Artificial Intelligence, pages 5226–5228, 2017.

[26] Noam Brown and Tuomas Sandholm. Superhuman ai for multiplayer poker. Science, 365(6456):885–890, 2019.

[27] Rainer Buckdahn, Boualem Djehiche, and Juan Li. A general stochastic maximum principle for SDEs of mean-field type. Applied Mathematics & Optimization, 64(2):197–216, 2011.

[28] Theophile Cabannes, Mathieu Lauriere, Julien Perolat, Raphael Marinier, Sertan Girgin, Sarah Perrin, Olivier Pietquin, Alexandre M Bayen, Eric Goubault, and Romuald Elie. Solving n-player dynamic routing games with congestion: a mean-field approach. arXiv preprint arXiv:2110.11943, 2021.

[29] Qi Cai, Zhuoran Yang, Jason D Lee, and Zhaoran Wang. Neural temporal-difference learning converges to global optima. In Advances in Neural Information Processing Systems, volume 32, pages 11315–11326, 2019.

[30] Dan Calderone and S Shankar Sastry. Markov decision process routing games. In International Conference on Cyber-Physical Systems, pages 273–280. IEEE, 2017.

[31] Yongcan Cao, Wenwu Yu, Wei Ren, and Guanrong Chen. An overview of recent progress in the study of distributed multi-agent coordination. IEEE Transactions on Industrial informatics, 9(1):427–438, 2012.

[32] René Carmona and François Delarue. Forward–backward stochastic differential equations and controlled McKean–Vlasov dynamics. The Annals of Probability, 43(5):2647–2700, 2015.

[33] René Carmona and François Delarue. Probabilistic Theory of Mean Field Games with Applications I-II. Springer, 2018.

[34] René Carmona, Jean-Pierre Fouque, and Li-Hsien Sun. Mean-field games and systemic risk. Communications in Mathematical Sciences, 13(4):911–933, 2015.

[35] René Carmona, Mathieu Laurière, and Zongjun Tan. Linear-quadratic mean-field reinforcement learning: convergence of policy gradient methods. arXiv preprint arXiv:1910.04295, 2019.

[36] René Carmona, Mathieu Laurière, and Zongjun Tan. Model-free mean-field reinforcement learning: mean-field MDP and mean-field Q-learning. arXiv preprint arXiv:1910.12802, 2019.

[37] Philippe Casgrain and Sebastian Jaimungal. Mean-field games with differing beliefs for algorithmic trading. Mathematical Finance, 30(3):995–1034, 2020.

[38] Semih Cayci, Siddhartha Satpathi, Niao He, and R Srikant. Sample complexity and overparameterization bounds for projection-free neural TD learning. arXiv preprint arXiv:2103.01391, 2021.

[39] Hyeong Soo Chang, Michael C Fu, Jiaqiao Hu, and Steven I Marcus. An adaptive sampling algorithm for solving markov decision processes. Operations Research, 53(1):126–139, 2005.

[40] Tianyi Chen, Kaiqing Zhang, Georgios B Giannakis, and Tamer Basar. Communication-efficient policy gradient methods for distributed reinforcement learning. IEEE Transactions on Control of Network Systems, 2021.

[41] Wei Chen, Dayu Huang, Ankur A Kulkarni, Jayakrishnan Unnikrishnan, Quanyan Zhu, Prashant Mehta, Sean Meyn, and Adam Wierman. Approximate dynamic programming using fluid and diffusion approximations with applications to power management. In Conference on Decision and Control, pages 3575–3580. IEEE, 2009.

[42] Rémi Coulom. Efficient selectivity and backup operators in monte-carlo tree search. In International Conference of Computers and Games, pages 72–83. Springer, 2007.

[43] Christoph Dann and Emma Brunskill. Sample complexity of episodic fixed-horizon reinforcement learning. In Advances in Neural Information Processing Systems, volume 28, pages 2818–2826, 2015.

[44] Christoph Dann, Tor Lattimore, and Emma Brunskill. Unifying pac and regret: Uniform pac bounds for episodic reinforcement learning. In Advances in Neural Information Processing Systems, volume 30, pages 5713–5723, 2017.

[45] Christoph Dann, Gerhard Neumann, Jan Peters, et al. Policy evaluation with temporal differences: A survey and comparison. Journal of Machine Learning Research, 15:809–883, 2014.

[46] Donald Dawson. Measure-valued Markov processes. In École d'été de probabilités de Saint-Flour XXI-1991, pages 1–260. Springer, 1993.

[47] Richard Dearden, Nir Friedman, and Stuart Russell. Bayesian Q-learning. In AAAI Conference on Artificial Intelligence, pages 761–768, 1998.

[48] Boualem Djehiche and Hamidou Tembine. Risk-sensitive mean-field type control under partial observation. In Stochastics of Environmental and Financial Economics, pages 243–263. Springer, Cham, 2016.

[49] Boualem Djehiche, Hamidou Tembine, and Raul Tempone. A stochastic maximum principle for risk-sensitive mean-field type control. IEEE Transactions on Automatic Control, 60(10):2640–2649, 2015.

[50] Mao Fabrice Djete, Dylan Possamaï, and Xiaolu Tan. McKean-Vlasov optimal control: the dynamic programming principle. arXiv preprint arXiv:1907.08860, 2019.

[51] Mo Dong, Tong Meng, Doron Zarchy, Engin Arslan, Yossi Gilad, Brighten Godfrey, and Michael Schapira. {PCC} vivace: Online-learning congestion control. In 15th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 18), pages 343–356, 2018.

[52] Kenji Doya. Reinforcement learning in continuous time and space. Neural Computation, 12(1):219–245, 2000.

[53] Kenji Doya, Kazuyuki Samejima, Ken-ichi Katagiri, and Mitsuo Kawato. Multiple model-based reinforcement learning. Neural computation, 14(6):1347–1369, 2002.

[54] Samah El-Tantawy, Baher Abdulhai, and Hossam Abdelgawad. Multiagent reinforcement learning for integrated network of adaptive traffic signal controllers (MARLIN-ATSC): Methodology and large-scale application on downtown Toronto. IEEE Transactions on Intelligent Transportation Systems, 14(3):1140–1150, 2013.

[55] Eyal Even-Dar, Yishay Mansour, and Peter Bartlett. Learning rates for Q-learning. Journal of machine learning Research, 5(1), 2003.

[56] Jerzy Filar and Koos Vrieze. Competitive Markov Decision Processes. Springer Science & Business Media, 2012.

[57] Wendell H Fleming and Halil Mete Soner. Controlled Markov processes and viscosity solutions, volume 25. Springer Science & Business Media, 2006.

[58] Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. In AAAI Conference on Artificial Intelligence, volume 32, 2018.

[59] Zuyue Fu, Zhuoran Yang, Yongxin Chen, and Zhaoran Wang. Actor-critic provably finds Nash equilibria of linear-quadratic mean-field games. arXiv preprint arXiv:1910.07498, 2019.

[60] Zuyue Fu, Zhuoran Yang, and Zhaoran Wang. Single-timescale actor-critic provably finds globally optimal policy. In International Conference on Learning Representations, 2020.

[61] David Gamarnik. Correlation decay method for decision, optimization, and inference in large-scale networks. In Theory Driven by Influential Applications, pages 108–121. INFORMS, 2013.

[62] David Gamarnik, David A Goldberg, and Theophane Weber. Correlation decay in random decision networks. Mathematics of Operations Research, 39(2):229–261, 2014.

[63] Josselin Garnier, George Papanicolaou, and Tzu-Wei Yang. Large deviations for a mean field model of systemic risk. SIAM Journal on Financial Mathematics, 4(1):151–184, 2013.

[64] Jürgen Gärtner. On the McKean-Vlasov limit for interacting diffusions. Mathematische Nachrichten, 137(1):197–248, 1988.

[65] Alborz Geramifard, Thomas J Walsh, Stefanie Tellex, Girish Chowdhary, Nicholas Roy, Jonathan P How, et al. A tutorial on linear function approximators for dynamic programming and reinforcement learning. Foundations and Trends in Machine Learning, 6(4):375–451, 2013.

[66] Maximilien Germain, Huyên Pham, and Xavier Warin. A level-set approach to the control of state-constrained mckean-vlasov equations: application to renewable energy storage and portfolio selection. arXiv preprint arXiv:2112.11059, 2021.

[67] Alison L Gibbs and Francis Edward Su. On choosing and bounding probability metrics. International statistical review, 70(3):419–435, 2002.

[68] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In International Conference on Artificial Intelligence and Statistics, pages 249–256, 2010.

[69] Arthur Gretton, Karsten Borgwardt, Malte J Rasch, Bernhard Scholkopf, and Alexander J Smola. A kernel method for the two-sample problem. arXiv preprint arXiv:0805.2368, 2008.

[70] Haotian Gu, Xin Guo, Xiaoli Wei, and Renyuan Xu. Dynamic programming principles for mean-field controls with learning. Operations Research, 2023.

[71] Maxime Guériau and Ivana Dusparic. Samod: Shared autonomous mobility-on-demand using decentralized reinforcement learning. In International Conference on Intelligent Transportation Systems, pages 1558–1563. IEEE, 2018.

[72] Xin Guo, Anran Hu, Renyuan Xu, and Junzi Zhang. Learning mean-field games. In Advances in Neural Information Processing Systems, pages 4966–4976, 2019.

[73] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In International Conference on Machine Learning, pages 1861–1870. PMLR, 2018.

[74] William B Haskell, Rahul Jain, and Dileep Kalathil. Empirical dynamic programming. Mathematics of Operations Research, 41(2):402–429, 2016.

[75] Pablo Hernandez-Leal, Bilal Kartal, and Matthew E Taylor. A survey and critique of multiagent deep reinforcement learning. Autonomous Agents and Multi-Agent Systems, 33(6):750–797, 2019.

[76] Karl Hinderer. Lipschitz continuity of value functions in Markovian decision processes. Mathematical Methods of Operations Research, 62(1):3–22, 2005.

[77] Junling Hu and Michael P Wellman. Nash q-learning for general-sum stochastic games. Journal of Machine Learning Research, 4:1039–1069, 2003.

[78] Ruimeng Hu and Thaleia Zariphopoulou. N-player and mean-field games in Itô-diffusion markets with competitive or homophilous interaction. arXiv preprint arXiv:2106.00581, 2021.

[79] Minyi Huang, Peter E Caines, and Roland P Malhamé. Large-population cost-coupled LQG problems with nonuniform agents: individual-mass behavior and decentralized ε-nash equilibria. IEEE Transactions on Automatic Control, 52(9):1560–1571, 2007.

[80] Minyi Huang, Roland P Malhamé, and Peter E Caines. Large population stochastic dynamic games: closed-loop McKean-Vlasov systems and the Nash certainty equivalence principle. Communications in Information & Systems, 6(3):221–252, 2006.

[81] Maximilian Hüttenrauch, Adrian Šošić, and Gerhard Neumann. Guided deep reinforcement learning for swarm systems. arXiv preprint arXiv:1709.06011, 2017.

[82] Krishnamurthy Iyer, Ramesh Johari, and Mukund Sundararajan. Mean-field equilibria of dynamic auctions with learning. Management Science, 60(12):2949–2970, 2014.

[83] Nathan Jay, Noga Rotman, Brighten Godfrey, Michael Schapira, and Aviv Tamar. A deep reinforcement learning perspective on internet congestion control. In International Conference on Machine Learning, pages 3050–3059, 2019.

[84] Ziwei Ji, Matus Telgarsky, and Ruicheng Xian. Neural tangent kernels, transportation mappings, and universal approximation. In International Conference on Learning Representations, 2020.

[85] Daniel Jiang, Emmanuel Ekwedike, and Han Liu. Feedback-based tree search for reinforcement learning. In International Conference on Machine Learning, pages 2284–2293. PMLR, 2018.

[86] Chi Jin, Zhuoran Yang, Zhaoran Wang, and Michael I Jordan. Provably efficient reinforcement learning with linear function approximation. In Conference on Learning Theory, pages 2137–2143. PMLR, 2020.

[87] Junqi Jin, Chengru Song, Han Li, Kun Gai, Jun Wang, and Weinan Zhang. Real-time bidding with multi-agent reinforcement learning in display advertising. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management, pages 2193–2201, 2018.

[88] Mark Kac. Foundations of kinetic theory. In Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability, volume 3, pages 171–197. University of California Press Berkeley and Los Angeles, California, 1956.

[89] Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In International Conference on Machine Learning, pages 267–274. PMLR, 2002.

[90] Jens Kober, J Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. International Journal of Robotics Research, 32(11):1238–1274, 2013.

[91] Levente Kocsis and Csaba Szepesvári. Bandit based Monte-Carlo planning. In Machine Learning, pages 282–293. Springer, 2006.

[92] Vijay R Konda and John N Tsitsiklis. Actor-critic algorithms. In Advances in Neural Information Processing Systems, volume 12, pages 1008–1014, 2000.

[93] Daniel Lacker. Mean field games via controlled martingale problems: existence of Markovian equilibria. Stochastic Processes and their Applications, 125(7):2856–2894, 2015.

[94] Daniel Lacker. Limit theory for controlled McKean–Vlasov dynamics. SIAM Journal on Control and Optimization, 55(3):1641–1672, 2017.

[95] Daniel Lacker and Thaleia Zariphopoulou. Mean-field and n-agent games for optimal investment under relative performance criteria. Mathematical Finance, 29(4):1003–1038, 2019.

[96] Jean-Michel Lasry and Pierre-Louis Lions. Mean field games. Japanese journal of mathematics, 2(1):229–260, 2007.

[97] Jean-Michel Lasry and Pierre-Louis Lions. Mean field games. Japanese Journal of Mathematics, 2(1):229–260, 2007.

[98] Martin Lauer and Martin Riedmiller. An algorithm for distributed reinforcement learning in cooperative multi-agent systems. In International Conference on Machine Learning. Citeseer, 2000.

[99] Mathieu Laurière and Olivier Pironneau. Dynamic programming for mean-field type control. Comptes Rendus Mathematique, 352(9):707–713, 2014.

[100] Minne Li, Zhiwei Qin, Yan Jiao, Yaodong Yang, Jun Wang, Chenxi Wang, Guobin Wu, and Jieping Ye. Efficient ridesharing order dispatching with mean field multi-agent reinforcement learning. In The World Wide Web Conference, pages 983–994, 2019.

[101] Yingying Li, Yujie Tang, Runyu Zhang, and Na Li. Distributed reinforcement learning for decentralized linear quadratic control: A derivative-free policy optimization approach. IEEE Transactions on Automatic Control, 2021.

[102] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971, 2015.

[103] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In International Conference on Learning Representations, 2016.

[104] Kaixiang Lin, Renyu Zhao, Zhe Xu, and Jiayu Zhou. Efficient large-scale fleet management via multi-agent deep reinforcement learning. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pages 1774–1783, 2018.

[105] Yiheng Lin, Guannan Qu, Longbo Huang, and Adam Wierman. Multi-agent reinforcement learning in stochastic networked systems. In Advances in Neural Information Processing Systems, volume 34, 2021.

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

[107] Michael L Littman. Friend-or-foe q-learning in general-sum games. In International Conference on Machine Learning, pages 322–328, 2001.

[108] Michael L Littman. A tutorial on partially observable markov decision processes. Journal of Mathematical Psychology, 53(3):119–125, 2009.

[109] Bo Liu, Ji Liu, Mohammad Ghavamzadeh, Sridhar Mahadevan, and Marek Petrik. Finite-sample analysis of proximal gradient td algorithms. In Uncertainty in Artificial Intelligence, pages 504–513. PMLR, 2015.

[110] Boyi Liu, Qi Cai, Zhuoran Yang, and Zhaoran Wang. Neural trust region/proximal policy optimization attains globally optimal policy. In Advances in Neural Information Processing Systems, volume 32, pages 10565–10576, 2019.

[111] Yao Liu, Adith Swaminathan, Alekh Agarwal, and Emma Brunskill. Off-policy policy gradient with stationary distribution correction. In Uncertainty in Artificial Intelligence, pages 1180–1190. PMLR, 2019.

[112] Ryan Lowe, Yi I Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In Advances in Neural Information Processing Systems, volume 30, pages 6382–6393, 2017.

[113] Yuwei Luo, Zhuoran Yang, Zhaoran Wang, and Mladen Kolar. Natural actor-critic converges globally for hierarchical linear quadratic regulator. arXiv preprint arXiv:1912.06875, 2019.

[114] Yongfeng Lv and Xuemei Ren. Approximate nash solutions for multiplayer mixed-zero-sum game with reinforcement learning. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 49(12):2739–2750, 2018.

[115] Hamid Maei, Csaba Szepesvari, Shalabh Bhatnagar, Doina Precup, David Silver, and Richard S Sutton. Convergent temporal-difference learning with arbitrary smooth function approximation. In Advances in Neural Information Processing Systems, volume 22, pages 1204–1212, 2009.

[116] Shie Mannor and John N Tsitsiklis. Algorithmic aspects of mean–variance optimization in Markov decision processes. European Journal of Operational Research, 231(3):645–653, 2013.

[117] Weichao Mao, Kaiqing Zhang, Qiaomin Xie, and Tamer Basar. Poly-hoot: Monte-carlo planning in continuous space mdps with non-asymptotic analysis. In Advances in Neural Information Processing Systems, volume 33, pages 4549–4559, 2020.

[118] Henry McKean. Propagation of chaos for a class of non-linear parabolic equations. lecture series in differential equations 7. Stochastic Differential Equations, pages 41–57, 1969.

[119] Henry P McKean. Propagation of chaos for a class of non-linear parabolic equations. Stochastic Differential Equations (Lecture Series in Differential Equations, Session 7, Catholic Univ., 1967), pages 41–57, 1967.

[120] Prashant Mehta and Sean Meyn. Q-learning and Pontryagin's minimum principle. In Conference on Decision and Control, pages 3598–3605. IEEE, 2009.

[121] Sean Meyn. Algorithms for optimization and stabilization of controlled markov chains. Sadhana, 24(4):339–367, 1999.

[122] Sean Meyn. Control techniques for complex networks. Cambridge University Press, 2008.

[123] Sean P Meyn. The policy iteration algorithm for average reward markov decision processes with general state space. IEEE Transactions on Automatic Control, 42(12):1663–1680, 1997.

[124] Charles A Micchelli, Yuesheng Xu, and Haizhang Zhang. Universal kernels. Journal of Machine Learning Research, 7(12):2651–2667, 2006.

[125] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In International Conference on Machine Learning, pages 1928–1937. PMLR, 2016.

[126] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. Nature, 518(7540):529–533, 2015.

[127] George E Monahan. State of the art: A survey of partially observable markov decision processes: theory, models, and algorithms. Management Science, 28(1):1–16, 1982.

[128] Médéric Motte and Huyên Pham. Mean-field markov decision processes with common noise and open-loop controls. arXiv preprint arXiv:1912.07883, 2019.

[129] Médéric Motte and Huyên Pham. Mean-field Markov decision processes with common noise and open-loop controls. The Annals of Applied Probability, 32(2):1421–1458, 2022.

[130] Rémi Munos and Andrew Moore. Variable resolution discretization in optimal control. Machine Learning, 49(2-3):291–323, 2002.

[131] Rémi Munos and Csaba Szepesvári. Finite-time bounds for fitted value iteration. Journal of Machine Learning Research, 9(27):815–857, 2008.

[132] Gergely Neu and Ciara Pike-Burke. A unifying view of optimism in episodic reinforcement learning. In Advances in Neural Information Processing Systems, volume 33, pages 1392–1403, 2020.

[133] Galo Nuño. Optimal social policies in mean field games. Applied Mathematics & Optimization, 76(1):29–57, 2017.

[134] Dirk Ormoneit and Peter Glynn. Kernel-based reinforcement learning in average-cost problems. IEEE Transactions on Automatic Control, 47(10):1624–1636, 2002.

[135] Dirk Ormoneit and Śaunak Sen. Kernel-based reinforcement learning. Machine Learning, 49(2-3):161–178, 2002.

[136] Afshin Oroojlooy and Davood Hajinezhad. A review of cooperative multi-agent deep reinforcement learning. Applied Intelligence, pages 1–46, 2022.

[137] Panos M Pardalos, Athanasios Migdalas, and Leonidas Pitsoulis. Pareto Optimality, Game Theory and Equilibria, volume 17. Springer Science & Business Media, 2008.

[138] Huyên Pham and Xiaoli Wei. Discrete time McKean–Vlasov control problem: a dynamic programming approach. Applied Mathematics & Optimization, 74(3):487–506, 2016.

[139] Matteo Pirotta, Marcello Restelli, and Luca Bascetta. Policy gradient in lipschitz Markov decision processes. Machine Learning, 100(2):255–283, 2015.

[140] Warren B Powell. Approximate Dynamic Programming: Solving the Curses of Dimensionality. John Wiley & Sons, 2011.

[141] Zhiwei Tony Qin, Hongtu Zhu, and Jieping Ye. Reinforcement learning for ridesharing: An extended survey. Transportation Research Part C: Emerging Technologies, 144:103852, 2022.

[142] Guannan Qu, Adam Wierman, and Na Li. Scalable reinforcement learning of localized policies for multi-agent networked systems. In Learning for Dynamics and Control, pages 256–266. PMLR, 2020.

[143] Michael Rabbat and Robert Nowak. Distributed optimization in sensor networks. In International Symposium on Information Processing in Sensor Networks, pages 20–27, 2004.

[144] Ali Rahimi and Benjamin Recht. Uniform approximation of functions with random bases. In Annual Allerton Conference on Communication, Control, and Computing, pages 555–561. IEEE, 2008.

[145] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning. In International Conference on Machine Learning, pages 4295–4304. PMLR, 2018.

[146] Naci Saldi. Discrete-time average-cost mean-field games on polish spaces. Turkish Journal of Mathematics, 44(2):463–480, 2020.

[147] Naci Saldi, Tamer Basar, and Maxim Raginsky. Markov–Nash equilibria in mean-field games with discounted cost. SIAM Journal on Control and Optimization, 56(6):4256–4287, 2018.

[148] Naci Saldi, Tamer Başar, and Maxim Raginsky. Approximate nash equilibria in partially observed stochastic games with mean-field interactions. Mathematics of Operations Research, 44(3):1006–1033, 2019.

[149] Naci Saldi, Tamer Başar, and Maxim Raginsky. Approximate markov-nash equilibria for discrete-time risk-sensitive mean-field games. Mathematics of Operations Research, 2020.

[150] Ahmad EL Sallab, Mohammed Abdou, Etienne Perot, and Senthil Yogamani. Deep reinforcement learning framework for autonomous driving. Electronic Imaging, 2017(19):70–76, 2017.

[151] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In International Conference on Machine Learning, pages 1889–1897. PMLR, 2015.

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

[153] Devavrat Shah and Qiaomin Xie. Q-learning with nearest neighbors. In Advances in Neural Information Processing Systems, pages 3111–3121, 2018.

[154] Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. Safe, multi-agent, reinforcement learning for autonomous driving. arXiv preprint arXiv:1610.03295, 2016.

[155] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, and Marc Lanctot. Mastering the game of go with deep neural networks and tree search. Nature, 529(7587):484, 2016.

[156] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In International Conference on Machine Learning, pages 387–395. PMLR, 2014.

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

[158] Satinder Singh, Tommi Jaakkola, Michael L Littman, and Csaba Szepesvári. Convergence results for single-step on-policy reinforcement-learning algorithms. Machine Learning, 38:287–308, 2000.

[159] Alex Smola, Arthur Gretton, Le Song, and Bernhard Schölkopf. A Hilbert space embedding for distributions. In International Conference on Algorithmic Learning Theory, pages 13–31. Springer, 2007.

[160] Kyunghwan Son, Daewoo Kim, Wan Ju Kang, David Earl Hostallero, and Yung Yi. Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In International Conference on Machine Learning, pages 5887–5896. PMLR, 2019.

[161] Suvrit Sra, Sebastian Nowozin, and Stephen J Wright. Optimization for Machine Learning. MIT Press, 2012.

[162] Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, and Graepel Thore. Value-decomposition networks for cooperative multi-agent learning based on team reward. In International Conference on Autonomous Agents and Multi-agent Systems, volume 3, pages 2085–2087, 2018.

[163] Richard S Sutton and Andrew G Barto. Reinforcement Learning: An Introduction. MIT press, 2018.

[164] Richard S Sutton, Hamid Maei, and Csaba Szepesvári. A convergent O(n) temporal-difference algorithm for off-policy learning with linear function approximation. In Advances in Neural Information Processing Systems, volume 21, pages 1609–1616, 2009.

[165] Richard S Sutton, Hamid Reza Maei, Doina Precup, Shalabh Bhatnagar, David Silver, Csaba Szepesvári, and Eric Wiewiora. Fast gradient-descent methods for temporal-difference learning with linear function approximation. In International Conference on Machine Learning, pages 993–1000, 2009.

[166] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In Advances in Neural Information Processing Systems, volume 99, pages 1057–1063, 2000.

[167] Csaba Szepesvári. Algorithms for Reinforcement Learning. Morgan and Claypool Publishers, 2010.

[168] Csaba Szepesvári and Michael L Littman. A unified analysis of value-function-based reinforcement-learning algorithms. Neural Computation, 11(8):2017–2060, 1999.

[169] Alain-Sol Sznitman. Topics in propagation of chaos. In Ecole d'été de Probabilités de Saint-Flour XIX-1989, pages 165–251. Springer, 1991.

[170] Gavin Taylor and Ronald Parr. Kernelized value function approximation for reinforcement learning. In International Conference on Machine Learning, pages 1017–1024, 2009.

[171] Gerald Tesauro et al. Temporal difference learning and td-gammon. Communications of the ACM, 38(3):58–68, 1995.

[172] John Tsitsiklis and Benjamin Van Roy. Analysis of temporal-diffference learning with function approximation. In Advances in Neural Information Processing Systems, volume 9, pages 1075–1081, 1996.

[173] Nelson Vadori, Sumitra Ganesh, Prashant Reddy, and Manuela Veloso. Calibration of shared equilibria in general sum partially observable markov games. In Advances in Neural Information Processing Systems, volume 33, pages 14118–14128, 2020.

[174] Hado Van Hasselt. Reinforcement learning in continuous state and action spaces. In Reinforcement Learning, pages 207–251. Springer, 2012.

[175] Cédric Villani. Optimal transport: old and new, volume 338. Springer, 2009.

[176] Oriol Vinyals, Igor Babuschkin, Junyoung Chung, Michael Mathieu, Max Jaderberg, Wojciech M Czarnecki, Andrew Dudzik, Aja Huang, Petko Georgiev, and Richard Powell. Alphastar: Mastering the real-time strategy game starcraft II. DeepMind Blog, page 2, 2019.

[177] Martin J Wainwright. High-dimensional statistics: A non-asymptotic viewpoint, volume 48. Cambridge University Press, 2019.

[178] Yi Wan, Abhishek Naik, and Richard S Sutton. Learning and planning in average-reward markov decision processes. In International Conference on Machine Learning, pages 10653–10662. PMLR, 2021.

[179] Bing-Chang Wang and Yong Liang. Robust mean field social control problems with applications in analysis of opinion dynamics. arXiv preprint arXiv:2002.12040, 2020.

[180] Hongbing Wang, Xiaojun Wang, Xingguo Hu, Xingzhi Zhang, and Mingzhu Gu. A multi-agent reinforcement learning approach to dynamic service composition. Information Sciences, 363:96–119, 2016.

[181] Lingxiao Wang, Qi Cai, Zhuoran Yang, and Zhaoran Wang. Neural policy gradient methods: Global optimality and rates of convergence. In International Conference on Learning Representations, 2020.

[182] Lingxiao Wang, Zhuoran Yang, and Zhaoran Wang. Breaking the curse of many agents: Provable mean embedding Q-iteration for mean-field reinforcement learning. In International Conference on Machine Learning, pages 10092–10103. PMLR, 2020.

[183] Christopher JCH Watkins. Learning From Delayed Rewards. PhD thesis, King's College, Cambridge, 1989.

[184] Christopher JCH Watkins and Peter Dayan. Q-learning. Machine Learning, 8(3-4):279–292, 1992.

[185] Christopher JCH Watkins and Peter Dayan. Q-learning. Machine Learning, 8(3-4):279–292, 1992.

[186] Chen-Yu Wei, Mehdi Jafarnia Jahromi, Haipeng Luo, Hiteshi Sharma, and Rahul Jain. Model-free reinforcement learning in infinite-horizon average-reward markov decision processes. In International Conference on Machine Learning, pages 10170–10180. PMLR, 2020.

[187] Marco A Wiering. Multi-agent reinforcement learning for traffic light control. In Proceedings of the 17th International Conference Machine Learning, pages 1151–1158, 2000.

[188] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine Learning, 8:229–256, 1992.

[189] Cong Wu, Jianfeng Zhang, et al. Viscosity solutions to parabolic master equations and McKean–Vlasov SDEs with closed-loop controls. Annals of Applied Probability, 30(2):936–986, 2020.

[190] Michael Wunder, Michael L Littman, and Monica Babes. Classes of multiagent Q-learning dynamics with epsilon-greedy exploration. In International Conference on Machine Learning, pages 1167–1174. Citeseer, 2010.

[191] Pan Xu, Felicia Gao, and Quanquan Gu. Sample efficient policy gradient methods with recursive variance reduction. In International Conference on Learning Representations, 2019.

[192] Pan Xu, Felicia Gao, and Quanquan Gu. An improved convergence analysis of stochastic variance-reduced policy gradient. In Uncertainty in Artificial Intelligence, pages 541–551. PMLR, 2020.

[193] Xin Xu, Dewen Hu, and Xicheng Lu. Kernel-based least squares policy iteration for reinforcement learning. IEEE Transactions on Neural Networks, 18(4):973–992, 2007.

[194] Yaodong Yang, Jianye Hao, Guangyong Chen, Hongyao Tang, Yingfeng Chen, Yujing Hu, Changjie Fan, and Zhongyu Wei. Q-value path decomposition for deep multiagent reinforcement learning. In International Conference on Machine Learning, pages 10706–10715. PMLR, 2020.

[195] Yaodong Yang, Rui Luo, Minne Li, Ming Zhou, Weinan Zhang, and Jun Wang. Mean field multi-agent reinforcement learning. arXiv preprint arXiv:1802.05438, 2018.

[196] Yaodong Yang and Jun Wang. An overview of multi-agent reinforcement learning from game theoretical perspective. arXiv preprint arXiv:2011.00583, 2020.

[197] Yaodong Yang, Ying Wen, Jun Wang, Liheng Chen, Kun Shao, David Mguni, and Weinan Zhang. Multi-agent determinantal Q-learning. In International Conference on Machine Learning, pages 10757–10766. PMLR, 2020.

[198] Zhuoran Yang, Kaiqing Zhang, Mingyi Hong, and Tamer Başar. A finite sample analysis of the actor-critic algorithm. In Conference on Decision and Control, pages 2759–2764. IEEE, 2018.

[199] Huibing Yin, Prashant G Mehta, Sean P Meyn, and Uday V Shanbhag. Learning in mean-field games. IEEE Transactions on Automatic Control, 59(3):629–644, 2013.

[200] Xinyu You, Xuanjie Li, Yuedong Xu, Hui Feng, Jin Zhao, and Huaicheng Yan. Toward packet routing with fully distributed multiagent deep reinforcement learning. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2020.

[201] Kaiqing Zhang, Bin Hu, and Tamer Basar. Policy optimization for h-2 linear control with h-infinity robustness guarantee: Implicit regularization and global convergence. SIAM Journal on Control and Optimization, 59(6):4081–4109, 2021.

[202] Kaiqing Zhang, Alec Koppel, Hao Zhu, and Tamer Basar. Global convergence of policy gradient methods to (almost) locally optimal policies. SIAM Journal on Control and Optimization, 58(6):3586–3612, 2020.

[203] Kaiqing Zhang, Yang Liu, Ji Liu, Mingyan Liu, and Tamer Başar. Distributed learning of average belief over networks using sequential observations. Automatica, 115:108857, 2020.

[204] Kaiqing Zhang, Zhuoran Yang, and Tamer Basar. Networked multi-agent reinforcement learning in continuous spaces. In Conference on Decision and Control, pages 2771–2776. IEEE, 2018.

[205] Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. In Handbook of Reinforcement Learning and Control, pages 321–384. Springer, 2021.

[206] Kaiqing Zhang, Zhuoran Yang, Han Liu, Tong Zhang, and Tamer Basar. Fully decentralized multi-agent reinforcement learning with networked agents. In International Conference on Machine Learning, pages 5872–5881. PMLR, 2018.

[207] Kaiqing Zhang, Zhuoran Yang, Han Liu, Tong Zhang, and Tamer Basar. Finite-sample analysis for decentralized batch multi-agent reinforcement learning with networked agents. IEEE Transactions on Automatic Control, 2021.

[208] Stephan Zheng, Alexander Trott, Sunil Srinivasa, Nikhil Naik, Melvin Gruesbeck, David C Parkes, and Richard Socher. The AI economist: Improving equality and productivity with AI-driven tax policies. arXiv preprint arXiv:2004.13332, 2020.

[209] Zhengyuan Zhou, Panayotis Mertikopoulos, Aris L Moustakas, Nicholas Bambos, and Peter Glynn. Robust power management via learning and game design. Operations Research, 69(1):331–345, 2021.

[210] Yuanheng Zhu and Dongbin Zhao. Online minimax q network learning for two-player zero-sum markov games. IEEE Transactions on Neural Networks and Learning Systems, 33(3):1228–1241, 2020.

[211] Difan Zou and Quanquan Gu. An improved analysis of training over-parameterized deep neural networks. In Advances in Neural Information Processing Systems, volume 32, pages 2055–2064, 2019.