Mathematics of Operations Research

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Mean-Field Multiagent Reinforcement Learning: A Decentralized Network Approach

Haotian Gu, Xin Guo, Xiaoli Wei, Renyuan Xu

To cite this article:
Haotian Gu, Xin Guo, Xiaoli Wei, Renyuan Xu (2024) Mean-Field Multiagent Reinforcement Learning: A Decentralized Network Approach. Mathematics of Operations Research

Published online in Articles in Advance 13 Mar 2024

. https://doi.org/10.1287/moor.2022.0055

https://pubsonline.informs.org/journal/moor

MATHEMATICS OF OPERATIONS RESEARCH
Articles in Advance, pp. 1–31
ISSN 0364-765X (print), ISSN 1526-5471 (online)

Mean-Field Multiagent Reinforcement Learning: A Decentralized Network Approach

Haotian Gu,[a] Xin Guo,[b,*] Xiaoli Wei,[c] Renyuan Xu[d]

[a] Department of Mathematics, University of California, Berkeley, Berkeley, California 94720; [b] Department of Industrial Engineering & Operations Research, University of California, Berkeley, Berkeley, California 94720; [c] Tsinghua Shenzhen International Graduate School, Shenzhen 518071, China; [d] Industrial & Systems Engineering, University of Southern California, Los Angeles, California 90089

*Corresponding author

Contact: haotian_gu@berkeley.edu, https://orcid.org/0000-0002-0268-7147 (HG); xinguo@berkeley.edu, https://orcid.org/0000-0002-3350-4606 (XG); tyswxl@gmail.com, https://orcid.org/0000-0002-4787-2856 (XW); renyuanx@usc.edu, https://orcid.org/0000-0003-4293-3450 (RX)

Received: February 15, 2022
Revised: March 14, 2023; September 16, 2023
Accepted: January 1, 2024
Published Online in Articles in Advance: March 13, 2024

MSC2020 Subject Classification: Primary: 49N80, 93A16, 68T05; secondary: 90B15, 60K35

https://doi.org/10.1287/moor.2022.0055

Abstract. One of the challenges for multiagent 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. Whereas 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 paper proposes a framework of localized training and decentralized execution to study MARL with the 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 afterward 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, in which the actor–critic approach with overparameterized neural networks is proposed. The convergence and sample complexity are 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 provable convergence guarantee.

Funding: X. Wei is partially supported by NSFC no. 12201343. R. Xu is partially supported by the NSF CAREER award DMS-2339240.

Keywords: multiagent reinforcement learning • mean-field • neural network approximation

1. Introduction

Multiagent reinforcement learning (MARL) has achieved substantial successes in a broad range of cooperative games and their applications, including coordination of robot swarms (Hüttenrauch et al. [30]), self-driving vehicles (Cabannes et al. [6], Shalev-Shwartz et al. [52]), real-time bidding games (Jin et al. [34]), ride-sharing (Li et al. [39]), power management (Zhou et al. [70]) and traffic routing (El-Tantawy et al. [17]). 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, that is, 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 a paradigm of centralized training with decentralized execution has achieved significant empirical successes, especially with the computational power of deep neural networks (Chen et al. [15], Foerster et al. [18], Lowe et al. [43], Rashid et al. [51], Vadori et al. [56], Yang et al. [60]). Such a training approach, however, suffers from the curse of dimensionality as the computational complexity grows exponentially with the number of agents (Zhang et al. [64]); it also requires extensive and costly communications between the central controller and all agents (Rabbat and Nowak [49]). Moreover, policies derived from the centralized training stage may not be robust in the execution phase (Zhang et al. [66]). 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 a network of agents, Zhang et al. [67] establishes the asymptotic convergence of decentralized actor–critic algorithms that are scalable in agent actions. Similar ideas are extended to the continuous space in which a deterministic policy gradient method is used (Zhang et al. [63]) with finite-sample analysis for such framework established in the batch setting (Zhang et al. [68]). Qu et al. [48] study a network of agents in which 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. A similar framework is considered for the linear quadratic case with local policy gradients conducted with zero order optimization and parallel updating (Li et al. [38]).

The second type of network, the network of states, is 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 an agent's perspective (Sunehag et al. [54]), 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 (You et al. [62]), traffic routing (Calderone and Sastry [8], Guériau and Dusparic [27]), resource allocations (Cao et al. [9]), and social economic systems (Zheng et al. [69]). However, there is no existing theoretical analysis for this type of decentralized MARL. Moreover, the dynamic nature of agents' relationships makes it difficult to adopt existing methodology from the static network of agents. The goal of this paper is, therefore, to fill the gap.

1.1. Motivating Example

To get the essence of the network of states, let us consider the following ride-hailing dispatch problem, studied empirically in Li et al. [39] via the MARL approach. In this problem, the rides/demands are exogenous, and drivers/supplies are distributed at different locations on a (transportation) network, in which the state includes the location of drivers within the graph and the driver's status of being idle or occupied. The driver's action is state-dependent: the driver can only take a new order when the driver's status is "idle" and when the pickup location is reachable within k steps, that is, within the k-hop neighborhood of the driver's current location on the graph. If the driver is occupied, the driver's only allowable action is to continue with the current order until 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 that take into account the supply–demand imbalance in both the origin and the destination of any impending trip: one rebate for the driver when the driver accepts orders in locations where the demand is higher than the supply and another rebated for the driver 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, and it measures the potential of the origin for the next ride.

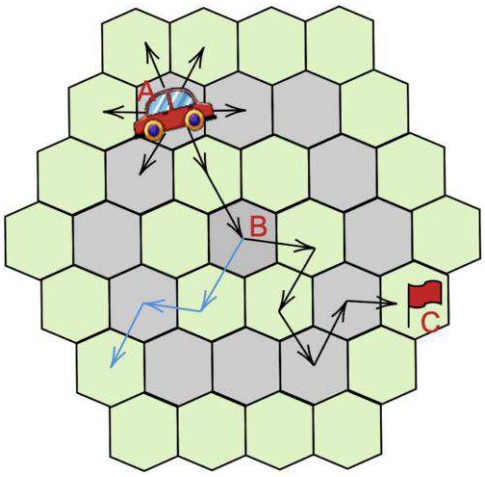
This example highlights a couple of features common in transportation networks: (1) the reward function relies on the aggregated information of drivers and riders with additional rebates for imbalance between the supply and the demand, and (2), the network is a hexagon grid system (Qin et al. [47]), shown in Figure 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.

1.2. Our Work

Motivated by this transportation network, this paper proposes and studies multiagent systems with a 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 afterward the learned decentralized policies that 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 (MDP) with

Figure 1. (Color online) Hexagon grid system.



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., Gamarnik [20], Gamarnik et al. [21]) and extensive analysis of network among heterogeneous agents (e.g., Lin et al. [40], Qu et al. [48]).

To design an efficient and scalable reinforcement learning algorithm for such a framework, the actor–critic approach with overparameterized 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 sample complexity of the proposed algorithm are 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 Wang et al. [57] to the multiagent setting.

1.3. Our Contribution

To the best of our knowledge, our work is the first neural network–based MARL algorithm with network structures and a provable convergence guarantee. In particular, our work contributes to two lines of research: MARL and centralized training, decentralized execution (CTDE).

First, we build a theoretical framework that incorporates network structures in the MARL framework and provide computationally efficient algorithms in which each agent only needs local information of neighborhood states to learn and execute the policy. In contrast, existing works for mean-field control with reinforcement learning require that each agent have the full information of the population distribution (Carmona et al. [11, 12], Gu et al. [25], Motte and Pham [45]) although, in most applications, agents only have access to partial or limited information (Yang et al. [61]).

Second, our work builds the theoretical foundation for the practically popular scheme of CTDE (Lowe et al. [43], Rashid et al. [51], Vadori et al. [56], Yang et al. [60]). The CTDE framework is first proposed in Lowe et al. [43] 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 an 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 Lowe et al. [43] for large N with local observations. Our paper analyzes for the first time with a 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 LTDE.

1.4. 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 $\mathbb{R}^{|\mathcal{X}|}$. Denote $[N] := \{1, 2, \ldots, N\}$. 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.

2. Mean-Field MARL with Local Dependency

The focus of this paper is to study a cooperative multiagent 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 the agent's present state to its connecting states. Moreover, the agent 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 (Carmona et al. [10], Casgrain and Jaimungal [13], Hu and Zariphopoulou [29], Lacker and Zariphopoulou [37]), energy markets (Aïd et al. [2], Germain et al. [23]), and auction systems (Guo et al. [28], Iyer et al. [31]).

2.1. Review of MARL

Let us first recall the cooperative MARL in an infinite time horizon, in which 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 $t = 0, 1, \ldots,$ the state of agent i $(= 1, 2, \ldots, N)$ is $s_t^i \in \mathcal{S}$ and the agent takes an action $a_t^i \in \mathcal{A}$. Given the current state profile $\boldsymbol{s}_t = (s_t^1, \ldots, s_t^N) \in \mathcal{S}^N$ and the current action profile $\boldsymbol{a}_t = (a_t^1, \ldots, a_t^N) \in \mathcal{A}^N$ of N agents, agent i receives a reward $r^i(\boldsymbol{s}_t, \boldsymbol{a}_t)$, and the agent's state changes 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 $a_t^i \sim \pi_t^i(s_t)$. 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

$$V(\boldsymbol{s}) = \max_{\boldsymbol{\pi}} \frac{1}{N} \sum_{i=1}^{N} v^i(\boldsymbol{s}, \boldsymbol{\pi}), \tag{2.1}$$

where

$$v^i(\boldsymbol{s}, \boldsymbol{\pi}) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r^i(\boldsymbol{s}_t, \boldsymbol{a}_t) | \boldsymbol{s}_0 = \boldsymbol{s}\right] \tag{2.2}$$

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 (2.1) is

$$V(\boldsymbol{s}) = \max_{\boldsymbol{a}\in\mathcal{A}^N} \left\{ \mathbb{E}\left[\frac{1}{N} \sum_{i=1}^{N} r^i(\boldsymbol{s}, \boldsymbol{a})\right] + \gamma \mathbb{E}_{\boldsymbol{s}'\sim P(\boldsymbol{s},\boldsymbol{a})}[V(\boldsymbol{s}')] \right\}, \tag{2.3}$$

with the population transition kernel $\boldsymbol{P} = (P^1, \ldots, P^N)$. 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

$$Q(\boldsymbol{s}, \boldsymbol{a}) = \mathbb{E}\left[\frac{1}{N} \sum_{i=1}^{N} r^i(\boldsymbol{s}, \boldsymbol{a})\right] + \gamma \mathbb{E}_{\boldsymbol{s}'\sim P(\boldsymbol{s},\boldsymbol{a})}[V(\boldsymbol{s}')], \tag{2.4}$$

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

$$Q(\boldsymbol{s},\boldsymbol{a}) = \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N} r^i(\boldsymbol{s},\boldsymbol{a})\right] + \gamma \mathbb{E}_{\boldsymbol{s}'\sim P(\boldsymbol{s},\boldsymbol{a})}\left[\max_{\boldsymbol{a}'\in\mathcal{A}^N} Q(\boldsymbol{s}',\boldsymbol{a}')\right]. \tag{2.5}$$

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

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 agent i's neighborhood $\mathcal{N}_{s_t^i}$ change stochastically over time.

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

$$r^i(\boldsymbol{s}_t,\boldsymbol{a}_t) := r(s_t^i,\ \mu_t(\mathcal{N}_{s_t^i}),\ a_t^i) \le r_{\max}, \quad i \in [N]; \tag{2.6}$$

agent i's state changes to a neighboring state $s_{t+1}^i \in \mathcal{N}_{s_t^i}$ according to a localized transition probability such that

$$s_{t+1}^i \sim P^i(\boldsymbol{s}_t,\boldsymbol{a}_t) := P(\cdot|s_t^i,\ \mu_t(\mathcal{N}_{s_t^i}),\ a_t^i), \quad i \in [N], \tag{2.7}$$

where $\mu_t(\cdot) = \sum_{i=1}^N \mathbf{1}(s_t^i = \cdot)/N \in \mathcal{P}^N(\mathcal{S}) := \{\mu \in \mathcal{P}(\mathcal{S}) : \mu(s) \in \{0,1/N,2/N,\ldots,N-1/N,1\}$ for all $s \in \mathcal{S}\}$ 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 $\mathcal{N}_{s_t^i}$, that is, $\mu_t(\mathcal{N}_{s_t^i}) := \{\mu_t(s)\}_{s\in\mathcal{N}_{s_t^i}}$.

Equations (2.6) and (2.7) indicate that the reward and the transition probability of agent i at time t depend on both agent i's individual information (a_t^i, s_t^i) and the mean-field of agent i's one-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 and localized meaning that agent i depends on the mean-field information of agent i's one-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.

2.2.1. 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:

$$a_t^i \sim \pi^i(\boldsymbol{s}_t) := \pi(s_t^i, \mu_t(s_t^i)) \in \mathcal{P}(\mathcal{A}), \quad i \in [N], \tag{2.8}$$

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 the agent's own state. This is more realistic than centralized policies that assume full access to the state information of all agents.

2.2.2. Value Function and Q-Function.

The goal for this mean-field MARL is to maximize the expected discounted accumulated reward averaged over all agents, that is,

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

subject to (2.6)–(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:

$$Q(\mu,h) := \underbrace{\mathbb{E}\left[\sum_{i=1}^{N}\frac{1}{N}r(s_0^i,\mu(\mathcal{N}_{s_0^i}),a_0^i)\middle|\boldsymbol{s}_0\sim\mu,\boldsymbol{a}_0\sim h(\boldsymbol{s}_0)\right]}_{\text{Expected reward of taking } \boldsymbol{a}_0=(a_0^1,\dots,a_0^N)}$$
$$+\underbrace{\mathbb{E}_{s_1^i\sim P(\cdot\mid s_0^i,\ \mu(\mathcal{N}_{s_0^i}),\ a_0^i)}\left[\sum_{t=1}^{\infty}\gamma^t\sum_{i=1}^{N}\frac{1}{N}r(s_t^i,\mu_t(\mathcal{N}_{s_t^i}),a_t^i)\middle|a_t^i\sim\pi_t^\star\right]}_{\text{Expected reward of playing optimally thereafter } a_t^i\sim\pi_t^\star}, \tag{2.9}$$

where $\mu(\cdot)=\sum_{i=1}^{N}\mathbf{1}(s_0^i=\cdot)/N$ is the initial empirical state distribution and $h(s)(a)=\sum_{i=1}^{N}\mathbf{1}(s_0^i=s,a_0^i=a)/\sum_{i=1}^{N}\mathbf{1}(s_0^i=s)$ is a decentralized policy representing the proportion of agents in state s that take 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,

$$h(s)\in\mathcal{P}^{N\cdot\mu(s)}(\mathcal{A}) := \left\{\varsigma\in\mathcal{P}(\mathcal{A}):\varsigma(a)\in\left\{0,\frac{1}{N\cdot\mu(s)},\dots,\frac{N\cdot\mu(s)-1}{N\cdot\mu(s)}\right\}\text{ for all } a\in\mathcal{A}\right\}\subset\mathcal{P}(\mathcal{A}),$$

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 2.1. Before further analysis, let us recall some important properties for the value function in (MF-MARL) and the Q-function in (2.9).

First is the dynamics programming principle for the mean-field Q function. Take an N-player game; the value function for any $\boldsymbol{s}:=(s_1,s_2,\dots,s_N)\in\mathcal{S}^N$ 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 $\mu(\cdot)=\sum_{i=1}^{N}\mathbf{1}(s_0^i=\cdot)/N$ is the sufficient statistic for the dynamic programming principle (DPP) of the corresponding value function. Analogously, for the mean-field Q function, it is shown in Gu et al. [25, 26] that the empirical state distribution $\mu(\cdot)=\sum_{i=1}^{N}\mathbf{1}(s_0^i=\cdot)/N$ and the empirical action distribution $h:\mathcal{S}\times\mathcal{A}\to\mathbb{R}$, $h(s)(a)=\sum_{i=1}^{N}\mathbf{1}(s_0^i=s,a_0^i=a)/\sum_{i=1}^{N}\mathbf{1}(s_0^i=s)$ are 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.

Second, $Q(\mu,h)$ defined in (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 (2.9) is independent of the number of agents N in the system, which renders it more scalable in the large population regime. This differs from the Q-function defined in (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 Gu et al. [25, 26].)

3. Analysis of MF-MARL with Local Dependency

The theoretical study of this mean-field MARL with local dependency (Section 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 3.1). The second is the Bellman equation for the value function and the Q-function on the probability measure space (Section 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 3.3).

3.1. MDP on Network of States

This section shows that the mean-field MARL (2.6)–(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 setup and to regroup these N agents according to their states. This regrouping translates (MF-MARL) with N agents into a networked MDP with $|\mathcal{S}|$ agent teams, indexed by their states.

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, and each agent i in state s independently chooses action $a_i \sim \pi(s, \mu(s))$ according to the individual-decentralized policy $\pi(s, \mu(s)) \in \mathcal{P}(\mathcal{A})$ in (2.8). Therefore, the empirical action distribution of $\{a_1, \ldots, a_{N\cdot\mu(s)}\}$ 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

$$\begin{aligned}
&\mathbb{P}(h(s) \text{ is the empirical action distribution of } \{a_1, \ldots, a_{N\cdot\mu(s)}\}, a_i \overset{i.i.d}{\sim} \pi(s, \mu(s))) \\
&= \mathbb{P}(\text{for each } a \in \mathcal{A}, a \text{ appears } N \cdot \mu(s)h(s)(a) \text{ times in } \{a_1, \ldots, a_{N\cdot\mu(s)}\}, a_i \overset{i.i.d}{\sim} \pi(s, \mu(s))) \\
&= \frac{(N \cdot \mu(s))!}{\prod_{a\in\mathcal{A}}(N \cdot \mu(s)h(s)(a))!} \prod_{a\in\mathcal{A}} (\pi(s, \mu(s))(a))^{N\cdot\mu(s)h(s)(a)}. \qquad (3.1)
\end{aligned}$$

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 (2.8) induces a team-decentralized policy of the following form:

$$\Pi_s(h(s)|\mu(s)) = \frac{(N \cdot \mu(s))!}{\prod_{a\in\mathcal{A}}(N \cdot \mu(s)h(s)(a))!} \prod_{a\in\mathcal{A}} (\pi(s, \mu(s))(a))^{N\cdot\mu(s)h(s)(a)}, \qquad (3.2)$$

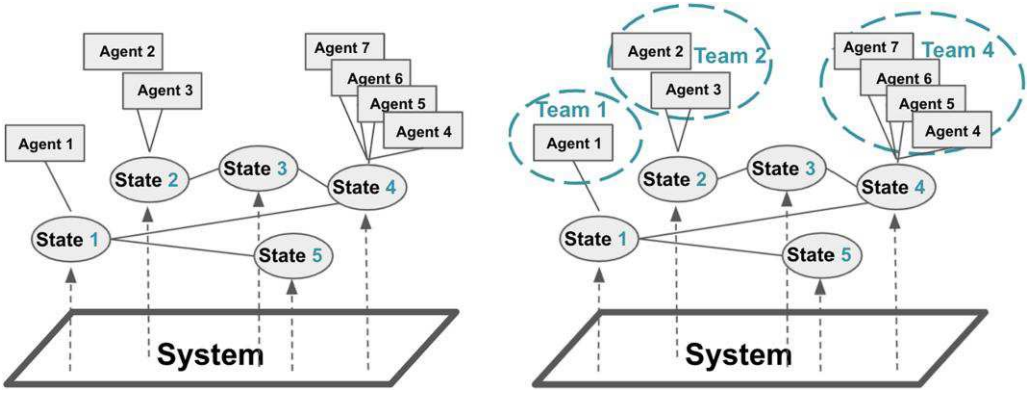
where $h(s) \in \mathcal{P}^{N\cdot\mu(s)}(\mathcal{A})$. Conversely, given a team-decentralized policy $\Pi_s(\cdot|\mu(s))$, one can recover the individual-decentralized policy $\pi(s, \mu(s))$ by choosing appropriate $h(s) \in \mathcal{P}^{N\cdot\mu(s)}(\mathcal{A})$ and querying the value of $\Pi_s(h(s)|\mu(s))$: 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 (3.2), $\Pi_s(h_i(s)|\mu(s)) = (\pi(s, \mu(s))(a_i))^{N\cdot\mu(s)}$, 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

$$\Pi(h|\mu) := \prod_{s\in\mathcal{S}} \Pi_s(h(s)|\mu(s)), \qquad (3.3)$$

then $\mathfrak{u}$, the admissible policy set of individual-decentralized policies in the form of (2.8), is now replaced by $\mathfrak{U}$, the set of all team-decentralized policies Π induced from $\pi \in \mathfrak{u}$ through (3.2) and (3.3). In addition, denote the set of all

Figure 2. (Color online) Left: MF-MARL problem (2.6)–(2.8). Right: Reformulation of team game (3.2)–(3.6).



state–action distribution pairs as

$$\Xi := \cup_{\mu\in\mathcal{P}^N(\mathcal{S})}\{\zeta = (\mu, h) : h \in \mathcal{H}^N(\mu)\}, \tag{3.4}$$

and moreover, from the team perspective, the transition probability in (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 (2.7) such that

$$\mu_{t+1} \sim \mathbf{P}^N(\cdot|\mu_t, h_t). \tag{3.5}$$

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 two-hop neighborhood of s, and $h_t(\mathcal{N}_s)$. More specifically, each agent can only move from the agent's 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 outflow of agents from state s to neighboring states in $\mathcal{N}_s$; (2) the inflow of agents from states in $\mathcal{N}_s$ to state s. The outflow of agents depends on the actions of the agents in state s as well as the transition kernel. Because both the policy and the transition kernel only depend on information $\mu(\mathcal{N}_s)$, the outflow has a one-hop neighbor dependence. Similarly, the inflow from any state $s' \in \mathcal{N}_s$ depends on the information $\mu(\mathcal{N}_{s'})$, which is contained in $\mu(\mathcal{N}_s^2)$ because $\mathcal{N}_{s'} \subset \mathcal{N}_s^2$ for any $s' \in \mathcal{N}_s$. Therefore, the inflow to s has a two-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 one-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

$$r_s(\mu(\mathcal{N}_s), h(s)) = \sum_{a\in\mathcal{A}} r(s, \mu(\mathcal{N}_s), a)h(s)(a), \tag{3.6}$$

which depends on the state s and its one-hop neighborhood, and define the maximal expected discounted accumulative localized team rewards over all teams as

$$\tilde{V}(\mu) := \sup_{\Pi\in\mathfrak{U}} \tilde{V}^{\Pi}(\mu) = \sup_{\Pi\in\mathfrak{U}} \mathbb{E}\left[\sum_{t=0}^{\infty}\sum_{s\in\mathcal{S}} \gamma^t r_s(\mu_t(\mathcal{N}_s), h_t(s))\middle|\mu_0 = \mu\right]. \tag{3.7}$$

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 (3.7) and can, thus, reformulate the (MF-MARL) problem as an equivalent MDP of (3.2)–(3.7) with $|\mathcal{S}|$ teams, the latter denoted as (MF-DEC-MARL). (The proof is detailed in Appendix A.). See Figure 2 for illustration.

Lemma 3.1 (Value Function and Q-Function Decomposition).

$$V(\mu) = \tilde{V}(\mu) = \sup_{\Pi\in\mathfrak{U}} \sum_{s\in\mathcal{S}} \tilde{V}_s^{\Pi}(\mu), \tag{3.8}$$

where $h_t \sim \Pi(\cdot|\mu_t)$, $\mu_{t+1} \sim \mathbf{P}^N(\cdot|\mu_t, h_t)$, *and*

$$\tilde{V}_s^{\Pi}(\mu) = \mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t r_s(\mu_t(\mathcal{N}_s), h_t(s))\middle|\mu_0 = \mu\right] \tag{3.9}$$

is called the value function under policy Π *for team* s. *Similarly,*

$$Q^{\Pi}(\mu, h) := \mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t\sum_{s\in\mathcal{S}} r_s(\mu_t(\mathcal{N}_s), h_t(s))\middle|\mu_0 = \mu, h_0 = h\right] = \sum_{s\in\mathcal{S}} Q_s^{\Pi}(\mu, h), \tag{3.10}$$

where

$$Q_s^{\Pi}(\mu, h) = \mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t r_s(\mu_t(\mathcal{N}_s), h_t(s))\middle|\mu_0 = \mu, h_0 = h\right], \tag{3.11}$$

is the Q-function under policy Π *for team* s, *called the team-decentralized Q-function.*

The decomposition for the Q-function in (3.10) is one of the key elements to allow for approximation of $Q_s^{\Pi}(\mu,h)$ 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 is clear in the Section 3.3.

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 Gu et al. [26] after establishing the dynamic programming principle on an appropriate probability measure space.

Lemma 3.2 (Bellman Equation for Q-Function). *The Q-function defined in* (2.9) *satisfies*

$$Q(\mu,h)=\mathbb{E}\left[\sum_{i=1}^{N}\frac{1}{N}r(s_0^i,\mu(\mathcal{N}_{s_0^i}),a_0^i)\middle|\boldsymbol{s}_0,\boldsymbol{a}_0\right]+\gamma\mathbb{E}_{s_1^i\sim P(\cdot\mid s_0^i,\ \mu(\mathcal{N}_{s_0^i}),\ a_0^i)}\left[\sup_{h'\in\mathcal{H}^N(\mu_1)}Q(\mu_1,h')\right]. \tag{3.12}$$

Here, $\mu_1(\cdot)=\sum_{i=1}^{N}\mathbf{1}(s_1^i=\cdot)/N$ *is the empirical state distribution at time 1.*

Note that the Bellman equation (3.12) is for the Q-function defined in (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 the Q-function (3.10) and the updating rule based on the team-decentralized Q-function (3.11). The corresponding Bellman equation for the team-decentralized Q-function (3.11) follows.

Lemma 3.3. *Given a policy* $\Pi\in\mathfrak{U}$, Q_s^{Π} *defined in* (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*

$$\mathcal{T}_s^{\Pi}Q_s^{\Pi}(\mu,h)=\mathbb{E}_{\mu'\sim\mathbf{P}^N(\cdot\mid\mu,h),\,h'\sim\Pi(\cdot\mid\mu)}[r_s(\mu,h)+\gamma\cdot Q_s^{\Pi}(\mu',h')],\ \forall(\mu,h)\in\Xi. \tag{3.13}$$

These Bellman equations are the basis for general Q-function–based algorithms in mean-field MARL.

3.3. Exponential Decay of Q-Function

This section shows 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 (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 the sth k-hop neighborhood. Next, rewrite any given empirical state distribution $\mu\in\mathcal{P}^N(\mathcal{S})$ as $(\mu(\mathcal{N}_s^k),\mu(\mathcal{N}_s^{-k}))$ and, similarly, $h\in\mathcal{H}^N(\mu)$ as $(h(\mathcal{N}_s^k),h(\mathcal{N}_s^{-k}))$.

Definition 3.1. The Q^{Π} is said to have a (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 $h(\mathcal{N}_s^k)=h'(\mathcal{N}_s^k)$

$$\left|Q_s^{\Pi}(\mu(\mathcal{N}_s^k),\mu(\mathcal{N}_s^{-k}),h(\mathcal{N}_s^k),h(\mathcal{N}_s^{-k}))-Q_s^{\Pi}(\mu(\mathcal{N}_s^k),\mu'(\mathcal{N}_s^{-k}),h(\mathcal{N}_s^k),h'(\mathcal{N}_s^{-k}))\right|\le c\rho^{k+1}.$$

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. Its proof is given in Appendix B.

Lemma 3.4. *When the reward* r_s *in* (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.*

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 that 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:

$$\widehat{Q}_s^{\Pi}(\mu(\mathcal{N}_s^k),h(\mathcal{N}_s^k))=\sum_{\mu(\mathcal{N}_s^{-k}),h(\mathcal{N}_s^{-k})}[w_s(\mu(\mathcal{N}_s^{-k}),h(\mathcal{N}_s^{-k});\mu(\mathcal{N}_s^k),h(\mathcal{N}_s^k))\cdot Q_s^{\Pi}(\mu(\mathcal{N}_s^k),\mu(\mathcal{N}_s^{-k}),h(\mathcal{N}_s^k),h(\mathcal{N}_s^{-k}))], \tag{Local Q-function}$$

where $w_s(\mu(\mathcal{N}_s^{-k}), h(\mathcal{N}_s^{-k}); \mu(\mathcal{N}_s^k), h(\mathcal{N}_s^{-k}))$ are any nonnegative weights of

$$\sum_{\mu(\mathcal{N}_s^{-k}), h(\mathcal{N}_s^{-k})} w_s(\mu(\mathcal{N}_s^{-k}), h(\mathcal{N}_s^{-k}); \mu(\mathcal{N}_s^k), h(\mathcal{N}_s^k)) = 1$$

for any $\mu(\mathcal{N}_s^k)$ and $h(\mathcal{N}_s^k)$.

Then, direct computation yields the following proposition.

Proposition 3.1. *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 3.1 holds. Then, for any $\mu \in \mathcal{P}^N(\mathcal{S})$ and $h \in \mathcal{H}^N(\mu)$,*

$$|\widehat{Q}_s^{\Pi}(\mu(\mathcal{N}_s^k), h(\mathcal{N}_s^k)) - Q_s^{\Pi}(\mu, h)| \le c\rho^{k+1}. \tag{3.14}$$

Moreover, (3.14) 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 $\widehat{Q}_s^{\Pi}(\mu(\mathcal{N}_s^k), h(\mathcal{N}_s^k))$ is defined as a weighted average of Q_s^{Π} over all (μ, h)-pairs that agree with $(\mu(\mathcal{N}_s^k), h(\mathcal{N}_s^k))$ 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 3.1 (Exponential Decay Property). In a discounted reward setting (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 (3.2)–(3.7). For problems of finite-time or infinite horizons with ergodic reward functions, this property can be established by imposing an additional Lipschitz condition on the transition kernel. (See Qu et al. [48, theorem 1] for a network of heterogeneous agents and $\gamma = 1$).

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 are parameterized by two-layer neural networks with parameters ω_s and θ_s respectively (Section 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 4.3): the critic aims to find a proper estimate for the localized Q-function under a fixed policy (parameterized by θ), whereas 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; afterward, agents in the system 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, overparameterization 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.

4.1. Basic Setup

4.1.1. 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 $\{\theta : \theta \in \Theta\}$.

4.1.2. Initialization. Let us also assume that the initial state distribution μ_0 of N agents is sampled from a given distribution P_0 over $\mathcal{P}^N(\mathcal{S})$, that is, $\mu_0 \sim P_0$, and define the expected total reward function $J(\theta)$ under policy Π^θ by

$$J(\theta) = \mathbb{E}_{\mu_0\sim P_0}[\tilde{V}^{\Pi^\theta}(\mu_0)]. \tag{4.1}$$

4.1.3. Visitation Measure. Denote ν_θ as the stationary distribution on Ξ of the Markov process (3.5) induced by Π^θ.

Similar to the single-agent reinforcement learning problem (Agarwal et al. [1], Fu et al. [19]), 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

$$\sigma_\theta(\mu,h) := (1-\gamma)\cdot\sum_{t=0}^{\infty}\gamma^t\cdot\mathbb{P}(\mu_t=\mu,h_t=h|\Pi^\theta), \tag{4.2}$$

where $\mu_0\sim P_0$, $h_t\sim\Pi^\theta(\cdot|\mu_t)$, and $\mu_{t+1}\sim\mathbf{P}^N(\cdot|\mu_t,h_t)$.

4.1.4. Policy Gradient Theorem. In order to find the optimal parameterized policy Π^θ that maximizes the expected total reward function $J(\theta)$, the policy optimization step searches 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 (4.2), which is indirectly determined by Π^θ.

A simple and elegant result called the policy gradient theorem (Lemma 4.1) proposed in Sutton et al. [55], reformulates the gradient $\nabla J(\theta)$ in terms of Q^{Π_θ} in (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 4.1 (Sutton et al. [55]). $\nabla J(\theta)=\frac{1}{1-\gamma}\mathbb{E}_{\sigma_\theta}[Q^{\Pi^\theta}(\mu,h)\nabla\log\Pi^\theta(h|\mu)]$.

Now, direct implementation of the actor–critic algorithm with the centralized policy gradient theorem in Lemma 4.1 suffers from high sample complexity because of the dimension of the Q-function. Instead, we show that the exponential decay property of the Q-function allows efficient approximation of the policy gradient via localization and hence a scalable algorithm to solve (MF-MARL).

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 a convergence guarantee.

4.2.1. 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 $\tilde{f}(x;W,b)$ with input $x\in\mathcal{X}$ and width $M\in\mathbb{N}$ takes the form of

$$\tilde{f}(x;W,b)=\frac{1}{\sqrt{M}}\sum_{m=1}^{M}b_m\cdot\mathrm{ReLU}(x\cdot[W]_m). \tag{4.3}$$

Here, the scaling factor $1/\sqrt{M}$ called the Xavier initialization (Glorot and Bengio [24]) 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 $W=([W]_1^\top,\ldots,[W]_M^\top)^\top\in\mathbb{R}^{M\times d_x}$ in (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 (Allen-Zhu et al. [4], Cai et al. [7], Wang et al. [57]) to fix b throughout the training and only to update W in the sequel. Consequently, denote $\tilde{f}(x;W,b)$ 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.

4.2.2. 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 (4.3), the team-decentralized policy $\Pi_s^{\theta_s}$, called the actor, is parameterized in the form of an energy-based policy,

$$\Pi_s^{\theta_s}(h(s)|\mu(s))=\frac{\exp[\tau\cdot f((\mu(s),h(s));\theta_s)]}{\sum_{h'(s)\in\mathcal{P}^{N\cdot\mu(s)}(\mathcal{A})}\exp[\tau\cdot f((\mu(s),h'(s));\theta_s)]}, \tag{4.4}$$

where τ is the temperature parameter and f is the energy function.

To study the policy gradient for (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 (4.6) is crucial in the convergence analysis of Theorems 5.1 and 5.2. Specifically, rewrite the two-layer neural network in (4.3) as

$$f(\zeta_s;\theta_s)=\frac{1}{\sqrt{M}}\sum_{m=1}^{M}\mathrm{ReLU}(\zeta_s^\top[\theta_s]_m)=\frac{1}{\sqrt{M}}\sum_{m=1}^{M}\mathbb{1}\{\zeta_s^\top[\theta_s]_m>0\}\cdot\zeta_s^\top[\theta_s]_m.:=\phi_{\theta_s}(\zeta_s)^\top\theta_s. \tag{4.5}$$

Then, the feature mapping $\phi_{\theta_s}=([\phi_{\theta_s}]_1^\top,\ldots,[\phi_{\theta_s}]_M^\top)^\top:\mathbb{R}^{d_{\zeta_s}}\to\mathbb{R}^{M\times d_{\zeta_s}}$ may take the following form:

$$[\phi_{\theta_s}]_m(\zeta_s)=\frac{1}{\sqrt{M}}\cdot\mathbb{1}\{\zeta_s^\top[\theta_s]_m>0\}\cdot\zeta_s. \tag{4.6}$$

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. Because $f(\zeta_s;\theta_s)$ 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 (4.6) is different from the linear feature literature (Geramifard et al. [22], Jin et al. [33]). This is because the feature mapping ϕ_{θ_s} in (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

$$\Phi(\theta,s,\mu,h):=\phi_{\theta_s}(\mu(s),h(s))-\mathbb{E}_{h(s)'\sim\Pi_s^{\theta_s}(\cdot\,|\,\mu(s))}[\phi_{\theta_s}(\mu(s),h'(s))]. \tag{4.7}$$

Note that, when policy Π^θ takes the energy-based form (4.4), $\Phi=\frac{1}{\tau}\nabla_\theta\log\Pi^\theta$.

Lemma 4.2. *For any $\theta\in\Theta$, $s\in\mathcal{S}$, $\mu\in\mathcal{P}^N(\mathcal{S})$ and $h\in\mathcal{H}^N(\mu)$, $\|\Phi(\theta,s,\mu,h)\|_2\le 2$, and*

$$\nabla_{\theta_s}J(\theta)=\frac{\tau}{1-\gamma}\cdot\mathbb{E}_{\sigma_\theta}[Q^{\Pi^\theta}(\mu,h)\cdot\Phi(\theta,s,\mu,h)]. \tag{4.8}$$

Moreover, for each $s\in\mathcal{S}$, define the following localized policy gradient:

$$g_s(\theta)=\frac{\tau}{1-\gamma}\mathbb{E}_{\sigma_\theta}\left[\left[\sum_{y\in\mathcal{N}_s^k}\widehat{Q}_y^{\Pi^\theta}(\mu(\mathcal{N}_y^k),h(\mathcal{N}_y^k)\right]\cdot\Phi(\theta,s,\mu,h)\right], \tag{4.9}$$

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

$$\|g_s(\theta)-\nabla_{\theta_s}J(\theta)\|\le\frac{c_0\tau|\mathcal{S}|}{1-\gamma}\rho^{k+1}. \tag{4.10}$$

4.2.3. Neural Q-Function. Note that $\widehat{Q}_s^{\Pi^\theta}$ in (Local Q-function) is unknown a priori. To obtain the localized policy gradient (4.9), the neural network (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.

4.3. Actor–Critic

4.3.1. 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 (3.13) is for $Q_s^{\Pi^\theta}$ instead of $\widehat{Q}_s^{\Pi^\theta}$. Indeed, $Q_s^{\Pi^\theta}$ takes (μ,h) as the input, whereas $\widehat{Q}_s^{\Pi^\theta}$ 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 (3.13). 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 is updated to minimize the error:

$$(\delta_{s,t})^2 = [Q_s(\mu_t(\mathcal{N}_s^k), h_t(\mathcal{N}_s^k); \omega_s) - r_s(\mu_t(\mathcal{N}_s), h_t(s)) - \gamma \cdot Q_s(\mu_{t+1}(\mathcal{N}_s^k), h_{t+1}(\mathcal{N}_s^k); \omega_s)]^2.$$

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 5.1.)

The neural critic update takes the iterative forms of

$$\omega_s(t+1/2) \leftarrow \omega_s(t) - \eta_{\text{critic}} \cdot \delta_{s,t} \cdot \nabla_{\omega_s} Q_s(\mu_t(\mathcal{N}_s^k), h_t(\mathcal{N}_s^k); \omega_s), \tag{4.11}$$

$$\omega_s(t+1) \leftarrow \arg\min_{\omega \in \mathcal{B}_s^{\text{critic}}} \|\omega - \omega_s(t+1/2)\|_2, \tag{4.12}$$

$$\bar{\omega}_s \leftarrow (t+1)/(t+2) \cdot \bar{\omega}_s + 1/(t+2) \cdot \omega_s(t+1), \tag{4.13}$$

in which η_{critic} is the learning rate. Here, (4.11) is the stochastic semigradient step, (4.12) is a projection to the parameter space $\mathcal{B}_s^{\text{critic}} := \{\omega_s \in \mathbb{R}^{M \times d_{\zeta_s^k}} : \|\omega_s - \omega_s(0)\|_\infty \le R/\sqrt{M}\}$ for some $R > 0$, and (4.13) is the averaging step. This critic update is summarized in Algorithm 1.

Algorithm 1 (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 $b_m \sim \text{Unif}(\{-1, 1\})$, $[\omega_s(0)]_m \sim N(0, I_{d_{\zeta_s^k}}/d_{\zeta_s^k})$, $\bar{\omega}_s = \omega_s(0)$.
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 $\zeta_{s,t}^k = (\mu_t(\mathcal{N}_s^k), h_t(\mathcal{N}_s^k))$, $\zeta_{s,t}^{k'} = (\mu_t'(\mathcal{N}_s^k), h_t'(\mathcal{N}_s^k))$.
    Residual calculation: $\delta_{s,t} \leftarrow Q_s(\zeta_{s,t}^k; \omega_s(t)) - r_s(\mu_t(\mathcal{N}_s), h_t(s)) - \gamma \cdot Q_s(\zeta_{s,t}^{k'}; \omega_s(t))$.
    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: $\omega_s(t+1) \leftarrow \arg\min_{\omega \in \mathcal{B}_s^{\text{critic}}} \|\omega - \omega_s(t+1/2)\|_2$.
    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: $Q_s(\cdot; \bar{\omega}_s)$, $\forall s \in \mathcal{S}$.

4.3.2. 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 (4.2) and define an estimator $\widehat{\Phi}(\theta, s, \mu_l, h_l)$ of $\Phi(\theta, s, \mu_l, h_l)$ in (4.7):

$$\widehat{\Phi}(\theta, s, \mu_l, h_l) = \phi_{\theta_s}(\mu_l(s), h_l(s)) - \mathbb{E}_{\Pi_s^{\theta_s}}[\phi_{\theta_s}(\mu_l(s), h'(s))].$$

By Lemma 4.2, one can compute the following estimator of $g_s(\theta(t))$ defined in (4.9):

$$\widehat{g}_s(\theta(t)) = \frac{\tau}{(1-\gamma)B} \sum_{l \in [B]} \left[\left[\sum_{y \in \mathcal{N}_s^k} Q_y\left(\mu_l(\mathcal{N}_y^k), h_l(\mathcal{N}_y^k); \bar{\omega}_y\right) \right] \cdot \widehat{\Phi}(\theta(t), s, \mu_l, h_l) \right]. \tag{4.14}$$

This estimator $\widehat{g}_s$ in (4.14) 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 $\mathcal{B}_s^{\text{actor}} := \{\theta_s \in \mathbb{R}^{M \times d_{\zeta_s}} : \|\theta_s - \theta_s(0)\|_\infty \le R/\sqrt{M}\}$.

This actor update is summarized in Algorithm 2.

4.3.3. Sampling from ν_θ and the Visitation Measure σ_θ. In Algorithms 1 and 2, 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 a weakly

dependent sequence of samples (Bhandari et al. [5]) 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, Konda and Tsitsiklis [36] suggest 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, Liu et al. [42] propose an importance sampling–based algorithm that enables off-policy evaluation with low variance.

Algorithm 2 (Localized Training, Decentralized Execution Neural Actor–Critic)

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.
Initialize: For all $m \in [M]$ and $s \in \mathcal{S}$, sample $b_m \sim \text{Unif}(\{-1,1\})$, $[\theta_s(0)]_m \sim N(0, I_{d_{\zeta_s}}/d_{\zeta_s})$.
for $t = 1$ to T_{actor} **do**
  Define the decentralized policy $\Pi_s^{\theta_s}$ for each state $s \in \mathcal{S}$,

$$\Pi_s^{\theta_s}(h(s)\,|\,\mu(s)) = \frac{\exp[\tau \cdot f((\mu(s), h(s)); \theta_s)]}{\sum_{h'(s)\in\mathcal{H}^N} \exp[\tau \cdot f((\mu(s), h'(s)); \theta_s)]}.$$

  Output $Q_s(\cdot\,;\bar{\omega}_s)$ using Algorithm 1 with the inputs policy $\Pi^\theta = \{\Pi_s^{\theta_s}\}_{s\in\mathcal{S}}$, 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.
  Sample $\{\mu_l, h_l\}_{l\in[B]}$ from the state–action visitation measure σ_θ (4.2) of Π^θ.
  for $s \in \mathcal{S}$ **do**
    Compute the local gradient estimator $\widehat{g}_s(\theta(t))$ using (4.14).
    Policy update: $\theta_s(t+1/2) \leftarrow \theta_s(t) + \eta_{\text{actor}} \cdot \widehat{g}_s(\theta(t))$
    Projection onto the parameter space: $\theta_s(t+1) \leftarrow \arg\min_{\theta\in\mathcal{B}_s^{\text{actor}}} \|\theta - \theta_s(t+1/2)\|_2$.
  end for
end for
Output: $\{\Pi^{\theta(t)}\}_{t\in[T_{\text{actor}}]}$.

5. Convergence of the Critic and Actor Updates

We now establish the global convergence for LTDE-Neural-AC proposed in Section 4. Our analysis of convergence relies on the use of an overparameterization technique, which involves a two-layer neural network with a large width M. This technique is critical to our analysis as it allows us to address the nonconvexity 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, a gradient-based method or other first order algorithms may be trapped at some undesired stationary points because of the nonconvex optimization landscape. Meanwhile, it is shown that the training problem in the overparameterization regime is almost equivalent to a regression problem in a reproducing kernel Hilbert space (RKHS) (Allen-Zhu et al. [3, 4], Cayci et al. [14], Zou and Gu [71]). In addition, the optimization landscape can also be improved by overparameterization in the sense that all stationary points are nearly optimal. These key properties of the overparameterized neural network facilitate our convergence analysis.

5.1. Convergence of the Critic Update

The convergence of the decentralized neural critic update in Algorithm 1 relies on the following assumptions.

Assumption 5.1 (Action-Value Function Class). *For each $s \in \mathcal{S}$, $k \in \mathbb{N}$, define*

$$\mathcal{F}_{R,\infty}^{s,k} = \left\{ f(\zeta_s^k) = Q_s(\zeta_s^k; \omega_s(0)) + \int \mathbb{1}\{v^\top \zeta_s^k > 0\} \cdot (\zeta_s^k)^\top \iota(v)\, d\mu(v) : \|\iota(v)\|_\infty \le R \right\}, \tag{5.1}$$

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 $Q_s(\zeta_s^k; \omega_s(0))$ 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 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 $\alpha \geq 0$, 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*

$$\mathbb{P}_{\zeta\sim\nu_\theta}(|v^\top\zeta| \leq \alpha) \leq c_0 \cdot \alpha, \quad \mathbb{P}_{\zeta\sim\sigma_\theta}(|v^\top\zeta| \leq \alpha) \leq c_0 \cdot \alpha.$$

Remark 5.1. Both Assumptions 5.1 and 5.2 are similar to the standard assumptions in the analysis of single-agent neural actor–critic algorithms (Cai et al. [7], Cayci et al. [14], Liu et al. [41], Wang et al. [57]).

In particular, Assumption 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 RKHS induced by the random feature $\mathbb{1}\{v^\top\zeta_s^k > 0\}\cdot(\zeta_s^k)$ with $v \sim N(0, I_{d_{\zeta_s^k}}/d_{\zeta_s^k})$ up to the shift of $Q_s(\zeta_s^k;\omega_s(0))$ (Rahimi and Recht [50]). This RKHS is dense in the space of continuous functions on any compact set (Ji et al. [32], Micchelli et al. [44]). (See also Section D.1.1 for details of the connection between $\mathcal{F}_{R,\infty}^{s,k}$ and the linearizations of two-layer neural networks (D.4)).

Assumption 5.2 holds when σ_θ and ν_θ have uniformly upper bounded probability densities (Cai et al. [7]).

Theorem 5.1 (Convergence of Critic Update). *Assume Assumptions 5.1 and 5.2. Set $T_{\text{critic}} = \Omega(M)$ and $\eta_{\text{critic}} = \min\{(1-\gamma)/8, (T_{\text{critic}})^{-1/2}\}$ in Algorithm 1. Then, $Q_s(\cdot;\bar{\omega}_s)$ generated by Algorithm 1 satisfies*

$$\mathbb{E}_{\text{init}}[\|Q_s(\cdot;\bar{\omega}_s) - Q_s^{\Pi^\theta}(\cdot)\|_{L^2(\nu_\theta)}^2] \leq \mathcal{O}\left(\frac{R^3 d_{\zeta_s^k}^{3/2}}{M^{1/2}} + \frac{R^{5/2} d_{\zeta_s^k}^{5/4}}{M^{1/4}} + \frac{r_{\max}^2\gamma^{k+1}}{(1-\gamma)^2}\right), \tag{5.2}$$

where $\|f\|_{L^2(\nu_\theta)} := (\mathbb{E}_{\zeta\sim\nu_\theta}[f(\zeta)^2])^{1/2}$, and the expectation (5.2) is taken with respect to the random initialization.

Theorem 5.1 indicates the trade-off between the approximation–optimization error and the localization error. The first two terms in (5.2) correspond to the neural network approximation–optimization error, similar to the single-agent case (Cai et al. [7], Cayci et al. [14]). This approximation–optimization error decreases when the width of the hidden layer M increases. Meanwhile, the last term in (5.2) represents the additional error from using the localized information in (4.11), unique for the mean-field MARL case. This localization error and γ^k decrease as the number of truncated neighborhoods 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 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 5.1 indicates the superior performance of the localized training scheme in efficiency over directly approximating the centralized Q-function.

Proof of Theorem 5.1 is presented in Section D.1.

5.2. 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 $\tilde{\theta}$; the second step controls the gap between the stationary point $\tilde{\theta}$ and the optimality θ^* in the overparameterization regime. The convergence is built under the following assumptions and definition.

Assumption 5.3 (Variance Upper Bound). *For every $t \in [T_{\text{actor}}]$ and $s \in \mathcal{S}$, denote $\xi_s(t) = \widehat{g}_s(\theta(t)) - \mathbb{E}[\widehat{g}_s(\theta(t))]$ with $\widehat{g}_s(\theta(t))$ defined in (4.14). Assume there exists $\Sigma > 0$ such that $\mathbb{E}[\|\xi_s(t)\|_2^2] \leq \tau^2\Sigma^2/B$. Here, the expectations are taken over $\sigma_{\theta(t)}$ given $\{\bar{\omega}_s\}_{s\in\mathcal{S}}$.*

Assumption 5.4 (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*

$$\{\mathbb{E}_{\nu_\theta}[(\mathrm{d}\sigma_\theta/\mathrm{d}\nu_\theta(\mu,h))^2]\} \leq D^2,$$

where $\mathrm{d}\sigma_\theta/\mathrm{d}\nu_\theta$ is the Radon–Nikodym derivative of σ_θ with respect to ν_θ.

Assumption 5.5 (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; that is, 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 5.1. $\tilde{\theta} \in \mathcal{B}^{\text{actor}}$ is called a stationary point of $J(\theta)$ if, for all $\theta \in \mathcal{B}^{\text{actor}}$,

$$\nabla_\theta J(\tilde{\theta})^\top(\theta - \tilde{\theta}) \leq 0. \tag{5.3}$$

Meanwhile, $\theta^* \in \mathcal{B}^{\text{actor}}$ is called an optimal point of $J(\theta)$ if

$$\theta^* \in \arg\max_{\theta \in \mathcal{B}^{\text{actor}}} J(\theta). \tag{5.4}$$

Assumption 5.6 (Policy Function Class). *Define a function class*

$$\mathcal{F}_{R,\infty} = \left\{ f(\zeta) = \sum_{s\in\mathcal{S}} \left[\phi_{\theta_s(0)}(\zeta_s)^\top \theta_s(0) + \int \mathbb{1}\{v^\top \zeta_s > 0\} \cdot (\zeta_s)^\top \iota(v)\, d\mu(v) \right] : \|\iota(v)\|_\infty \le R \right\},$$

where $\mu : \mathbb{R}^{d_{\zeta_s}} \to \mathbb{R}$ is the density function of the Gaussian distribution $N(0, I_{d_{\zeta_s}}/d_{\zeta_s})$ and $\theta(0)$ is the initial parameter. For any stationary point $\tilde{\theta}$, define the function

$$u_{\tilde{\theta}}(\mu, h) := \frac{d\sigma_{\theta^*}}{d\sigma_{\tilde{\theta}}}(\zeta) - \frac{d\bar{\sigma}_{\theta^*}}{d\bar{\sigma}_{\tilde{\theta}}}(\mu) + \sum_{s\in\mathcal{S}} \phi_{\tilde{\theta}_s}(\zeta_s)^\top \tilde{\theta}_s,$$

with $\bar{\sigma}_\theta$ the state visitation measure under policy Π^θ, and $\mathrm{d}\sigma_{\theta^*}/\mathrm{d}\sigma_{\tilde{\theta}}$, $\mathrm{d}\bar{\sigma}_{\theta^*}/\mathrm{d}\bar{\sigma}_{\tilde{\theta}}$ the Radon–Nikodym derivatives between corresponding measures. We assume that $u_{\tilde{\theta}} \in \mathcal{F}_{R,\infty}$ for any stationary point $\tilde{\theta}$.

A few remarks are in place for these Assumption 5.3–5.6.

Remark 5.2. All these assumptions are counterparts of standard assumption in the analysis of the single-agent policy gradient method (Pirotta et al. [46], Wang et al. [57], Xu et al. [58, 59], Zhang et al. [65]).

In particular, Assumptions 5.3 and 5.4 hold if the Markov chain (3.5) mixes sufficiently fast, and the critic $Q_s(\cdot;\omega_s)$ has an upper bounded second moment under $\sigma_{\theta(t)}$ (Wang et al. [57]). Note that different from Assumption 5.2, in which regularity conditions are imposed separately on ν_θ and σ_θ, Assumption 5.4 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 5.2. In general, Assumption 5.2 does not necessarily imply Assumption 5.4.

We also emphasize that Assumption 5.3 holds under mild conditions and can be justified by certain properties of the estimator $\widehat{g}_s$ in (4.14). More specifically, when the estimator $\widehat{g}_s$ in (4.14) can be viewed as an average of B independent and identically distributed (i.i.d.) samples

$$\left[\sum_{y\in\mathcal{N}_s^k} Q_y\left(\mu_l(\mathcal{N}_y^k), h_l(\mathcal{N}_y^k); \bar{\omega}_y\right) \right] \cdot \widehat{\Phi}(\theta(t), s, \mu_l, h_l), \quad l \in [B],$$

and Assumption 5.3 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. Therefore, when the parameters of the critic networks are uniformly bounded, Assumption 5.3 holds, and the dependency on the algorithm trajectory becomes less concerning.

Assumption 5.5 holds when the transition probability and the reward function are both Lipschitz continuous with respect to their inputs (Pirotta et al. [46]) 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 (Zhang et al. [65]).

As for Assumption 5.6, we first emphasize that $u_{\tilde{\theta}}(\mu, h)$ is a key element in the proof of Theorem 5.2. More specifically, this assumption is motivated by the well-known performance difference lemma (Kakade and Langford [35]) in order to characterize the optimality gap of a stationary point $\tilde{\theta}$. In particular, it guarantees that $u_{\tilde{\theta}}$ 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 5.1). Appendix E provides a concrete network example that satisfies all Assumptions 5.1–5.6 (or their mild relaxations).

With all these assumptions, we now establish the rate of convergence for Algorithm 2.

Theorem 5.2. *Assume Assumptions 5.1–5.6. Set $T_{\text{critic}} = \Omega(M)$, $\eta_{\text{critic}} = \min\{(1-\gamma)/8, (T_{\text{critic}})^{-1/2}\}$, $\eta_{\text{actor}} = (T_{\text{actor}})^{-1/2}$, $R = \tau = 1$, $M = \Omega((f(k)|\mathcal{A}|)^5 (T_{\text{actor}})^8)$, $\gamma \le (T_{\text{actor}})^{-2/k}$ 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 2 satisfies*

$$\min_{t\in[T_{\text{actor}}]} \mathbb{E}[J(\theta^*) - J(\theta(t))] \le \mathcal{O}(|\mathcal{S}|^{1/2} B^{-1/2} + |\mathcal{S}||\mathcal{A}|^{1/4} (\gamma^{k/8} + (T_{\text{actor}})^{-1/4})). \tag{5.5}$$

Note that the error $\mathcal{O}(\gamma^{k/8}|\mathcal{S}||\mathcal{A}|^{1/4})$ in Theorem 5.2, coming from the localized training, decays exponentially quickly as k increases and is negligible with a careful choice of k. According to Theorem 5.2, Algorithm 2 converges at rate $T_{\text{actor}}^{-1/4}$ with sufficiently large width M and batch size B.

Indeed, Theorem 5.2 manages to incorporate the neural network optimization error, which is analyzed in Cai et al. [7] and Wang et al. [57] 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 Cai et al. [7] and Wang et al. [57]. A detailed proof of Theorem 5.2 is provided in Section D.2.

Remark 5.3 (Convergence to Optimal Decentralized Neural Policy). By Definition 5.1, 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) and a subset of all possible decentralized policies. The convergence in Theorem 5.2 relies on the neural network parameterization and may not necessarily imply the convergence under a different policy class.

Remark 5.4 (Choice of k). The particular form $\gamma < (T_{\text{actor}})^{-2/k}$ in Theorem 5.2 is not essential and is mainly chosen to highlight the error bound in (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 5.2 (see (D.42) in Appendix D.2).

Remark 5.5 (Total Sample Complexity). The sample complexity T_{actor} is of the order $\mathcal{O}(\epsilon^{-4})$, which, in turn, leads to the width of the neural network and the sample complexity for the critic T_{critic} being of the order $\mathcal{O}(\epsilon^{-32})$. As a result, the total sample complexity becomes $T_{\text{critic}} \times T_{\text{actor}} = \mathcal{O}(\epsilon^{-36})$. Note that this sample complexity $\mathcal{O}(\epsilon^{-36})$ is of the same order as that in Wang et al. [57, theorem 4.7] for single-agent reinforcement learning. In fact, the key reason for such complexity is because of the adoption of the overparameterization technique. Even in supervised learning settings, large network width is often needed for achieving desirable generalization error guarantees (Allen-Zhu et al. [3, 4], Zou and Gu [71]), resulting in large sample complexities similar to our result.

Acknowledgment

The authors express their gratitude to the area editor, the associate editor, and three anonymous reviewers for their insightful comments, which significantly contributed to the improvement of our paper.

Appendix A. Proof of Lemma 3.1

The goal is to show that $V(\mu) = \tilde{V}(\mu)$ with the former the value function of (MF-MARL) subject to the transition probability P defined in (2.7) under a given individual policy $\pi \in \mathfrak{u}$ and the latter the value function of (3.7) subject to the joint transition probability $\mathbf{P}^N$ defined in (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 measure-valued Markov decision problem. Step 2 shows that the measure-valued Markov decision problem from step 1 is equivalent to $\tilde{V}(\mu)$ in (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 (2.7). First, one can show that μ_t is a measure-valued Markov decision process under π. To see this, denote $\mathcal{F}_t^s = \sigma(s_t^1, \ldots, s_t^N)$ as the σ-algebra generated by $s_t^1, \ldots, s_t^N$. Then, it suffices to show

$$\mathbb{P}(\mu_{t+1}|\sigma(\mu_t) \vee \mathcal{F}_t^s) = \mathbb{P}(\mu_{t+1}|\sigma(\mu_t)),\ \mathbb{P}-a.s.. \tag{A.1}$$

Following similar arguments for Dawson [16, lemma 2.3.1 and proposition 2.3.3], (A.1) holds because of the exchangeability of the individual transition dynamics (2.7) under π. Equation (A.1) implies that there exists a joint transition probability induced from (2.7) under π, denoted as $\tilde{\mathbf{P}}^N$ such that

$$\mu_{t+1} \sim \tilde{\mathbf{P}}^N(\cdot|\mu_t, \pi). \tag{A.2}$$

Meanwhile, rewrite $V^\pi(\mu)$ in (MF-MARL) by regrouping the agents according to their states

$$\begin{aligned} V^\pi(\mu) &:= \mathbb{E}\left[\sum_{t=0}^\infty \gamma^t \sum_{i=1}^N \frac{1}{N} r(s_t^i, \mu_t(\mathcal{N}_{s_t^i}), a_t^i) \middle| \mu_0 = \mu\right], \\ &= \mathbb{E}\left[\sum_{t=0}^\infty \gamma^t \sum_{s\in\mathcal{S}} \mu_t(s) \sum_{a\in\mathcal{A}} r(s, \mu_t(\mathcal{N}_s), a)\pi(s, \mu_t(s))(a) \middle| \mu_0 = \mu\right]. \end{aligned} \tag{A.3}$$

We see that (2.7) and (MF-MARL) is reformulated in an equivalent form of (A.2) and (A.3).

Step 2: It suffices to show that (A.2) under π is the same as (3.5) under Π and that V^π in (A.3) equals to $\tilde{V}^\Pi$ in (3.7). To see this, denote $\langle g, \mu\rangle = \sum_{s\in\mathcal{S}} g(s)\mu(s)$ for any measurable bounded function $g : \mathcal{S} \to \mathbb{R}$, and then

$$\begin{aligned}
&\mathbb{E}[\langle g, \mu_{t+1}\rangle \,|\sigma(\mu_t)] \\
&= \frac{1}{N}\mathbb{E}\left[\sum_{i=1}^{N}\mathbb{E}[g(s_{t+1}^j)|\sigma(\mu_t)\vee\mathcal{F}_t^s]\right] \\
&= \frac{1}{N}\sum_{s'\in\mathcal{S}}\sum_{i=1}^{N}\sum_{a\in\mathcal{A}} g(s')P(s'\,|s_t^i,\mu_t(\mathcal{N}(s_t^i)),a)\pi(s_t^i,\mu_t(s_t^i))(a) \\
&= \frac{1}{N}\sum_{s'\in\mathcal{S}} g(s')\sum_{s\in\mathcal{S}}\sum_{i=1}^{N}\mathbb{1}(s_t^i=s)\sum_{a\in\mathcal{A}} P(s'\,|s_t^i,\mu_t(\mathcal{N}(s_t^i)),a)\pi(s_t^i,\mu_t(s_t^i))(a) \\
&= \sum_{s'\in\mathcal{S}} g(s')\sum_{s\in\mathcal{S}}\mu_t(s)\sum_{a\in\mathcal{A}} P(s'\,|s,\mu_t(\mathcal{N}(s)),a)\pi(s,\mu_t(s))(a) \\
&= \sum_{s'\in\mathcal{S}} g(s')\sum_{s\in\mathcal{S}}\mu_t(s)\sum_{h\in\mathcal{P}^{\mathcal{N}_s,\mu_t(s)}(\mathcal{A})}\Pi(h|\mu_t(s))\sum_{a\in\mathcal{A}} P(s'\,|s,\mu_t(\mathcal{N}(s)),a)h(s)(a),
\end{aligned} \tag{A.4}$$

where in the last step, the expectation of random variable $h(s)(a)$ with respect to distribution $\Pi(h|\mu)$ 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 $\Pi(h_t|\mu_t)$. This implies the equivalence of (A.2) and (3.5). As a byproduct, when taking $g(s') = \mathbb{1}(s'=s^o)$ for any fixed $s^o\in\mathcal{S}$, (A.4) becomes

$$\mathbb{E}[\mu_{t+1}(s^o)|\sigma(\mu_t)] = \sum_{s\in\mathcal{N}(s^o)}\mu_t(s)\sum_{h\in\mathcal{P}^{\mathcal{N}_s,\mu_t(s)}(\mathcal{A})}\Pi(h|\mu_t(s))\sum_{a\in\mathcal{A}} P(s^o|s,\mu_t(\mathcal{N}(s)),a)h(s)(a),$$

where the local structure (2.7) is used. This suggests that $\mu_{t+1}(s^o)$ only depends on $\mu_t(\mathcal{N}_{s^o}^2)$ and $h_t(\mathcal{N}_{s^o})$ because $\mathcal{N}(s) = \mathcal{N}^2(s^o)$ for $s\in\mathcal{N}(s^o)$.

Now, we show that $V^\pi(\mu)$ in (A.3) and $\tilde{V}^\Pi(\mu)$ in (3.7) are equal. Take $\tilde{V}^\Pi$ defined in (3.7),

$$\begin{aligned}
\tilde{V}^\Pi(\mu) &:= \mathbb{E}_{h_t\sim\Pi(\cdot\,|\,\mu_t),\,\mu_{t+1}\sim\mathbf{P}^N(\cdot\,|\,\mu_t,h_t)}\left[\sum_{t=0}^{\infty}\sum_{s\in\mathcal{S}}\gamma^t r_s(\mu_t(\mathcal{N}_s),h_t)\middle|\mu_0=\mu\right] \\
&= \mathbb{E}_{\mu_{t+1}\sim\mathbf{P}^N(\cdot\,|\,\mu_t,h_t)}\left[\sum_{t=0}^{\infty}\gamma^t\sum_{s\in\mathcal{S}}\mathbb{E}_{h_t\sim\Pi(\cdot\,|\,\mu_t)}[r_s(\mu_t(\mathcal{N}_s),h_t)|\mu_t]\middle|\mu_0=\mu\right] \\
&= \mathbb{E}_{\mu_{t+1}\sim\mathbf{P}^N(\cdot\,|\,\mu_t,h_t)}\left[\sum_{t=0}^{\infty}\gamma^t\sum_{s\in\mathcal{S}}\sum_{h_t\in\mathcal{P}^{\mathcal{N}_s,\mu_t(s)}(\mathcal{A})} r_s(\mu_t(\mathcal{N}_s),h_t(s))\Pi(h;\pi)\middle|\mu_0=\mu\right] \\
&= \mathbb{E}_{\mu_{t+1}\sim\mathbf{P}^N(\cdot\,|\,\mu_t,h_t)}\left[\sum_{t=0}^{\infty}\gamma^t\sum_{s\in\mathcal{S}}\mu_t(s)\sum_{h_t\in\mathcal{P}^{\mathcal{N}_s,\mu_t(s)}(\mathcal{A})}\Pi(h_t|\mu_t)\sum_{a\in\mathcal{A}} r(s,\mu_t(\mathcal{N}_s),a)h(a)\middle|\mu_0=\mu\right] \\
&= \mathbb{E}_{\mu_{t+1}\sim\tilde{\mathbf{P}}^N(\cdot\,|\,\mu_t,\pi)}\left[\sum_{t=0}^{\infty}\gamma^t\sum_{s\in\mathcal{S}}\mu_t(s)\sum_{a\in\mathcal{A}} r(s,\mu_t(\mathcal{N}_s),a)\pi_t(s,\mu_t(s))(a)\middle|\mu_0=\mu\right] \\
&= V^\pi(\mu),
\end{aligned}$$

where in the last second step, $\mathbf{P}^N$ under π is equivalent to $\tilde{\mathbf{P}}^N$ under Π, and the expectation of $h_t(s)(a)$ with distribution $\Pi(h_t|\mu_t)$ is $\pi(s,\mu_t(s))(a)$ such that

$$\begin{aligned}
\sum_{h\in\mathcal{P}^{\mathcal{N}_s,\mu_t(s)}(\mathcal{A})}\Pi(h_t|\mu_t)\sum_{a\in\mathcal{A}} r(s,\mu_t(\mathcal{N}_s),a)h(a) &= \mathbb{E}_{h\sim\Pi(\cdot\,|\,\mu_t)}\left[\sum_{a\in\mathcal{A}} r(s,\mu_t(\mathcal{N}_s,a)h(a)\right] \\
&= \sum_{a\in\mathcal{A}} r(s,\mu_t(\mathcal{N}_s),a)\pi_t(s,\mu_t(s))(a).
\end{aligned}$$

Finally, the decomposition of $\tilde{V}(\mu)$ and $\tilde{V}(\mu)$ according to the states is straightforward. Q.E.D.

Appendix B. Proof of Lemma 3.4

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 the localized transition kernel (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, (3.5) can be rewritten as

$$\mu_{t+1}(s) \sim \mathbf{P}_s^N(\cdot \mid \mu_t(\mathcal{N}_s^2), h_t(\mathcal{N}_s)). \tag{B.1}$$

Because of the local structure of dynamics (B.1) and local dependence of Π^θ, the distribution $\mathfrak{P}_{t,s}$, $t \le \lfloor \frac{k}{2} \rfloor$ only depends on the initial value $(\mu(\mathcal{N}_s^k), h(\mathcal{N}_s^k))$. Therefore, $\mathfrak{P}_{t,s} = \mathfrak{P}'_{t,s}$, $t \le \lfloor \frac{k}{2} \rfloor$,

$$\begin{aligned}
&\left| Q_s^{\Pi^\theta}(\mu(\mathcal{N}_s^k), \mu(\mathcal{N}_s^{-k}), h(\mathcal{N}_s^k), h(\mathcal{N}_s^{-k})) - Q_s^{\Pi^\theta}(\mu(\mathcal{N}_s^k), \mu'(\mathcal{N}_s^{-k}), h(\mathcal{N}_s^k), h'(\mathcal{N}_s^{-k})) \right| \\
&= \sum_{t=\lfloor \frac{k}{2} \rfloor + 1}^{\infty} \mathbb{E}_{(\mu_t(\mathcal{N}_s), h_t(s)) \sim \mathfrak{P}_{t,s}}[r_s(\mu_t(\mathcal{N}_s), h_t(s))] - \mathbb{E}_{(\mu'_t(\mathcal{N}_s), h'_t(s)) \sim \mathfrak{P}'_{t,s}}[r_s(\mu'_t(\mathcal{N}_s), h'_t(s))] \\
&\le \sum_{t=\lfloor \frac{k}{2} \rfloor + 1}^{\infty} \gamma^t r_{\max} \mathrm{TV}(\mathfrak{P}_{t,s}, \mathfrak{P}'_{t,s}) \le \frac{r_{\max}}{1-\gamma} \gamma^{\lfloor \frac{k}{2} \rfloor + 1},
\end{aligned}$$

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 one. Q.E.D.

Appendix C. Proof of Lemma 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 $\|\Phi(\theta, s, \mu, h)\|_2 \le \|\zeta_s\|_2 \le 2$, by the definitions of the feature mapping ϕ in (4.6) and the center feature mapping Φ in (4.7).

To prove (4.8), note that, by Lemma 4.1 and the definition of energy-based policy $\Pi_s^{\theta_s}$ (4.4),

$$\begin{aligned}
\nabla_{\theta_s} \log \Pi_s^{\theta_s}(h(s) \mid \mu(s)) &= \tau \cdot \nabla_{\theta_s} f((\mu(s), h(s)); \theta_s) - \tau \cdot \mathbb{E}_{h(s)' \sim \Pi^{\theta_s}(\cdot \mid \mu(s))}[\nabla_{\theta_s} f(\mu(s), h'(s))] \\
&= \tau \cdot \phi_{\theta_s}(\mu(s), h(s)) - \tau \cdot \mathbb{E}_{h(s)' \sim \Pi^{\theta_s}(\cdot \mid \mu(s))}[\phi_{\theta_s}(\mu(s), h(s))] \\
&= \tau \cdot \Phi(\theta, s, \mu, h).
\end{aligned}$$

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,

$$\nabla_{\theta_s} J(\theta) = \frac{\tau}{1-\gamma} \mathbb{E}_{\sigma_\theta}[Q^{\Pi^\theta}(\mu, h) \cdot \Phi(\theta, s, \mu, h)] = \frac{\tau}{1-\gamma} \mathbb{E}_{\sigma_\theta}\left[\sum_{y \in \mathcal{S}} Q_y^{\Pi^\theta}(\mu, h) \cdot \Phi(\theta, s, \mu, h) \right],$$

where the second equality is by the decomposition of the Q-function in Lemma 3.1.

The proof of (4.9) is based on the exponential decay property in Definition 3.1. Notice that

$$\begin{aligned}
g_s(\theta) &= \frac{1}{1-\gamma} \mathbb{E}_{\sigma_\theta}\left[\left[\sum_{y \in \mathcal{N}_s^k} \widehat{Q}_y^{\Pi^\theta}(\mu(\mathcal{N}_y^k), h(\mathcal{N}_y^k)) \right] \nabla_{\theta_s} \log \Pi^{\theta_s}(h(s) \mid \mu(s)) \right] \\
&= \frac{1}{1-\gamma} \mathbb{E}_{\sigma_\theta}\left[\left[\sum_{y \in \mathcal{S}} \widehat{Q}_y^{\Pi^\theta}(\mu(\mathcal{N}_y^k), h(\mathcal{N}_y^k)) \right] \nabla_{\theta_s} \log \Pi^{\theta_s}(h(s) \mid \mu(s)) \right].
\end{aligned} \tag{C.1}$$

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 4.1 and (C.1), we have the following bound:

$$\begin{aligned}&\|g_s(\theta)-\nabla_{\theta_s}J(\theta)\|_2\\&\le\frac{1}{1-\gamma}\sum_{y\in\mathcal{S}}\sup_{\substack{\mu\in\mathcal{P}^N(\mathcal{S}),\\ h\in\mathcal{H}^N(\mu)}}\left[|\widehat{Q}_y^{\Pi^\theta}(\mu(\mathcal{N}_y^k),h(\mathcal{N}_y^k))-Q_y^{\Pi^\theta}(\mu,h)|\cdot\|\nabla_{\theta_s}\log\Pi^{\theta_s}(h(s)|\mu(s))\|_2\right]\\&\le\frac{c_0\tau|\mathcal{S}|}{1-\gamma}\rho^{k+1}.\end{aligned}$$

The last inequality follows from (3.14) and $\|\log\Pi^{\theta_s}(h(s)|\mu(s))\|_2=\|\Phi(\theta,s,\mu,h)\|_2\le 2$ for any $\mu\in\mathcal{P}^N(\mathcal{S}),h\in\mathcal{H}^N(\mu)$. Q.E.D.

Appendix D. Proof of Theorems 5.1 and 5.2

D.1. Proof of Theorem 5.1: Convergence of Critic Update

This section presents the proof of convergence of the decentralized neural critic update. It consists of several steps. Section D.1.1 introduces necessary notations and definitions. Section D.1.2 proves that the critic update minimizes the projected mean-square Bellman error given a two-layer neural network. Section D.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 $M\to\infty$.

D.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 $\|\zeta_s^k\|_2\le 1$ for any $\zeta_s^k\in\Xi_s^k$.

Let d_ζ denote the dimension of the space Ξ. Because $\mathcal{P}^N(\mathcal{S})$ 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

$$\|f\|_{L^2(\nu_\theta)}:=(\mathbb{E}_{\zeta\sim\nu_\theta}[f(\zeta)^2])^{1/2},\quad\forall f\in\mathbb{R}^\Xi.\tag{D.1}$$

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 $\zeta\in\Xi$. Because 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 $\mathrm{Proj}_{\mathcal{F}}$ from $\mathbb{R}^\Xi$ onto $\mathcal{F}$ by

$$\mathrm{Proj}_{\mathcal{F}}(g):=\arg\min_{f\in\mathcal{F}}\|f-g\|_{L^2(\nu_\theta)}.\tag{D.2}$$

This projection operator $\mathrm{Proj}_{\mathcal{F}}$ is nonexpansive in the sense that

$$\|\mathrm{Proj}_{\mathcal{F}}(f)-\mathrm{Proj}_{\mathcal{F}}(g)\|_{L^2(\nu_\theta)}\le\|f-g\|_{L^2(\nu_\theta)}.\tag{D.3}$$

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:

$$\mathcal{F}_{R,M}=\left\{Q_0(\zeta_s^k;\omega):=\frac{1}{\sqrt{M}}\sum_{m=1}^M\mathbb{1}\{[\omega(0)]_m^\top\zeta_s^k>0\}\omega_m^\top\zeta_s^k:\omega\in\mathbb{R}^{M\times d_{\zeta_s^k}},\|\omega-\omega(0)\|_\infty\le R/\sqrt{M}\right\}.\tag{D.4}$$

$Q_0(\cdot;\omega)$ 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 (4.6) and the parameter ω, that is, $Q_0(\cdot;\omega)=\phi_{\omega(0)}(\cdot)^\top\omega$. 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 (3.13),

$$\mathcal{T}_s^\theta Q_s^{\Pi^\theta}(\mu,h)=\mathbb{E}_{\mu'\sim\mathbf{P}^N(\cdot\,|\,\mu,h),\,h'\sim\Pi^\theta(\cdot\,|\,\mu)}[r_s(\mu,h)+\gamma\cdot Q_s^{\Pi^\theta}(\mu',h')],\ \forall(\mu,h)\in\Xi.$$

The team-decentralized Q-function $Q_s^{\Pi^\theta}$ in (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):

$$\min_{\omega} \text{PMSBE}(\omega) = \mathbb{E}_{\zeta\sim\nu_\theta}[(Q_s(\zeta_s^k;\omega) - \text{Proj}_{\mathcal{F}}\, \mathcal{T}_s^\theta Q_s(\zeta_s^k;\omega))^2]. \tag{D.5}$$

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 (D.4)) and consider the following PMSBE:

$$\min_{\omega} \mathbb{E}_{\zeta\sim\nu_\theta}[(Q_0(\zeta_s^k;\omega) - \text{Proj}_{\mathcal{F}_{R,M}}\, \mathcal{T}_s^\theta Q_0(\zeta_s^k;\omega))^2]. \tag{D.6}$$

We show in Section D.1.2 that the output of Algorithm 1 converges to the global minimizer of (D.6).

D.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 (D.6).

Lemma D.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 a 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 (D.6).*

Proof of Lemma D.1. We first show that the operator $\mathcal{T}_s^\theta : \mathbb{R}^\Xi \to \mathbb{R}^\Xi$ (3.13) is a γ-contraction in the $L^2(\nu_\theta)$-norm:

$$\begin{aligned}
&\|\mathcal{T}_s^\theta Q_1 - \mathcal{T}_s^\theta Q_2\|_{L^2(\nu_\theta)}^2 = \mathbb{E}_{\zeta\sim\nu_\theta}[(\mathcal{T}_s^\theta Q_1(\zeta) - \mathcal{T}_s^\theta Q_2(\zeta))^2] \\
&= \gamma^2 \mathbb{E}_{\zeta\sim\nu_\theta}[(\mathbb{E}[Q_1(\zeta') - Q_2(\zeta')|\zeta' = (\mu', h'), \mu' \sim P^N(\cdot|\zeta), h' \sim \Pi^\theta(\cdot|\mu')])^2] \\
&\le \gamma^2 \mathbb{E}_{\zeta\sim\nu_\theta}[\mathbb{E}[(Q_1(\zeta') - Q_2(\zeta'))^2|\zeta' = (\mu', h'), \mu' \sim P^N(\cdot|\zeta), h' \sim \Pi^\theta(\cdot|\mu')]] \\
&= \gamma^2 \mathbb{E}_{\zeta'\sim\nu_\theta}[(Q_1(\zeta') - Q_2(\zeta'))^2] = \gamma^2 \|Q_1 - Q_2\|_{L^2(\nu_\theta)}^2,
\end{aligned}$$

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 $\text{Proj}_{\mathcal{F}_{R,M}} : \mathbb{R}^\Xi \to \mathcal{F}_{R,M}$ is nonexpansive. Therefore, the operator $\text{Proj}_{\mathcal{F}_{R,M}} \mathcal{T}_s^\theta : \mathcal{F}_{R,M} \to \mathcal{F}_{R,M}$ is a γ-contraction in the $L^2(\nu_\theta)$-norm. Hence, $\text{Proj}_{\mathcal{F}_{R,M}}$ 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 (D.6). Q.E.D.

We show that the function class $\mathcal{F}_{R,M}$ approximately becomes $\mathcal{F}_{R,\infty}^{s,k}$ (defined in Assumption 5.1) as $M \to \infty$, where $\mathcal{F}_{R,\infty}^{s,k}$ is a rich RKHS. Consequently, $Q_0(\cdot;\omega^*)$ becomes the global minimum of the MSPBE (D.6) on $\mathcal{F}_{R,\infty}^{s,k}$ given Lemma D.1. Moreover, by using similar argument and technique developed in Cai et al. [7, theorem 4.6], we can establish the convergence of Algorithm 1 to $Q_0(\cdot;\omega^*)$ as the following.

Theorem D.1 (Convergence to $Q_0(\cdot;\omega^*)$). *Set $\eta_{\text{critic}} = \min\{(1-\gamma)/8, 1/\sqrt{T_{\text{critic}}}\}$ in Algorithm 1. Then, the output $Q_s(\cdot;\bar{\omega})$ of Algorithm 1 satisfies*

$$\mathbb{E}_{\text{init}}[\|Q_s(\cdot;\bar{\omega}) - Q_0(\cdot;\omega^*)\|_{L^2(\nu_\theta)}^2] \le \mathcal{O}\left(\frac{R^3 d_{\zeta_s^k}^{3/2}}{\sqrt{M}} + \frac{R^{5/2} d_{\zeta_s^k}^{5/4}}{\sqrt[4]{M}} + \frac{R^2 d_{\zeta_s^k}}{\sqrt{T_{\text{critic}}}}\right),$$

where the expectation is taken with respect to the random initialization.

The proof of Theorem D.1 is straightforward from Cai et al. [7, theorem 4.6] and, hence, omitted.

D.1.3. Convergence to $Q_s^{\Pi^\theta}$. Next, we analyze the error between the global minimizer of (D.6) and the team-decentralized Q-function $Q_s^{\Pi^\theta}$ (defined in (3.10)) to complete the convergence analysis. Different from the single-agent case as in Cai et al. [7], 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 5.1. First, recall that, by Lemma 3.4, $Q_s^{\Pi^\theta}$ satisfies the (c,ρ)-exponential decay property in Definition 3.1 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), and then,

$$|Q_s^{\Pi^\theta}(\zeta) - \widehat{Q}_s^{\Pi^\theta}(\zeta_s^k)| \le c\rho^{k+1}, \quad \forall \zeta \in \Xi. \tag{D.7}$$

By the triangle inequality and $(a+b)^2 \le 2(a^2+b^2)$,

$$\begin{aligned}
\|Q_s(\cdot;\bar{\omega}) - Q_s^{\Pi^\theta}(\cdot)\|_{L^2(\nu_\theta)}^2 &\le (\|Q_s(\cdot;\bar{\omega}) - Q_0(\cdot;\omega^*)\|_{L^2(\nu_\theta)} + \|Q_s^{\Pi^\theta}(\cdot) - Q_0(\cdot;\omega^*)\|_{L^2(\nu_\theta)})^2 \\
&\le 2(\|Q_s(\cdot;\bar{\omega}) - Q_0(\cdot;\omega^*)\|_{L^2(\nu_\theta)}^2 + \|Q_s^{\Pi^\theta}(\cdot) - Q_0(\cdot;\omega^*)\|_{L^2(\nu_\theta)}^2).
\end{aligned} \tag{D.8}$$

The first term in (D.8) is studied in Theorem D.1, and it suffices to bound the second term. By interpolating two intermediate terms $\widehat{Q}_s^{\Pi^\theta}$ and $\text{Proj}_{\mathcal{F}_{R,M}}\widehat{Q}_s^{\Pi^\theta}$, we have

$$\|Q_s^{\Pi^\theta}(\cdot) - Q_0(\cdot;\omega^*)\|_{L^2(\nu_\theta)} \le \underbrace{\|Q_s^{\Pi^\theta}(\cdot) - \widehat{Q}_s^{\Pi^\theta}(\cdot)\|_{L^2(\nu_\theta)}}_{\text{(I)}} + \underbrace{\|\widehat{Q}_s^{\Pi^\theta}(\cdot) - \text{Proj}_{\mathcal{F}_{R,M}}\widehat{Q}_s^{\Pi^\theta}(\cdot)\|_{L^2(\nu_\theta)}}_{\text{(II)}}$$
$$+ \underbrace{\|Q_0(\cdot;\omega^*) - \text{Proj}_{\mathcal{F}_{R,M}}\widehat{Q}_s^{\Pi^\theta}(\cdot)\|_{L^2(\nu_\theta)}}_{\text{(III)}}. \tag{D.9}$$

First, we have (I) $\le c\rho^{k+1}$ according to (D.7). To bound (III), we have

$$\begin{aligned}
\text{(III)} &= \|\text{Proj}_{\mathcal{F}_{R,M}}\mathcal{T}_s^\theta Q_0(\cdot;\omega^*) - \text{Proj}_{\mathcal{F}_{R,M}}\widehat{Q}_s^{\Pi^\theta}(\cdot)\|_{L^2(\nu_\theta)} \\
&\le \|\text{Proj}_{\mathcal{F}_{R,M}}\mathcal{T}_s^\theta Q_0(\cdot;\omega^*) - \text{Proj}_{\mathcal{F}_{R,M}}\mathcal{T}_s^\theta Q_s^{\Pi^\theta}(\cdot)\|_{L^2(\nu_\theta)} + \|\text{Proj}_{\mathcal{F}_{R,M}}\mathcal{T}_s^\theta Q_s^{\Pi^\theta}(\cdot) - \text{Proj}_{\mathcal{F}_{R,M}}\widehat{Q}_s^{\Pi^\theta}(\cdot)\|_{L^2(\nu_\theta)} \\
&\le \gamma\|Q_0(\cdot;\omega^*) - Q_s^{\Pi^\theta}(\cdot)\|_{L^2(\nu_\theta)} + \|\mathcal{T}_s^\theta Q_s^{\Pi^\theta}(\cdot) - \widehat{Q}_s^{\Pi^\theta}(\cdot)\|_{L^2(\nu_\theta)} \\
&= \gamma\|Q_0(\cdot;\omega^*) - Q_s^{\Pi^\theta}(\cdot)\|_{L^2(\nu_\theta)} + \underbrace{\|Q_s^{\Pi^\theta}(\cdot) - \widehat{Q}_s^{\Pi^\theta}(\cdot)\|_{L^2(\nu_\theta)}}_{\text{(I)}} \\
&\le \gamma\|Q_0(\cdot;\omega^*) - Q_s^{\Pi^\theta}(\cdot)\|_{L^2(\nu_\theta)} + c\rho^{k+1}.
\end{aligned} \tag{D.10}$$

The first line in (D.10) is because $Q_0(\cdot;\omega^*)$ is the unique fixed point of the operator $\text{Proj}_{\mathcal{F}_{R,M}}\mathcal{T}_s^\theta$ (as proved in Lemma D.1); the third line in (D.10) is because the operator $\text{Proj}_{\mathcal{F}_{R,M}}\mathcal{T}_s^\theta$ is a γ-contraction in the $L^2(\nu_\theta)$ norm, and $\text{Proj}_{\mathcal{F}_{R,M}}$ is nonexpansive; the fourth line in (D.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 (D.10) with (D.9) and the bound on (I) gives us

$$\|Q_s^{\Pi^\theta}(\cdot) - Q_0(\cdot;\omega^*)\|_{L^2(\nu_\theta)} \le \frac{1}{1-\gamma}\left(2c\rho^{k+1} + \underbrace{\|\widehat{Q}_s^{\Pi^\theta}(\cdot) - \text{Proj}_{\mathcal{F}_{R,M}}\widehat{Q}_s^{\Pi^\theta}(\cdot)\|_{L^2(\nu_\theta)}}_{\text{(II)}}\right),$$

and consequently,

$$\|Q_s^{\Pi^\theta}(\cdot) - Q_0(\cdot;\omega^*)\|_{L^2(\nu_\theta)}^2 \le \frac{1}{(1-\gamma)^2}\left(8c^2\rho^{2k+2} + 2\underbrace{\|\widehat{Q}_s^{\Pi^\theta}(\cdot) - \text{Proj}_{\mathcal{F}_{R,M}}\widehat{Q}_s^{\Pi^\theta}(\cdot)\|_{L^2(\nu_\theta)}^2}_{\text{(II)}}\right). \tag{D.11}$$

Plugging (D.11) into (D.8) yields

$$\begin{aligned}
&\mathbb{E}_{\text{init}}[\|Q_s(\cdot;\bar{\omega}) - Q_s^{\Pi^\theta}(\cdot)\|_{L^2(\nu_\theta)}^2] \\
&\le 2\left(\mathbb{E}_{\text{init}}[\|Q_s(\cdot;\bar{\omega}) - Q_0(\cdot;\omega^*)\|_{L^2(\nu_\theta)}^2] + \mathbb{E}_{\text{init}}[\|Q_s^{\Pi^\theta}(\cdot) - Q_0(\cdot;\omega^*)\|_{L^2(\nu_\theta)}^2]\right) \\
&\le \mathcal{O}\left(\frac{R^3 d_{\zeta_s^k}^{3/2}}{\sqrt{M}} + \frac{R^{5/2} d_{\zeta_s^k}^{5/4}}{\sqrt[4]{M}} + \frac{R^2 d_{\zeta_s^k}}{\sqrt{T}} + c^2\rho^{2k+2}\right) + \frac{4}{(1-\gamma)^2}\mathbb{E}_{\text{init}}\left[\underbrace{\|\widehat{Q}_s^{\Pi^\theta}(\cdot) - \text{Proj}_{\mathcal{F}_{R,M}}\widehat{Q}_s^{\Pi^\theta}(\cdot)\|_{L^2(\nu_\theta)}^2}_{\text{(II)}}\right].
\end{aligned} \tag{D.12}$$

Term (II) measures the distance between $\widehat{Q}_s^{\Pi^\theta}$ and the class $\mathcal{F}_{R,M}$. As discussed in Section D.1.1, the function class $\mathcal{F}_{R,M}$ converges to $\mathcal{F}_{R,\infty}^{s,k}$ (defined in Assumption 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 Rahimi and Recht [50] and Cai et al. [7, proposition 4.3]:

Lemma D.2. *Assume Assumption 5.1, and we have*

$$\mathbb{E}_{\text{init}}\left[\underbrace{\|\widehat{Q}_s^{\Pi^\theta}(\cdot) - \text{Proj}_{\mathcal{F}_{R,M}}\widehat{Q}_s^{\Pi^\theta}(\cdot)\|_{L^2(\nu_\theta)}^2}_{\text{(II)}}\right] \le \mathcal{O}\left(\frac{R^2 d_{\zeta_s^k}}{M}\right). \tag{D.13}$$

With this lemma, Theorem 5.1 follows immediately by plugging (D.13) into (D.12), and setting $c = \frac{r_{\max}}{1-\gamma}$, $\rho = \sqrt{\gamma}$, $T_{\text{critic}} = \Omega(M)$ in (D.12). Q.E.D.

D.2. Proof of Theorem 5.2: Convergence of Actor Update

The proof of Theorem 5.2 consists of two steps: the first step in Section D.2.1 shows that the actor update converges to a stationary point of J (4.1), and the second step in Section D.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 $\mathcal{B}_s^{\text{actor}} := \{\theta_s \in \mathbb{R}^{M\times d_{\zeta_s}} : \|\theta_s - \theta_s(0)\|_\infty \le R/\sqrt{M}\}$ 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 $\mathbb{R}^{M\times d_\zeta}$.

D.2.1. Convergence to Stationary Point.

Definition D.1. A point $\tilde{\theta} \in \mathcal{B}$ is called a stationary point of $J(\cdot)$ if it holds that

$$\nabla_\theta J(\tilde{\theta})^\top(\theta - \tilde{\theta}) \le 0, \quad \forall \theta \in \mathcal{B}. \tag{D.14}$$

Define the following mapping G from $\mathbb{R}^{M\times d_\zeta}$ to itself:

$$G(\theta) := \eta^{-1}\cdot[\text{Proj}_{\mathcal{B}}\Big(\theta + \eta\cdot\nabla_\theta J(\theta)\Big) - \theta]. \tag{D.15}$$

It is well-known that (D.14) holds if and only if $G(\tilde{\theta}) = 0$ (Sra et al. [53]). Now, denote $\rho(t) := G(\theta(t))$, where $\theta(t) = \{\theta_s(t)\}_{s\in\mathcal{S}}$ is the actor parameter updated in Algorithm 2 in iteration t.

To show that Algorithm 2 converges to a stationary point, we focus on analyzing $\|\rho(t)\|_2$.

Theorem D.2. *Assume Assumptions 5.3–5.5. Set $\eta = (T_{\text{actor}})^{-1/2}$ and assume $1 - L\eta \ge 1/2$, where L is the Lipschitz constant in Assumption 5.5. Then, the output $\{\theta(t)\}_{t\in[T_{\text{actor}}]}$ of Algorithm 2 satisfies*

$$\min_{t\in[T_{\text{actor}}]}\mathbb{E}[\|\rho(t)\|_2^2] \le \frac{8\tau^2\Sigma^2|\mathcal{S}|}{B} + \frac{4}{\sqrt{T_{\text{actor}}}}\mathbb{E}[J(\theta(T_{\text{actor}}+1)) - J(\theta(1))] + \epsilon_Q(T_{\text{actor}}). \tag{D.16}$$

Here, ϵ_Q measures the error accumulated from the critic steps, which is defined as

$$\begin{aligned}\epsilon_Q(T_{\text{actor}}) = {}& \frac{32\tau DRd_{\zeta_s}^{1/2}|\mathcal{S}|}{(1-\gamma)\eta T_{\text{actor}}}\cdot\sum_{t=1}^{T_{\text{actor}}}\sum_{s\in\mathcal{S}}\mathbb{E}[\|Q_s(\cdot;\bar{\omega}_s,t) - Q_s^{\Pi^{\theta(t)}}(\cdot)\|_{L^2(\nu_{\theta(t)})}] \\ & + \frac{16\tau^2D^2|\mathcal{S}|^2}{(1-\gamma)^2T_{\text{actor}}}\cdot\sum_{t=1}^{T_{\text{actor}}}\sum_{s\in\mathcal{S}}\mathbb{E}[\|Q_s(\cdot;\bar{\omega}_s,t) - Q_s^{\Pi^{\theta(t)}}(\cdot)\|_{L^2(\nu_{\theta(t)})}^2],\end{aligned} \tag{D.17}$$

where $\{Q_s(\cdot;\bar{\omega}_s,t)\}_{s\in\mathcal{S}}$ is the output of the critic update at step t in Algorithm 2. All expectations in (D.16) and (D.17) are taken over all randomness in Algorithms 1 and 2.

Proof of Theorem D.2. 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 5.5, $\nabla_\theta J(\theta)$ is L-Lipschitz continuous. Hence, by Taylor's expansion,

$$J(\theta(t+1)) - J(\theta(t)) \ge \eta\cdot\nabla_\theta J(\theta(t))^\top\delta(t) - L/2\cdot\|\theta(t+1) - \theta(t)\|_2^2, \tag{D.18}$$

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 (4.14) and the expectation is taken over $\sigma_{\theta(t)}$ given $\{\bar{\omega}_s\}_{s\in\mathcal{S}}$. Then,

$$\begin{aligned}\nabla_\theta J(\theta(t))^\top\delta(t) &= \sum_{s\in\mathcal{S}}\nabla_{\theta_s}J(\theta(t))^\top\delta_s(t) \\ &= \sum_{s\in\mathcal{S}}[(\nabla_{\theta_s}J(\theta(t)) - \mathbb{E}[\widehat{g}_s(\theta(t))])^\top\delta_s(t) - \xi_s(t)^\top\delta_s(t) + \widehat{g}_s(\theta(t))^\top\delta_s(t)],\end{aligned} \tag{D.19}$$

where $\delta_s(t) := (\theta_s(t+1) - \theta_s(t))/\eta$. The first term in (D.19) represents the error of estimating $\nabla_{\theta_s}J(\theta(t))$ using

$$\mathbb{E}[\widehat{g}_s(\theta(t))] = \frac{1}{1-\gamma}\mathbb{E}_{\sigma_{\theta(t)}}\left[\left[\sum_{y\in\mathcal{N}_s^k}Q_y(\mu(\mathcal{N}_y^k),h(\mathcal{N}_y^k);\bar{\omega}_y,t)\right]\nabla_{\theta_s}\log\Pi^{\theta_s}(h(s)|\mu(s))\right].$$

To bound the first term, first notice that

$$\mathbb{E}[\widehat{g}_s(\theta(t))] = \frac{1}{1-\gamma}\mathbb{E}_{\sigma_{\theta(t)}}\left[\left[\sum_{y\in\mathcal{S}}Q_y(\mu(\mathcal{N}_y^k),h(\mathcal{N}_y^k);\bar{\omega}_y,t)\right]\nabla_{\theta_s}\log\Pi^{\theta_s}(h(s)|\mu(s))\right].$$

This is because, for all $y \notin \mathcal{N}_s^k$, $Q_y(\mu(\mathcal{N}_y^k), h(\mathcal{N}_y^k); \bar{\omega}_y)$ is independent of s, and consequently, we can verify that

$$\mathbb{E}_{\sigma_{\theta(t)}}\left[\left[\sum_{y\notin\mathcal{N}_s^k} Q_y(\mu(\mathcal{N}^k(y)), h(\mathcal{N}^k(y)); \bar{\omega}_y, t)\right]\nabla_{\theta_s}\log\Pi^{\theta_s}(h(s)|\mu(s))\right] = 0.$$

Therefore, following the similar computation in Cai et al. [7, lemma D.2], we have

$$|(\nabla_{\theta_s}J(\theta(t)) - \mathbb{E}[\widehat{g}_s(\theta(t))])^\top\delta_s(t)| \le \frac{4\tau DRd_{\zeta_s}^{1/2}}{(1-\gamma)\eta}\sum_{s\in\mathcal{S}}\|Q_s(\cdot;\bar{\omega}_s,t) - Q_s^{\theta(t)}(\cdot)\|_{L^2(\nu_{\theta(t)})}. \tag{D.20}$$

To bound the second term in (D.19), we simply have

$$\xi_s(t)^\top\delta_s(t) \le \|\xi_s(t)\|_2^2 + \|\delta_s(t)\|_2^2. \tag{D.21}$$

To handle the last term in (D.19), we have

$$\begin{aligned}
&\widehat{g}_s(\theta(t))^\top\delta_s(t) - \|\delta_s(t)\|_2^2 = \eta^{-1}\cdot(\eta\widehat{g}_s(\theta(t)) - (\theta_s(t+1) - \theta_s(t)))^\top\delta_s\\
=&\eta^{-1}\cdot(\theta_s(t+1/2) - \mathrm{Proj}_{\mathcal{B}_s}(\theta_s(t+1/2)))^\top\delta_s(t)\\
=&\eta^{-2}\cdot(\theta_s(t+1/2) - \mathrm{Proj}_{\mathcal{B}_s}(\theta_s(t+1/2)))^\top(\mathrm{Proj}_{\mathcal{B}_s}(\theta_s(t+1/2)) - \theta_s(t)) \ge 0\,.
\end{aligned} \tag{D.22}$$

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 (D.19)–(D.22) suggests

$$\nabla_{\theta_s}J(\theta(t))^\top\delta_s(t) \ge -\frac{4\tau DRd_{\zeta_s}^{1/2}}{(1-\gamma)\eta}\sum_{s\in\mathcal{S}}[\|Q_s(\cdot;\bar{\omega}_s,t) - Q_s^{\theta(t)}(\cdot)\|_{L^2(\nu_{\theta(t)})}] + \frac{1}{2}(\|\delta_s(t)\|_2^2 - \|\xi_s(t)\|_2^2).$$

Consequently,

$$\nabla_\theta J(\theta(t))^\top\delta(t) \ge -\frac{4\tau DRd_{\zeta_s}^{1/2}}{(1-\gamma)\eta}|\mathcal{S}|\sum_{s\in\mathcal{S}}[\|Q_s(\cdot;\bar{\omega}_s,t) - Q_s^{\Pi^{\theta(t)}}(\cdot)\|_{L^2(\nu_{\theta(t)})}] + \frac{1}{2}(\|\delta(t)\|_2^2 - \|\xi(t)\|_2^2). \tag{D.23}$$

Thus, by plugging (D.23) into (D.18) and by Assumption 5.3, we have

$$\begin{aligned}
\frac{1-L\cdot\eta}{2}\mathbb{E}[\|\delta(t)\|_2^2] \le{}& \eta^{-1}\cdot\mathbb{E}[J(\theta(t+1)) - J(\theta(t))] + \frac{\tau^2\Sigma^2|\mathcal{S}|}{2B}\\
&+ \frac{4\tau DRd_{\zeta_s}^{1/2}|\mathcal{S}|}{(1-\gamma)\eta}\sum_{s\in\mathcal{S}}\|Q_s(\cdot;\bar{\omega}_s,t) - Q_s^{\Pi^{\theta(t)}}(\cdot)\|_{L^2(\nu_{\theta(t)})}.
\end{aligned} \tag{D.24}$$

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 (D.24) and $\|\rho(t)\|_2 = \|G(\theta(t))\|_2$ in (D.15), we next 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 $\mathbb{R}^{M\times d_{\zeta_s}}$:

$$G_s(\theta) := \eta^{-1}\cdot[\mathrm{Proj}_{\mathcal{B}_s}(\theta_s + \eta\cdot\nabla_{\theta_s}J(\theta)) - \theta_s]. \tag{D.25}$$

Because $\mathcal{B}_s$ 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 $\rho_s(t)$,

$$\begin{aligned}
\|\delta_s(t) - \rho_s(t)\|_2 &= \eta^{-1}\cdot\|\mathrm{Proj}_{\mathcal{B}_s}(\theta_s + \eta\cdot\nabla_{\theta_s}J(\theta)) - \theta_s - \mathrm{Proj}_{\mathcal{B}_s}(\theta_s + \eta\cdot\widehat{g}_s(\theta)) + \theta_s\|_2\\
&= \eta^{-1}\cdot\|\mathrm{Proj}_{\mathcal{B}_s}(\theta_s + \eta\cdot\nabla_{\theta_s}J(\theta)) - \mathrm{Proj}_{\mathcal{B}_s}(\theta_s + \eta\cdot\widehat{g}_s(\theta))\|_2\\
&\le \eta^{-1}\cdot\|\theta_s + \eta\cdot\nabla_{\theta_s}J(\theta) - \theta_s + \eta\cdot\widehat{g}_s(\theta)\|_2 = \|\nabla_{\theta_s}J(\theta) - \widehat{g}_s(\theta)\|_2.
\end{aligned}$$

Following similar calculations in Cai et al. [7, lemma D.3],

$$\mathbb{E}[\|\nabla_{\theta_s} J(\theta) - \widehat{g}_s(\theta)\|_2^2] \le \frac{2\tau^2\Sigma^2}{B} + \frac{8\tau^2 D^2}{(1-\gamma)^2}\left(\sum_{s\in\mathcal{S}} \|Q_s(\cdot\,;\bar{\omega}_s,t) - Q_s^{\Pi^{\theta(t)}}(\cdot)\|_{L^2(\nu_{\theta(t)})}\right)^2$$
$$\le \frac{2\tau^2\Sigma^2}{B} + \frac{8\tau^2 D^2|\mathcal{S}|}{(1-\gamma)^2}\left(\sum_{s\in\mathcal{S}} \|Q_s(\cdot\,;\bar{\omega}_s,t) - Q_s^{\Pi^{\theta(t)}}(\cdot)\|_{L^2(\nu_{\theta(t)})}^2\right). \tag{D.26}$$

The expectation is taken over $\sigma_{\theta(t)}$ given $\{\bar{\omega}_s\}_{s\in\mathcal{S}}$. Consequently,

$$\mathbb{E}[\|\delta(t) - \rho(t)\|_2^2] \le \frac{2\tau^2\Sigma^2|\mathcal{S}|}{B} + \frac{8\tau^2 D^2|\mathcal{S}|^2}{(1-\gamma)^2}\left(\sum_{s\in\mathcal{S}} \|Q_s(\cdot\,;\bar{\omega}_s,t) - Q_s^{\Pi^{\theta(t)}}(\cdot)\|_{L^2(\nu_{\theta(t)})}^2\right). \tag{D.27}$$

Set $\eta = 1/\sqrt{T_{\text{actor}}}$ and take (D.24) and (D.27). We obtain (D.16) from the following estimations:

$$\min_{t\in[T_{\text{actor}}]} \mathbb{E}[\|\rho(t)\|_2^2] \le \frac{1}{T_{\text{actor}}}\cdot\sum_{t=1}^{T_{\text{actor}}}\|\rho(t)\|_2^2 \le \frac{2}{T_{\text{actor}}}\cdot\sum_{t=1}^{T_{\text{actor}}}(\mathbb{E}[\|\delta(t)-\rho(t)\|_2^2] + \mathbb{E}[\|\delta(t)\|_2^2])$$
$$\le \frac{2}{T_{\text{actor}}}\cdot\sum_{t=1}^{T_{\text{actor}}}(\mathbb{E}[\|\delta(t)-\rho(t)\|_2^2] + 2(1-L\cdot\eta)\mathbb{E}[\|\delta(t)\|_2^2])$$
$$\le \frac{8\tau^2\Sigma^2|\mathcal{S}|}{B} + \frac{4}{\sqrt{T_{\text{actor}}}}\mathbb{E}[J(\theta(T_{\text{actor}}+1)) - J(\theta(1))] + \epsilon_Q(T_{\text{actor}}),$$

where ϵ_Q measures the error accumulated from the critic steps, which are defined in (D.17), that is,

$$\epsilon_Q(T_{\text{actor}}) = \frac{32\tau D R d_{\zeta_s}^{1/2}|\mathcal{S}|}{(1-\gamma)\eta T_{\text{actor}}}\cdot\sum_{t=1}^{T_{\text{actor}}}\sum_{s\in\mathcal{S}}\mathbb{E}[\|Q_s(\cdot\,;\bar{\omega}_s) - Q_s^{\Pi^{\theta(t)}}(\cdot)\|_{L^2(\nu_{\theta(t)})}]$$
$$+\frac{16\tau^2 D^2|\mathcal{S}|^2}{(1-\gamma)^2 T_{\text{actor}}}\cdot\sum_{t=1}^{T_{\text{actor}}}\sum_{s\in\mathcal{S}}\mathbb{E}[\|Q_s(\cdot\,;\bar{\omega}_s) - Q_s^{\Pi^{\theta(t)}}(\cdot)\|_{L^2(\nu_{\theta(t)})}^2].$$

Here, the expectations in (D.16) and (D.17) are taken over all randomness in Algorithms 1 and 2. Q.E.D.

D.2.2. Bridging the Gap Between Stationarity and Optimality. Recall that σ_θ in (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,

$$\bar{\sigma}_\theta(\mu)\Pi^\theta(h|\mu) = \sigma_\theta(\mu,h).$$

Following similar steps in the proof of Cai et al. [7, theorem 4.8], one can characterize the global optimality of the obtained stationary point $\tilde{\theta}\in\mathcal{B}$ as the following.

Lemma D.3. *Let $\tilde{\theta}\in\mathcal{B}$ be a stationary point of $J(\cdot)$ satisfying Condition (D.14) and let $\theta^*\in\mathcal{B}$ be the global maximum point of $J(\cdot)$ in $\mathcal{B}$. Then, the following inequality holds:*

$$(1-\gamma)(J(\theta^*) - J(\tilde{\theta})) \le \frac{2r_{\max}}{1-\gamma}\inf_{\theta\in\mathcal{B}}\left\|u_{\tilde{\theta}}(\mu,h) - \sum_{s\in\mathcal{S}}\phi_{\tilde{\theta}_s}(\mu(s),h(s))^\top\theta_s\right\|_{L^2(\sigma_{\tilde{\theta}})}, \tag{D.28}$$

where $u_{\tilde{\theta}}(\mu,h) := d\sigma_{\theta^*}/d\sigma_{\tilde{\theta}}(\mu,h) - d\bar{\sigma}_{\theta^*}/d\bar{\sigma}_{\tilde{\theta}}(\mu) + \sum_{s\in\mathcal{S}}\phi_{\tilde{\theta}_s}(\mu(s),h(s))^\top\tilde{\theta}_s$, and $\mathrm{d}\sigma_{\theta^*}/\mathrm{d}\sigma_{\tilde{\theta}}$, $\mathrm{d}\bar{\sigma}_{\theta^*}/\mathrm{d}\bar{\sigma}_{\tilde{\theta}}$ are the Radon–Nikodym derivatives between the corresponding measures.

Proof of Lemma D.3. First, recall that, by (4.8), for any $\theta\in\mathcal{B}$,

$$\nabla_\theta J(\tilde{\theta})^\top(\theta-\tilde{\theta}) = \sum_{s\in\mathcal{S}}\nabla_{\theta_s}J(\tilde{\theta})^\top(\theta_s-\tilde{\theta}_s) = \frac{\tau}{1-\gamma}\sum_{s\in\mathcal{S}}\mathbb{E}_{\sigma_{\tilde{\theta}}}[Q^{\Pi^{\tilde{\theta}}}(\mu,h)\cdot\Phi(\tilde{\theta},s,\mu,h)^\top(\theta_s-\tilde{\theta}_s)],$$

in which $\Phi(\theta,s,\mu,h) := \phi_{\theta_s}(\mu(s),h(s)) - \mathbb{E}_{h(s)'\sim\Pi_s^{\theta_s}(\cdot|\mu(s))}[\phi_{\theta_s}(\mu(s),h'(s))]$ is defined in (4.7).

Because $\tilde{\theta}\in\mathcal{B}$ is a stationary point of $J(\cdot)$,

$$\sum_{s\in\mathcal{S}}\mathbb{E}_{\sigma_{\tilde{\theta}}}[Q^{\Pi^{\tilde{\theta}}}(\mu,h)\cdot\Phi(\tilde{\theta},s,\mu,h)^\top(\theta_s-\tilde{\theta}_s)] \le 0, \quad \forall\theta\in\mathcal{B}. \tag{D.29}$$

Denote $A^{\Pi^{\tilde{\theta}}}(\mu,h) := Q^{\Pi^{\tilde{\theta}}}(\mu,h) - V^{\Pi^{\tilde{\theta}}}(\mu)$ as the advantage function under policy $\Pi^{\tilde{\theta}}$. It holds from the definition that $\mathbb{E}_{h\sim\Pi^{\tilde{\theta}}}(\cdot|\mu)[A^{\Pi^{\tilde{\theta}}}(\mu,h)] = V^{\Pi^{\tilde{\theta}}}(\mu) - V^{\Pi^{\tilde{\theta}}}(\mu) = 0$. Meanwhile, $\sup_{(\mu,h)\in\Xi}|A^{\Pi^{\tilde{\theta}}}(\mu,h)| \le 2\sup_{\mu\in\mathcal{P}^N(\mathcal{S})}|V^{\Pi^{\tilde{\theta}}}(\mu)| \le \frac{2r_{\max}}{1-\gamma}$.

Given that $\mathbb{E}_{h\sim\Pi^{\tilde\theta}(\cdot|\mu)}[A^{\Pi^{\tilde\theta}}(\mu,h)]=0$ and $\mathbb{E}_{h\sim\Pi^{\tilde\theta}(\cdot|\mu)}[\Phi(\tilde\theta,s,\mu,h)]=0$, we have, for any $s\in\mathcal{S}$,

$$\mathbb{E}_{\sigma_{\tilde\theta}}\left[V^{\Pi^{\tilde\theta}}(\mu)\cdot\Phi(\tilde\theta,s,\mu,h)\right]=0,\qquad\text{and}\tag{D.30}$$

$$\mathbb{E}_{\sigma_{\tilde\theta}}\left[A^{\Pi^{\tilde\theta}}(\mu,h)\cdot\mathbb{E}_{h(s)'\sim\Pi_s^{\tilde\theta_s}(\cdot|\mu(s))}[\phi_{\tilde\theta_s}(\mu(s),h'(s))]\right]=0.\tag{D.31}$$

Combining (D.29) with (D.30) and (D.31),

$$\sum_{s\in\mathcal{S}}\mathbb{E}_{\sigma_{\tilde\theta}}\left[A^{\Pi^{\tilde\theta}}(\mu,h)\cdot\phi_{\tilde\theta_s}(\mu(s),h(s))^\top(\theta_s-\tilde\theta_s)\right]\le 0,\qquad\forall\theta\in\mathcal{B}.\tag{D.32}$$

Moreover, by the performance difference lemma (Kakade and Langford [35]),

$$(1-\gamma)\cdot(J(\theta^*)-J(\widehat\theta))=\mathbb{E}_{\bar\sigma_{\theta^*}}[\langle A^{\Pi^{\tilde\theta}}(\mu,\cdot),\Pi^{\theta^*}(\cdot|\mu)-\Pi^{\tilde\theta}(\cdot|\mu)\rangle].\tag{D.33}$$

Combining (D.33) with (D.32), it holds that, for any $\theta\in\mathcal{B}$,

$$\begin{aligned}&(1-\gamma)\cdot(J(\theta^*)-J(\widehat\theta))\\&\le\mathbb{E}_{\bar\sigma_{\theta^*}}[\langle A^{\Pi^{\tilde\theta}}(\mu,\cdot),\Pi^{\theta^*}(\cdot|\mu)-\Pi^{\tilde\theta}(\cdot|\mu)\rangle]-\sum_{s\in\mathcal{S}}\mathbb{E}_{\sigma_{\tilde\theta}}[A^{\Pi^{\tilde\theta}}(\zeta)\cdot\phi_{\tilde\theta_s}(\zeta_s)^\top(\theta_s-\tilde\theta_s)]\\&=\mathbb{E}_{\sigma_{\tilde\theta}}\left[A^{\Pi^{\tilde\theta}}(\mu,h)\cdot\left(\frac{\mathrm{d}\sigma_{\theta^*}}{\mathrm{d}\sigma_{\tilde\theta}}(\mu,h)-\frac{\mathrm{d}\bar\sigma_{\theta^*}}{\mathrm{d}\bar\sigma_{\tilde\theta}}(\mu)-\sum_{s\in\mathcal{S}}\phi_{\tilde\theta_s}(\mu(s),h(s))^\top(\theta_s-\tilde\theta_s)\right)\right].\end{aligned}\tag{D.34}$$

Therefore,

$$\begin{aligned}&(1-\gamma)\cdot(J(\theta^*)-J(\widehat\theta))\\&\le\frac{2r_{\max}}{1-\gamma}\inf_{\theta\in\mathcal{B}}\left\|\frac{\mathrm{d}\sigma_{\theta^*}}{\mathrm{d}\sigma_{\tilde\theta}}(\mu,h)-\frac{\mathrm{d}\bar\sigma_{\theta^*}}{\mathrm{d}\bar\sigma_{\tilde\theta}}(\mu)-\sum_{s\in\mathcal{S}}\phi_{\tilde\theta_s}(\mu(s),h(s))^\top(\theta_s-\tilde\theta_s)\right\|_{L^2(\sigma_{\tilde\theta})}\\&=\frac{2r_{\max}}{1-\gamma}\inf_{\theta\in\mathcal{B}}\left\|u_{\tilde\theta}(\mu,h)-\sum_{s\in\mathcal{S}}\phi_{\tilde\theta_s}(\mu(s),h(s))^\top\theta_s\right\|_{L^2(\sigma_{\tilde\theta})},\end{aligned}\tag{D.35}$$

where $u_{\tilde\theta}(\mu,h):=d\sigma_{\theta^*}/d\sigma_{\tilde\theta}(\mu,h)-d\bar\sigma_{\theta^*}/d\bar\sigma_{\tilde\theta}(\mu)+\sum_{s\in\mathcal{S}}\phi_{\tilde\theta_s}(\mu(s),h(s))^\top\tilde\theta_s$, and $\mathrm{d}\sigma_{\theta^*}/\mathrm{d}\sigma_{\tilde\theta}$, $\mathrm{d}\bar\sigma_{\theta^*}/\mathrm{d}\bar\sigma_{\tilde\theta}$ are the Radon–Nikodym derivatives between corresponding measures. Q.E.D.

To further bound the right-hand side of (D.28) in Lemma D.3, define the following function class:

$$\begin{aligned}\tilde{\mathcal{F}}_{R,M}=\Bigg\{f_0(\zeta;\theta):=\sum_{s\in\mathcal{S}}\underbrace{\left[\frac{1}{\sqrt{M}}\sum_{m=1}^{M}\mathbb{1}\{[\theta_s(0)]_m^\top\zeta_s>0\}[\theta_s]_m^\top\zeta_s\right]}_{(\star)}:\\\theta_s\in\mathbb{R}^{M\times d_{\zeta_s}},\|\theta_s-\theta_s(0)\|_\infty\le R/\sqrt{M}\Bigg\},\end{aligned}\tag{D.36}$$

given an initialization $\theta_s(0)\in\mathbb{R}^{M\times d_{\zeta_s}}$, $s\in\mathcal{S}$ and $b\in\mathbb{R}^M$. $\tilde{\mathcal{F}}_{R,M}$ (D.36) is a local linearization of the actor neural network. More specifically, term $(\star)$ in (D.36) locally linearizes the decentralized actor neural network $f(\zeta_s;\theta_s)$ (4.4) with respect to θ_s. Any $f_0(\zeta;\theta)\in\tilde{\mathcal{F}}_{R,M}$ is a sum of $|\mathcal{S}|$ inner products between feature mapping $\phi_{\theta_s(0)}(\cdot)$ (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 $M\to\infty$, $\tilde{\mathcal{F}}_{R,M}$ converges to $\mathcal{F}_{R,\infty}$ (defined in Assumption 5.6). The approximation error between $\tilde{\mathcal{F}}_{R,M}$ and $\mathcal{F}_{R,\infty}$ is bounded in the following lemma.

Lemma D.4. *For any function $f(\zeta)\in\mathcal{F}_{R,\infty}$ defined in Assumption 5.6, we have*

$$\mathbb{E}_{\text{init}}[\|f(\cdot)-\mathrm{Proj}_{\tilde{\mathcal{F}}_{R,M}}f(\cdot)\|_{L^2(\sigma_{\tilde\theta})}]\le\mathcal{O}\left(\frac{|\mathcal{S}|Rd_{\zeta_s}^{1/2}}{M^{1/2}}\right).\tag{D.37}$$

Lemma D.4 follows from Rahimi and Recht [50] and Cai et al. [7, 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 D.4, we are ready to establish an upper bound for the right-hand side of (D.28) in the following proposition.

Proposition D.1. *Under Assumption 5.6, let $\tilde\theta\in\mathcal{B}$ 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:*

$$(1-\gamma)(J(\theta^*)-J(\tilde\theta))\le\mathcal{O}\left(\frac{|\mathcal{S}|R^{3/2}d_{\zeta_s}^{3/4}}{M^{1/4}}\right).\tag{D.38}$$

Proof of Proposition D.1. First, by the triangle inequality,

$$\inf_{\theta\in\mathcal{B}}\left\|u_{\tilde\theta}(\zeta)-\sum_{s\in\mathcal{S}}\phi_{\tilde\theta_s}(\zeta_s)^\top\theta_s\right\|_{L^2(\sigma_{\tilde\theta})}\le\|u_{\tilde\theta}(\zeta)-\mathrm{Proj}_{\tilde{\mathcal{F}}_{R,M}}u_{\tilde\theta}(\zeta)\|_{L^2(\sigma_{\tilde\theta})}+\inf_{\theta\in\mathcal{B}}\left\|\mathrm{Proj}_{\tilde{\mathcal{F}}_{R,M}}u_{\tilde\theta}(\zeta)-\sum_{s\in\mathcal{S}}\phi_{\tilde\theta_s}(\zeta_s)^\top\theta_s\right\|_{L^2(\sigma_{\tilde\theta})},\tag{D.39}$$

where $\tilde{\mathcal{F}}_{R,M}$ is defined in (D.36). We denote $\mathrm{Proj}_{\tilde{\mathcal{F}}_{R,M}}u_{\tilde\theta}(\zeta)=\sum_{s\in\mathcal{S}}\phi_{\theta_s(0)}(\zeta_s)\cdot\widehat\theta_s\in\tilde{\mathcal{F}}_{R,M}$ for some $\widehat\theta\in\mathcal{B}$. Therefore, by Lemma D.4, the first term on the right-hand side of (D.39) is bounded by (D.37):

$$\left\|u_{\tilde\theta}(\zeta)-\sum_{s\in\mathcal{S}}\phi_{\theta_s(0)}(\zeta_s)\cdot\widehat\theta_s\right\|_{L^2(\sigma_{\tilde\theta})}\le\mathcal{O}\left(\frac{|\mathcal{S}|Rd_{\zeta_s}^{1/2}}{M^{1/2}}\right).$$

The following Lemma D.5 is a direct application of Wang et al. [57, lemma E.2], which is used to bound the second term on the right-hand side of (D.39).

Lemma D.5. *It holds for any* $\theta_s,\theta_s'\in\mathcal{B}_s=\{\alpha_s\in\mathbb{R}^{M\times d_{\zeta_s}}:\|\alpha_s-\theta_s(0)\|_\infty\le R/\sqrt{M}\}$ *that*

$$\mathbb{E}_{\mathrm{init}}[\|\phi_{\theta_s}(\zeta_s)^\top\theta_s'-\phi_{\theta_s(0)}(\zeta_s)^\top\theta_s'\|_{L^2(\sigma_\theta)}]\le\mathcal{O}\left(\frac{R^{3/2}d_{\zeta_s}^{3/4}}{M^{1/4}}\right),\tag{D.40}$$

where the expectation is taken over random initialization.

Taking $\theta=\tilde\theta$ and $\theta'=\widehat\theta$ in Lemma D.5 gives us

$$\sum_{s\in\mathcal{S}}\|\phi_{\theta_s(0)}(\zeta_s)\cdot\widehat\theta_s-\phi_{\tilde\theta_s}(\zeta_s)^\top\widehat\theta_s\|_{L^2(\sigma_{\tilde\theta})}\le\mathcal{O}\left(\frac{|\mathcal{S}|R^{3/2}d_{\zeta_s}^{3/4}}{M^{1/4}}\right).$$

Therefore, by Lemma D.1,

$$(1-\gamma)(J(\theta^*)-J(\tilde\theta))\le\inf_{\theta\in\mathcal{B}}\left\|u_{\tilde\theta}(\zeta)-\sum_{s\in\mathcal{S}}\phi_{\tilde\theta_s}(\zeta_s)^\top\theta_s\right\|_{L^2(\sigma_{\tilde\theta})}\le\mathcal{O}\left(\frac{|\mathcal{S}|R^{3/2}d_{\zeta_s}^{3/4}}{M^{1/4}}\right).\quad\text{Q.E.D.}$$

Now, we are ready to establish Theorem 5.2.

Proof of Theorem 5.2. Following similar calculations as in Wang et al. [57, section H.3], we obtain that, at iteration $t\in[T_{\mathrm{actor}}]$,

$$\nabla_\theta J(\theta(t))^\top(\theta-\theta(t))\le2\left(R+\frac{\eta\cdot r_{\max}}{1-\gamma}\right)\cdot\|\rho(t)\|_2,\quad\forall\theta\in\mathcal{B}.\tag{D.41}$$

The right-hand side of (D.41) quantifies the deviation of $\theta(t)$ from a stationary point $\tilde\theta$. Having (D.41) and following similar arguments for Lemma D.3 and Proposition D.1, we can show that

$$(1-\gamma)\min_{t\in[T_{\mathrm{actor}}]}\mathbb{E}[J(\theta^*)-J(\theta(t))]\le\mathcal{O}\left(\frac{|\mathcal{S}|R^{3/2}d_{\zeta_s}^{3/4}}{M^{1/4}}\right)+2\left(R+\frac{\eta\cdot r_{\max}}{1-\gamma}\right)\cdot\min_{t\in[T_{\mathrm{actor}}]}\mathbb{E}[\|\rho(t)\|_2].\tag{D.42}$$

Here, the last term $\min_{t\in[T_{\mathrm{actor}}]}\mathbb{E}[\|\rho(t)\|_2]$ is bounded by (D.16) in Theorem D.2, whereas the term $\epsilon_Q(T_{\mathrm{actor}})$ in (D.17) can be upper bounded by Theorem 5.1. Finally, with the parameters stated in Theorem 5.2, the following statement holds by straightforward calculation:

$$\min_{t\in[T_{\mathrm{actor}}]}\mathbb{E}[J(\theta^*)-J(\theta(t))]\le\mathcal{O}(|\mathcal{S}|^{1/2}B^{-1/2}+|\mathcal{S}||\mathcal{A}|^{1/4}(\gamma^{k/8}+(T_{\mathrm{actor}})^{-1/4})).\quad\text{Q.E.D.}$$

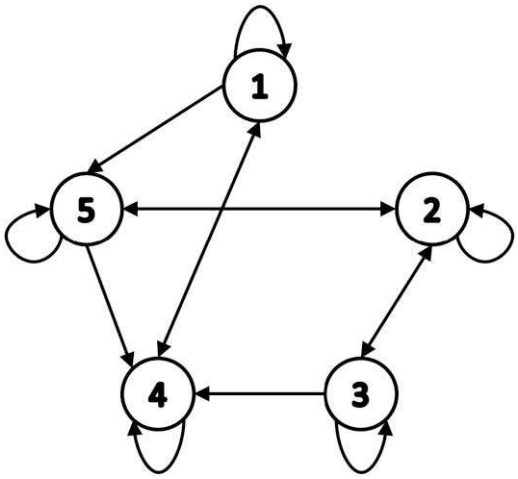
Appendix E. A Network Example Satisfying Technical Assumptions

In this section, we provide a concrete network example that satisfies all Assumptions 5.1–5.6 (or their mild relaxations). The structure of this network is shown in Figure E.1, 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 10 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 transits 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 parameterized in the form of (4.4).

E.1. Assumption 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 (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 5.1 with some controllable approximation error. More specifically, as pointed out in Remark 5.1, $\mathcal{F}_{R,\infty}^{s,k}$ in (5.1) is a subset of an RKHS,

Figure E.1. Five-state network.



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 5.1 is satisfied by the example in Figure E.1 up to some approximation error.

E.2. Assumption 5.2.

As mentioned in Remark 5.1, Assumption 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 E.1.

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 E.2. It is observed from the histogram that most of the randomly chosen θ's lead to a maximum value smaller than 0.02, whereas the overall upper bound is smaller than 0.03. Therefore, Assumption 5.2 holds numerically under this example.

E.3. Assumption 5.3.

Assumption 5.3 also holds under mild conditions. More specifically, when the estimator $\widehat{g}_s$ in (4.14) can be viewed as an average of B i.i.d. samples,

$$\left[\sum_{y\in\mathcal{N}_s^k} Q_y(\mu_l(\mathcal{N}_y^k), h_l(\mathcal{N}_y^k); \bar{\omega}_y)\right] \cdot \widehat{\Phi}(\theta(t), s, \mu_l, h_l), \quad l \in [B],$$

Figure E.2. (Color online) Upper bound of σ_θ and ν_θ over 800 random policies.

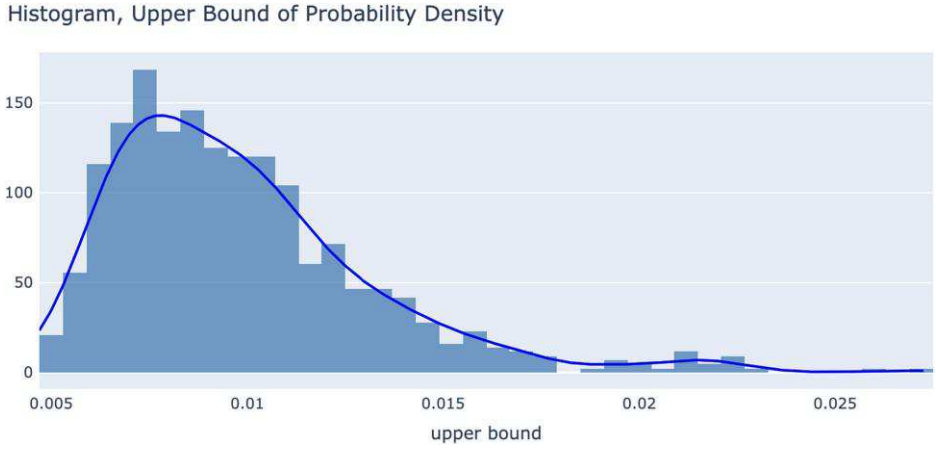
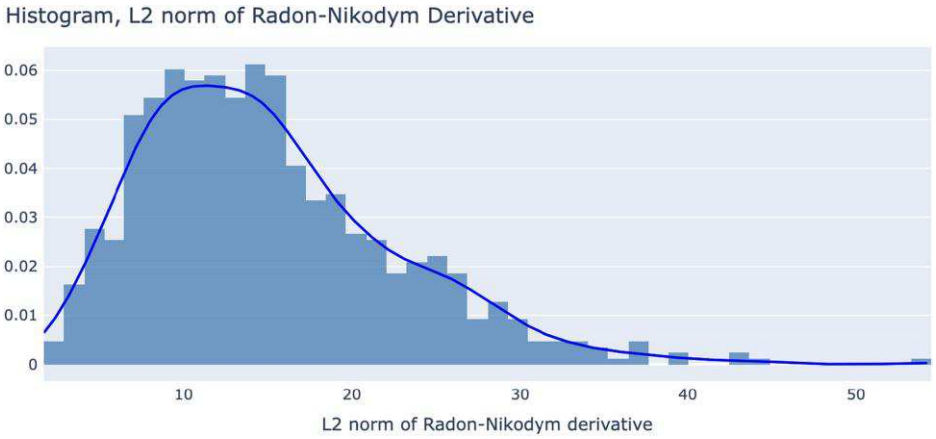


Figure E.3. (Color online) L_2 norm of Radon–Nikodym derivative between σ_θ and ν_θ over 800 random policies.



Assumption 5.3 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 5.3 holds when the parameters of the critic networks are uniformly bounded.

E.4. Assumption 5.4.

Similar to Assumption 5.2, because of the difficulty in directly computing ν_θ and σ_θ, Assumption 5.4 is verified numerically under the example in Figure E.1. Again, 800 random neural policies $\{\Pi^{\theta_i}\}_{i=1}^{800}$ are generated, and $\mathbb{E}_{\nu_\theta}[(\mathrm{d}\sigma_\theta/\mathrm{d}\nu_\theta(\mu,h))^2]$, the L_2 norm of the Radon–Nikodym derivative between σ_θ and ν_θ, is computed for each θ. The results are shown in Figure E.3. It is observed from the histogram that most of the randomly chosen θ's lead to a bounded L_2 norm smaller than 30, whereas the overall upper bound is smaller than 45. Therefore, Assumption 5.4 holds numerically under this example.

E.5. Assumption 5.5.

In general, Assumption 5.5 holds when the transition probability and the reward function are both Lipschitz continuous with respect to their inputs (Pirotta et al. [46]), 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 (Zhang et al. [65]. Under the particular example in Figure E.1, one can set the reward function to be constant so that the Lipschitz condition in Assumption 5.5 holds immediately.

E.6. Assumption 5.6.

Assumption 5.6 is similar to Assumption 5.1, and such assumption is satisfied by any continuous function up to an approximation error.

Overall, we have shown that Assumptions 5.1–5.6 in the paper (or their mild relaxations) are satisfied by the particular example in Figure E.1.

References

[1] Agarwal A, Kakade SM, Lee JD, Mahajan G (2021) On the theory of policy gradient methods: Optimality, approximation, and distribution shift. *J. Machine Learn. Res.* 22(98):1–76.
[2] Aïd R, Dumitrescu R, Tankov P (2021) The entry and exit game in the electricity markets: A mean-field game approach. *J. Dynamics Games* 8(4):331–358.
[3] Allen-Zhu Z, Li Y, Liang Y (2019) Learning and generalization in overparameterized neural networks, going beyond two layers. *Adv. Neural Inform. Processing Systems* 32:6158–6169.
[4] Allen-Zhu Z, Li Y, Song Z (2019) A convergence theory for deep learning via over-parameterization. Chaudhuri K, Salakhutdinov R, eds. *Internat. Conf. Machine Learn.*, vol. 97 (PMLR, New York), 242–252.
[5] Bhandari J, Russo D, Singal R (2018) A finite time analysis of temporal difference learning with linear function approximation. Bubeck S, Perchet, V, Rigollet, P, eds. *Conf. Learn. Theory*, vol. 75 (PMLR, New York), 1691–1692.
[6] Cabannes T, Lauriere M, Perolat J, Marinier R, Girgin S, Perrin S, Pietquin O, Bayen AM, Goubault E, Elie R (2021) Solving N-player dynamic routing games with congestion: A mean-field approach. Preprint, submitted October 22, https://arxiv.org/abs/2110.11943.
[7] Cai Q, Yang Z, Lee JD, Wang Z (2019) Neural temporal-difference learning converges to global optima. *Adv. Neural Inform. Processing Systems* 32:11315–11326.
[8] Calderone D, Sastry SS (2017) Markov decision process routing games. *Internat. Conf. Cyber-Physical Systems* (IEEE, Piscataway, NJ), 273–280.

[9] Cao Y, Yu W, Ren W, Chen G (2012) An overview of recent progress in the study of distributed multi-agent coordination. *IEEE Trans. Indust. Informatics* 9(1):427–438.
[10] Carmona R, Fouque JP, Sun LH (2015) Mean-field games and systemic risk. *Comm. Math. Sci.* 13(4):911–933.
[11] Carmona R, Laurière M, Tan Z (2019) Linear-quadratic mean-field reinforcement learning: Convergence of policy gradient methods. Preprint, submitted October 9, https://arxiv.org/abs/1910.04295.
[12] Carmona R, Laurière M, Tan Z (2023) Model-free mean-field reinforcement learning: Mean-field MDP and mean-field Q-learning. *Ann. Appl. Probab.* 33(6B):5334–5381.
[13] Casgrain P, Jaimungal S (2020) Mean-field games with differing beliefs for algorithmic trading. *Math. Finance* 30(3):995–1034.
[14] Cayci S, Satpathi S, He N, Srikant R (2023) Sample complexity and overparameterization bounds for projection-free neural TD learning. *IEEE Trans. Automatic Control* 68(5):2891–2905.
[15] Chen T, Zhang K, Giannakis GB, Basar T (2022) Communication-efficient policy gradient methods for distributed reinforcement learning. Hennequin PL, ed. *IEEE Trans. Control Network Systems* 9(2):917–929.
[16] Dawson D (1993) Measure-valued Markov processes. *Ecole d'Eté de Probabilités de Saint-Flour. XXI-1991* (Springer, Berlin, Heidelberg), 1–260.
[17] El-Tantawy S, Abdulhai B, Abdelgawad H (2013) Multi-agent reinforcement learning for integrated network of adaptive traffic signal controllers (MARLIN-ATSC): Methodology and large-scale application on downtown Toronto. *IEEE Trans. Intelligent Transportation Systems* 14(3):1140–1150.
[18] Foerster J, Farquhar G, Afouras T, Nardelli N, Whiteson S (2018) Counterfactual multi-agent policy gradients. McIlraith SA, Weinberger KQ, eds. *AAAI Conf. Artificial Intelligence*, vol. 32 (AAAI Press, Palo Alto, CA), 2974–2982.
[19] Fu Z, Yang Z, Wang Z (2020) Single-timescale actor-critic provably finds globally optimal policy. *Internat. Conf. Learn. Representations*.
[20] Gamarnik D (2013) Correlation decay method for decision, optimization, and inference in large-scale networks. *Theory Driven by Influential Applications* (INFORMS, Catonsville, MD), 108–121.
[21] Gamarnik D, Goldberg DA, Weber T (2014) Correlation decay in random decision networks. *Math. Oper. Res.* 39(2):229–261.
[22] Geramifard A, Walsh TJ, Tellex S, Chowdhary G, Roy N, How JP (2013) A tutorial on linear function approximators for dynamic programming and reinforcement learning. *Foundations Trends Machine Learn.* 6(4):375–451.
[23] Germain M, Pham H, Warin X (2023) A level-set approach to the control of state-constrained McKean-Vlasov equations: Application to renewable energy storage and portfolio selection. *Numerical Algebra Control Optim.* 14(3–4):555–582.
[24] Glorot X, Bengio Y (2010) Understanding the difficulty of training deep feedforward neural networks. Teh YW, Titterington M, eds. *Internat. Conf. Artificial Intelligence Statist.*, vol. 8 (PMLR, New York), 249–256.
[25] Gu H, Guo X, Wei X, Xu R (2021) Mean-field controls with Q-learning for cooperative MARL: Convergence and complexity analysis. *SIAM J. Math. Data Sci.* 3(4):1168–1196.
[26] Gu H, Guo X, Wei X, Xu R (2023) Dynamic programming principles for mean-field controls with learning. *Oper. Res.* 71(4):1040–1054.
[27] Guériau M, Dusparic I (2018) SAMoD: Shared autonomous mobility-on-demand using decentralized reinforcement learning. *Internat. Conf. Intelligent Transportation Systems* (IEEE, Piscataway, NJ), 1558–1563.
[28] Guo X, Hu A, Xu R, Zhang J (2019) Learning mean-field games. *Adv. Neural Inform. Processing Systems* 32:4966–4976.
[29] Hu R, Zariphopoulou T (2022) N-player and mean-field games in Itô-diffusion markets with competitive or homophilous interaction. *Stochastic Analysis, Filtering, and Stochastic Optimization: A Commemorative Volume to Honor Mark HA Davis's Contributions* (Springer, Berlin, Heidelberg), 209–237.
[30] Hüttenrauch M, Šošić A, Neumann G (2017) Guided deep reinforcement learning for swarm systems. Preprint, submitted September 18, https://arxiv.org/abs/1709.06011.
[31] Iyer K, Johari R, Sundararajan M (2014) Mean-field equilibria of dynamic auctions with learning. *Management Sci.* 60(12):2949–2970.
[32] Ji Z, Telgarsky M, Xian R (2020) Neural tangent kernels, transportation mappings, and universal approximation. *Internat. Conf. Learn. Representations*.
[33] Jin C, Yang Z, Wang Z, Jordan MI (2020) Provably efficient reinforcement learning with linear function approximation. Abernethy J, Agarwal S, eds. *Conf. Learn. Theory* (PMLR, New York), 2137–2143.
[34] Jin J, Song C, Li H, Gai K, Wang J, Zhang W (2018) Real-time bidding with multi-agent reinforcement learning in display advertising. *ACM Internat. Conf. Inform. Knowledge Management* (ACM, New York), 2193–2201.
[35] Kakade S, Langford J (2002) Approximately optimal approximate reinforcement learning. *Internat. Conf. Machine Learn.* (PMLR, New York), 267–274.
[36] Konda VR, Tsitsiklis JN (2000) Actor-critic algorithms. *Adv. Neural Inform. Processing Systems* 12:1008–1014.
[37] Lacker D, Zariphopoulou T (2019) Mean-field and N-agent games for optimal investment under relative performance criteria. *Math. Finance* 29(4):1003–1038.
[38] Li Y, Tang Y, Zhang R, Li N (2021) Distributed reinforcement learning for decentralized linear quadratic control: A derivative-free policy optimization approach. *IEEE Trans. Automatic Control* 67(12):6429–6444.
[39] Li M, Qin Z, Jiao Y, Yang Y, Wang J, Wang C, Wu G, Ye J (2019) Efficient ridesharing order dispatching with mean-field multi-agent reinforcement learning. *World Wide Web Conf.* (ACM, New York), 983–994.
[40] Lin Y, Qu G, Huang L, Wierman A (2021) Multi-agent reinforcement learning in stochastic networked systems. *Adv. Neural Inform. Processing Systems* 34:7825–7837.
[41] Liu B, Cai Q, Yang Z, Wang Z (2019) Neural trust region/proximal policy optimization attains globally optimal policy. *Adv. Neural Inform. Processing Systems* 32:10565–10576.
[42] Liu Y, Swaminathan A, Agarwal A, Brunskill E (2019) Off-policy policy gradient with stationary distribution correction. Globerson A, Hoffmann AG, eds. *Conf. Uncertainty Artificial Intelligence*, vol. 115 (PMLR, New York), 1180–1190.
[43] Lowe R, Wu YI, Tamar A, Harb J, Pieter Abbeel O, Mordatch I (2017) Multi-agent actor-critic for mixed cooperative-competitive environments. *Adv. Neural Inform. Processing Systems* 30:6382–6393.
[44] Micchelli CA, Xu Y, Zhang H (2006) Universal kernels. *J. Machine Learn. Res.* 7(12):2651–2667.
[45] Motte M, Pham H (2022) Mean-field Markov decision processes with common noise and open-loop controls. *Ann. Appl. Probab.* 32(2):1421–1458.
[46] Pirotta M, Restelli M, Bascetta L (2015) Policy gradient in Lipschitz Markov decision processes. *Machine Learn.* 100(2):255–283.

[47] Qin ZT, Zhu H, Ye J (2022) Reinforcement learning for ridesharing: An extended survey. *Transportation Res. Part C Emerging Tech.* 144:103852.
[48] Qu G, Wierman A, Li N (2020) Scalable reinforcement learning of localized policies for multi-agent networked systems. *Learning for Dynamics and Control*, vol. 120 (PMLR, New York), 256–266.
[49] Rabbat M, Nowak R (2004) Distributed optimization in sensor networks. *Internat. Sympos. Inform. Processing Sensor Networks* (IEEE, Piscataway, NJ), 20–27.
[50] Rahimi A, Recht B (2008) Uniform approximation of functions with random bases. *Annual Allerton Conf. Comm. Control Comput.* (IEEE, Piscataway, NJ), 555–561.
[51] Rashid T, Samvelyan M, Schroeder C, Farquhar G, Foerster J, Whiteson S (2018) QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning. *Internat. Conf. Machine Learn.*, vol. 21(1) (PMLR, New York), 4295–4304.
[52] Shalev-Shwartz S, Shammah S, Shashua A (2016) Safe, multi-agent, reinforcement learning for autonomous driving. Preprint, submitted October 11, https://arxiv.org/abs/1610.03295.
[53] Sra S, Nowozin S, Wright SJ (2012) *Optimization for Machine Learning* (MIT Press, Cambridge, MA).
[54] Sunehag P, Lever G, Gruslys A, Czarnecki WM, Zambaldi V, Jaderberg M, Lanctot M, et al. (2018) Value-decomposition networks for cooperative multi-agent learning based on team reward. Andre E, Koenig S, eds. *Internat. Conf. Autonomous Agents Multi-agent Systems*, vol. 3 (ACM, New York), 2085–2087.
[55] Sutton RS, McAllester DA, Singh SP, Mansour Y (2000) Policy gradient methods for reinforcement learning with function approximation. *Adv. Neural Inform. Processing Systems* 99:1057–1063.
[56] Vadori N, Ganesh S, Reddy P, Veloso M (2020) Calibration of shared equilibria in general sum partially observable Markov games. *Adv. Neural Inform. Processing Systems* 33:14118–14128.
[57] Wang L, Cai Q, Yang Z, Wang Z (2020) Neural policy gradient methods: Global optimality and rates of convergence. *Internat. Conf. Learn. Representations* (ICLR, Appleton, WI).
[58] Xu P, Gao F, Gu Q (2019) Sample efficient policy gradient methods with recursive variance reduction. *Internat. Conf. Learn. Representations* (ICLR, Appleton, WI).
[59] Xu P, Gao F, Gu Q (2020) An improved convergence analysis of stochastic variance-reduced policy gradient. Adams RP, Gogate V, eds. *Conf. Uncertainty Artificial Intelligence*, vol. 115 (PMLR, New York), 541–551.
[60] Yang Y, Wen Y, Wang J, Chen L, Shao K, Mguni D, Zhang W (2020) Multi-agent determinantal Q-learning. *Internat. Conf. Machine Learn.* (PMLR, New York), 10757–10766.
[61] Yang Y, Hao J, Chen G, Tang H, Chen Y, Hu Y, Fan C, Wei Z (2020) Q-value path decomposition for deep multiagent reinforcement learning. Daumé H, Singh A, eds. *Internat. Conf. Machine Learn.* (PMLR, New York), 10706–10715.
[62] You X, Li X, Xu Y, Feng H, Zhao J, Yan H (2020) Toward packet routing with fully distributed multiagent deep reinforcement learning. *IEEE Trans. Systems Man Cybernetics Systems* 52(2):855–868.
[63] Zhang K, Yang Z, Basar T (2018) Networked multi-agent reinforcement learning in continuous spaces. *Conf. Decision Control* (IEEE, Piscataway, NJ), 2771–2776.
[64] Zhang K, Yang Z, Başar T (2021) Multi-agent reinforcement learning: A selective overview of theories and algorithms. *Handbook of Reinforcement Learning and Control*, Chapter 12 (Springer, Cham, Switzerland), 321–384.
[65] Zhang K, Koppel A, Zhu H, Basar T (2020) Global convergence of policy gradient methods to (almost) locally optimal policies. *SIAM J. Control Optim.* 58(6):3586–3612.
[66] Zhang K, Liu Y, Liu J, Liu M, Başar T (2020) Distributed learning of average belief over networks using sequential observations. *Automatica J. IFAC* 115:108857.
[67] Zhang K, Yang Z, Liu H, Zhang T, Basar T (2018) Fully decentralized multi-agent reinforcement learning with networked agents. Dy J, Krause A, eds. *Internat. Conf. Machine Learn.* (PMLR, New York), 5872–5881.
[68] Zhang K, Yang Z, Liu H, Zhang T, Basar T (2021) Finite-sample analysis for decentralized batch multi-agent reinforcement learning with networked agents. *IEEE Trans. Automatic Control* 66(12):5925–5940.
[69] Zheng S, Trott A, Srinivasa S, Naik N, Gruesbeck M, Parkes DC, Socher R (2020) The AI economist: Improving equality and productivity with AI-driven tax policies. Preprint, submitted April 28, https://arxiv.org/abs/2004.13332.
[70] Zhou Z, Mertikopoulos P, Moustakas AL, Bambos N, Glynn P (2021) Robust power management via learning and game design. *Oper. Res.* 69(1):331–345.
[71] Zou D, Gu Q (2019) An improved analysis of training over-parameterized deep neural networks. *Adv. Neural Inform. Processing Systems* 32:2055–2064.