# Q-Learning Algorithm for Mean-Field Controls, with Convergence and Complexity Analysis

Haotian Gu * Xin Guo * Xiaoli Wei * Renyuan Xu †

July 9, 2020

**Abstract**

This paper studies multi-agent reinforcement learning (MARL) collaborative games under a mean-field control (MFC) approximation framework. It develops a model-free kernel-based Q-learning algorithm (MFC-K-Q) on a probability measure space, and shows that the convergence rate and the sample complexity of MFC-K-Q are independent of the number of agents $N$. Empirical studies on the network traffic congestion problem demonstrate that MFC-K-Q outperforms existing MARL algorithms (when $N$ is large) and MFC algorithms.

## 1 Introduction

Multi-agent reinforcement learning (MARL) has enjoyed substantial successes for analyzing the otherwise challenging collaborative games, including two-agent or two-team computer games [25, 28], self-driving vehicles [24], real-time bidding games [13], ride-sharing [17], and traffic routing [6]. Despite its empirical success, MARL suffers from the curse of dimensionality known also as the *combinatorial nature* of MARL [11]: its sample complexity by existing algorithms for stochastic dynamics grows exponentially with respect to the number of agents $N$. Indeed, one can show that this complexity is in the order of $\Omega((|\mathcal{X}||\mathcal{U}|)^N \text{poly}(N))$, with $\mathcal{X}$ and $\mathcal{U}$ the state and action spaces of the dynamic system, as to be seen in Proposition 2.1. In practice, this $N$ could be on the scale of thousands or more, for instance, in the cases of rider match-up for Uber-pool and network routing for Zoom.

One classical approach to resolve this curse of dimensionality is to focus on *local policies*, namely by exploiting special structures of MARL problems and by designing problem-dependent algorithms to reduce the complexity. For instance, [16] developed value-based distributed Q-learning algorithm for deterministic and finite Markov decision problems (MDPs), and [22] exploited special dependence structures among agents. (See the review by [30] and the references therein).

Another approach, largely unexplored, is to consider the MARL in the regime with a large number of homogeneous agents. In this paradigm, by the propagation of chaos documented

*Department of Industrial Engineering & Operations Research, University of California, Berkeley, USA. **Email:** {haotian_gu, xinguo, xiaoliwei }@berkeley.edu

†Mathematical Institute, University of Oxford, UK. **Email:** xur@maths.ox.ac.uk

in [14, 19, 26, 9] MARL becomes a learning problem for mean-field controls (MFC). This approach is appealing not only because the dimension of MFC is independent of the number of agents $N$, but also because MFC has shown, for example in [15, 20], to approximate the corresponding $N$-agent collaborative game in terms of both game values and optimal strategies.

However, learning MFC, i.e., analyzing MFC problems with unknown transition dynamics and unknown reward functions, requires simultaneously *controlling* and *learning* the MFC system. This is by and large an uncharted field. Its most recent theoretical development is [10, 20], which established the Dynamic Programming Principle for both the Q function and the value function for learning MFC. Their basic idea is to retrofit learning MFC problem within a MDP framework, with both the state space $\mathcal{X}$ and the action space $\mathcal{U}$ lifted to their respective probability measure spaces.

The key issue is, as pointed out in [20], there are no efficient RL algorithms available on probability measure spaces. Instead, [4] proposed a different approach for learning MFC by adding common noises to the underlying dynamics. This approach enables direct application of existing theory of learning MDP with stochastic dynamics. However, this first model-free algorithm for learning MFC suffers from high sample complexity (see Table 1 below) with weak performance (as demonstrated in Section 5). For special classes of linear-quadratic MFCs with stochastic dynamics [3] explored the policy gradient method and [18] developed an actor-critic type algorithm.

**Our work.** The paper proposes an efficient approximation algorithm (MFC-K-Q) for learning MFC. This model-free Q-learning-based algorithm combines the technique of kernel regression with approximated Bellman operator. The convergence rate and the sample complexity of this algorithm are shown to be independent of the number of agents $N$, and rely only on the size of the state-action space of the underlying single-agent MDP (Table 1).

Our kernel regression idea is inspired by [23]. Nevertheless, our problem setting and technique for error bound analysis are different from theirs: the error control in [23] was obtained via martingale concentration inequalities whereas ours is by the regularity property of the underlying dynamics.

Our experiment in Section 5 demonstrates that MFC-K-Q avoids the curse of dimensionality and outperforms existing MARL (when $N$ is large) and the MFC algorithm in [4]. Table 1 summarizes the complexity of our MFC-K-Q algorithm along with these relevant algorithms.

Table 1: Comparison of algorithms

| Work | MFC/N-player | Method | Sample Complexity Guarantee |
|---|---|---|---|
| Our work | MFC | Q-learning | $\Omega(T_{cov} \cdot \log(1/\delta))$ |
| [4] | MFC | Q-learning | $\Omega((T_{cov} \cdot \log(1/\delta))^l \cdot \text{poly}(\log(1/(\delta\epsilon))/\epsilon))$ |
| Vanilla N-player | N-player | Q-learning | $\Omega(\text{poly}((\lvert\mathcal{X}\rvert\lvert\mathcal{U}\rvert)^N \cdot \log(1/(\delta\epsilon)) \cdot N/\epsilon))$ |
| [22] | N-player | Actor-critic | $\Omega(\text{poly}((\lvert\mathcal{X}\rvert\lvert\mathcal{U}\rvert)^{f(\log(1/\epsilon))} \cdot \log(1/\delta) \cdot N/\epsilon))$ |

Here $T_{cov}$ is the covering time of the exploration policy and $l = \max\{3+1/\kappa, 1/(1-\kappa)\} > 4$ for some $\kappa \in (0.5, 1)$. Other parameters are as in Proposition 2.1 and also in Theorem

3.1. Note that [22] assumed that agents interact locally through a given graph so that local policies can approximate the global one, yet $f(\log(1/\epsilon))$ can scale as $N$ for a dense graph.

**Organizations.** Section 2 connects the collaborative MARL and the problem of learning MFC. Section 3 proposes the algorithm (MFC-K-Q) for MFC, with convergence and sample complexity analysis. Section 4 is dedicated to the proof of the main theorem. Finally, Section 5 tests performance of MFC-K-Q in a network congestion control example.

# 2 Problem Set-up

## 2.1 Collaborative Multi-Agent Reinforcement Learning (MARL)

First recall the $N$-agent collaborative game, where there are $N$ agents whose game strategies are coordinated by a central controller. At each step $t$, the state of player $j$ $(= 1, 2, \cdots, N)$ is $x_t^j \in \mathcal{X}$ and she takes an action $u_t^j \in \mathcal{U}$. Here $\mathcal{X}$ and $\mathcal{U}$ are finite state space and action space, respectively. Given the current state profile $\boldsymbol{x}_t = (x_t^1, \cdots, x_t^N) \in \mathcal{X}^N$ and the current action profile $\boldsymbol{u}_t = (u_t^1, \cdots, u_t^N) \in \mathcal{U}^N$ of $N$-agents, player $j$ will receive a reward $\tilde{r}^j(\boldsymbol{x}_t, \boldsymbol{u}_t)$ and her state will change to $x_{t+1}^j$ according to a transition probability function $P^j(\boldsymbol{x}_t, \boldsymbol{u}_t)$. A Markovian game further restricts the admissible policy/control for player $j$ to be of the form $u_t^j \sim \pi_t^j(\boldsymbol{x}_t)$. That is, $\pi_t^j : \mathcal{X}^N \to \mathcal{P}(\mathcal{U})$ maps each state profile $\boldsymbol{x} \in \mathcal{X}^N$ to a randomized action, with $\mathcal{P}(\mathcal{U})$ the probability measure space on space $\mathcal{U}$. The accumulated reward for agent $j$, under the initial state profile $\boldsymbol{x}_0 = \boldsymbol{x}$ and policy $\boldsymbol{\pi} = \{\boldsymbol{\pi}_t\}_{t=0}^{\infty}$ with $\boldsymbol{\pi}_t = (\pi_t^1, \ldots, \pi_t^N)$, is then defined as

$$J^j(\boldsymbol{x}, \boldsymbol{\pi}) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \tilde{r}^j(\boldsymbol{x}_t, \boldsymbol{u}_t) \big| \boldsymbol{x}_0 = \boldsymbol{x}\right],$$

where $\gamma \in (0, 1)$ is a discounted factor, $u_t^j \sim \pi_t^j(\boldsymbol{x}_t)$, and $x_{t+1}^j \sim P^j(\boldsymbol{x}_t, \boldsymbol{u}_t)$. In collaborative MARL, the central controller is to maximize the aggregated accumulated rewards over all policies, i.e., to find $\sup_{\boldsymbol{\pi}} \frac{1}{N} \sum_{j=1}^{N} J^j(\boldsymbol{x}, \boldsymbol{\pi})$. Now, take Theorem 4 in [7] and note that the corresponding covering time for the policy of the central controller will be at least $(|\mathcal{X}||\mathcal{U}|)^N$, then clearly the sample complexity for the Q learning algorithm of this game grows exponentially in $N$. That is,

**Proposition 2.1** *Let $|\mathcal{X}|$ and $|\mathcal{U}|$ be respectively the size of state space and action space. Let $Q^*$ and $Q_T$ be respectively the optimal value and the value of the asynchronous Q-learning algorithm in [7] using polynomial learning rate at time $T$. Then with probability at least $1-\delta$, the sup distance between $Q_T$ and $Q^*$ is $||Q_T - Q^*||_\infty \leq \epsilon$, with $T = \Omega\Big(poly\Big((|\mathcal{X}||\mathcal{U}|)^N \cdot \frac{N}{\epsilon} \cdot \ln(\frac{1}{\delta\epsilon})\Big)\Big)$.*

## 2.2 Learning MFC and Bellman Equation for Q Function

To overcome the curse of dimensionality in $N$, consider a mean-field approximation where all agents are assumed to be identical, indistinguishable, and interchangeable. Each agent

$j$ depends on all other agents only through the empirical distribution of their states $\mu_t^N = \frac{1}{N}\sum_{j=1}^N \delta_{x_t^j} \in \mathcal{P}(\mathcal{X})$ and the empirical distribution of their actions $\nu_t^N = \frac{1}{N}\sum_{j=1}^N \delta_{u_t^j} \in \mathcal{P}(\mathcal{U})$, where $\mathcal{P}(\mathcal{X})$ is the probability measure space on space $\mathcal{X}$. By the law of large numbers, this N-player collaborative game then becomes an MFC problem when $N \to \infty$. See [2] for more background. Due to indistinguishability of the agents, one can focus on a single representative agent who interacts with the population distribution. That is, at each time $t$, the representative agent in state $x_t$ takes an action $u_t \in \mathcal{U}$ according to the admissible policy $\pi_t(x_t, \mu_t) : \mathcal{X} \times \mathcal{P}(\mathcal{X}) \to \mathcal{P}(\mathcal{U})$ assigned by the central controller, who can observe the population state distribution $\mu_t \in \mathcal{P}(\mathcal{X})$. The agent will then receive a reward $\tilde{r}(x_t, \mu_t, u_t, \nu_t)$ and move to the next state $x_{t+1} \in \mathcal{X}$ according to a probability transition function $P(x_t, \mu_t, u_t, \nu_t)$, where $P$ and $\tilde{r}$, relying on the distribution of the state $\mu_t$ and the action $\nu_t(\cdot) := \sum_{x\in\mathcal{X}} \pi_t(x, \mu_t)(\cdot)\mu_t(x)$, are possibly unknown.

The central controller aims to maximize the accumulated reward over all admissible policies $\pi = \{\pi_t\}_{t=0}^{\infty}$, i.e.,

$$\sup_{\pi} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \tilde{r}(x_t, \mu_t, u_t, \nu_t) \,\middle|\, x_0 \sim \mu\right], \tag{2.1}$$

subject to

$$x_{t+1} \sim P(x_t, \mu_t, u_t, \nu_t), \quad u_t \sim \pi_t(x_t, \mu_t). \tag{2.2}$$

According to [10], the above learning MFC problem can be recast as a general MDP problem with probability measure space as the new state-action space. The idea behind [10] is to lift the finite state-action space $\mathcal{X}$ and $\mathcal{U}$ to a compact continuous state-action space embedded in Euclidean space $\mathcal{C} := \mathcal{P}(\mathcal{X}) \times \mathcal{H}$ with $\mathcal{H} := \{h : \mathcal{X} \mapsto \mathcal{P}(\mathcal{U})\}$, such that the dynamics become *deterministic* by the aggregation over the original state-action space. Moreover, according to [10, 20], the associated optimal Q function for this MFC problem (2.1)-(2.2) starting from arbitrary $(\mu, h) \in \mathcal{C}$ is

$$Q_{\mathcal{C}}(\mu, h) = \sup_{\pi} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \tilde{r}(x_t, \mu_t, u_t, \nu_t) \,\middle|\, x_0 \sim \mu, u_0 \sim h, u_t \sim \pi_t\right]. \tag{2.3}$$

**Proposition 2.2** *The Bellman equation for $Q_{\mathcal{C}} : \mathcal{C} \to \mathbb{R}$ is*

$$Q_{\mathcal{C}}(\mu, h) = r(\mu, h) + \gamma \sup_{\tilde{h}\in\mathcal{H}} Q_{\mathcal{C}}(\Phi(\mu, h), \tilde{h}). \tag{2.4}$$

*where $r$ and $\Phi$ are respectively aggregated reward and dynamics such that*

$$r(\mu, h) = \sum_{x\in\mathcal{X}} \sum_{u\in\mathcal{U}} \tilde{r}(x, \mu, u, \nu(\mu, h))\mu(x)h(x)(u),$$
$$\Phi(\mu, h) = \sum_{x\in\mathcal{X}} \sum_{u\in\mathcal{U}} P(x, \mu, u, \nu(\mu, h))\mu(x)h(x)(u),$$

*with $\nu(\mu, h)(\cdot) := \sum_{x\in\mathcal{X}} h(x)(\cdot)\mu(x)$. Moreover, $\mathcal{H}$ is the* **minimum** *space under which the Bellman equation (2.4) holds.*

# 3 MFC-K-Q Algorithm, Convergence, and Complexity

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

In this section, we first develop a kernel-based Q-learning algorithm (MFC-K-Q) for learning MFC based on (2.4), and then analyze its convergence and sample complexity.

To start, note that the lifted state space $\mathcal{P}(\mathcal{X})$ is the probability simplex in $\mathbb{R}^{|\mathcal{X}|}$, and the lifted action space $\mathcal{H}$ is the product of $|\mathcal{X}|$ copies of the probability simplex in $\mathbb{R}^{|\mathcal{U}|}$. Both lifted spaces are continuous and embedded in finite-dimensional Euclidean spaces. Take the $l_1$ distance as the metric for $\mathcal{P}(\mathcal{X})$ and $\mathcal{P}(\mathcal{U})$ and define metrics on $\mathcal{H}$ and $\mathcal{C}$ to be $d_{\mathcal{H}}(h_1, h_2) = \max_{x\in\mathcal{X}} ||h_1(x) - h_2(x)||_1$ and $d_{\mathcal{C}}((\mu_1, h_1), (\mu_2, h_2)) = ||\mu_1 - \mu_2||_1 + d_{\mathcal{H}}(h_1, h_2)$.

Our algorithm consists of two steps of approximations. The first step is to develop a kernel regression method on the discretized lifted measure spaces.

Kernel regression is a local averaging approach for approximating *unknown* state-action pair from *observed* data: $\epsilon$-net is its key building block on which the lifted space is discretized, and the choice of $\epsilon$ is critical for the convergence and the sample complexity analysis.

First note that compactness of $\mathcal{C}$ implies the existence of an $\epsilon$-net $\mathcal{C}_\epsilon$, consisting of $\{c^i = (\mu^i, h^i)\}_{i=1}^{N_\epsilon}$ if $\min_{1\le i\le N_\epsilon} d_{\mathcal{C}}(c, c^i) < \epsilon$ for all $c \in \mathcal{C}$. Denote $\mathcal{H}_\epsilon$ as an $\epsilon$-net on $\mathcal{H}$ induced from $\mathcal{C}_\epsilon$, i.e., $\mathcal{H}_\epsilon$ contains all the possible action choices in $\mathcal{C}_\epsilon$, whose size is denoted by $\mathcal{N}_{\epsilon,\mathcal{H}}$. Denote $\mathbb{R}^{\mathcal{C}_\epsilon}$ and $\mathbb{R}^{\mathcal{C}}$ as the sets of all bounded functions on $\mathcal{C}_\epsilon$ and $\mathcal{C}$, respectively. Then define the so-called *kernel regression operator* $\Gamma_K : \mathbb{R}^{\mathcal{C}_\epsilon} \to \mathbb{R}^{\mathcal{C}}$ such that

$$\Gamma_K f(c) = \sum_{i=1}^{N_\epsilon} K(c^i, c) f(c^i), \tag{3.5}$$

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

$$\sum_{i=1}^{N_\epsilon} K(c^i, c) = 1, \text{ and } K(c^i, c) = 0 \text{ if } d_{\mathcal{C}}(c^i, c) > \epsilon. \tag{3.6}$$

In fact, $K$ can be of any form

$$K(c^i, c) = \frac{\phi(c^i, c)}{\sum_{i=1}^{N_\epsilon} \phi(c^i, c)}, \tag{3.7}$$

with some function $\phi$ satisfying $\phi \ge 0$ and $\phi(x, y) = 0$ when $d_{\mathcal{C}}(x, y) \ge \epsilon$. (See Section 5 for some choices of $\phi$).

The second step of the algorithm is to approximate the optimal Q function in (2.3). Instead of maximizing over $\mathcal{H}$ as in the Bellman equation (2.4), we take the maximum over the $\epsilon$-net $\mathcal{H}_\epsilon$ on the action space. Since $(\Phi(c^i), \tilde{h})$ may not be on the $\epsilon$-net, we need to approximate the value at that point via the kernel regression $\Gamma_K q(\Phi(c^i), \tilde{h})$. That is to introduce an approximated Bellman operator $B_K$ acting on functions defined on the $\epsilon-$net $\mathcal{C}_\epsilon$: $\mathbb{R}^{\mathcal{C}_\epsilon} \to \mathbb{R}^{\mathcal{C}_\epsilon}$ such that

$$(B_K\, q)(c^i) = r(c^i) + \gamma \max_{\tilde{h}\in\mathcal{H}_\epsilon} \Gamma_K q(\Phi(c^i), \tilde{h}). \tag{3.8}$$

In practice, we may only have access to noisy estimations $\{\hat{r}(c^i), \hat{\Phi}(c^i)\}_{i=1}^{N_\epsilon}$ instead of the accurate data $\{r(c^i), \Phi(c^i)\}_{i=1}^{N_\epsilon}$ on $\mathcal{C}_\epsilon$. Taking this into consideration, Algorithm 1 consists of two steps. First, it collects samples on $\mathcal{C}$ given an exploration policy. For each component $c^i$ on the $\epsilon$-net $\mathcal{C}_\epsilon$, the estimated data $(\hat{r}(c^i), \hat{\Phi}(c^i))$ is computed by averaging samples in the $\epsilon$-neighborhood of $c^i$. Second, the fixed point iteration is applied to the approximated Bellman operator $B_K$ with $\{\hat{r}(c^i), \hat{\Phi}(c^i)\}_{i=1}^{N_\epsilon}$. Under appropriate conditions, Algorithm 1 provides an accurate estimation of the true Q function. (See Theorem 3.1).

**Algorithm 1 Kernel-based Q-learning Algorithm for MFC (MFC-K-Q)**

**Input**: Initial state distribution $\mu_0$, $\epsilon > 0$, $\epsilon$-net on $\mathcal{C}$ : $\mathcal{C}_\epsilon = \{c^i = (\mu^i, h^i)\}_{i=1}^{N_\epsilon}$, exploration policy $\pi$ taking actions from $\mathcal{H}_\epsilon$ induced from $\mathcal{C}_\epsilon$, regression kernel $K$ on $\mathcal{C}_\epsilon$.

**Initialize**: $\hat{r}(c^i) = 0$, $\hat{\Phi}(c^i) = 0$, $N(c^i) = 0, \forall i$.
**repeat**
  At the current state distribution $\mu_t$, act $h_t$ according to $\pi$, observe $\mu_{t+1} = \Phi(\mu_t, h_t)$ and $r_t = r(\mu_t, h_t)$.
  **for** $1 \le i \le N_\epsilon$ **do**
    **if** $d_{\mathcal{C}}(c^i, (\mu_t, h_t)) < \epsilon$ **then**
      $N(c^i) \leftarrow N(c^i) + 1$.
      $\hat{r}(c^i) \leftarrow \frac{N(c^i)-1}{N(c^i)} \cdot \hat{r}(c^i) + \frac{1}{N(c^i)} \cdot r_t$
      $\hat{\Phi}(c^i) \leftarrow \frac{N(c^i)-1}{N(c^i)} \cdot \hat{\Phi}(c^i) + \frac{1}{N(c^i)} \cdot \mu_t$
    **end if**
  **end for**
**until** $N(c^i) > 0, \forall i$.
**Initialize**: $\hat{q}_0(c^i) = 0, \forall c^i \in \mathcal{C}_\epsilon$, $l = 0$.
**repeat**
  **for** $c^i \in \mathcal{C}_\epsilon$ **do**
    $\hat{q}_{l+1}(c^i) \leftarrow \Big(\hat{r}(c^i) + \gamma \max_{\tilde{h} \in \mathcal{H}_\epsilon} \Gamma_K \hat{q}_l(\hat{\Phi}(c^i), \tilde{h})\Big)$.
  **end for**
  $l = l + 1$.
**until** converge

## 3.2 Convergence and Sample Complexity Analysis of MFC-K-Q

Note that unlike learning algorithms for stochastic dynamics where the choice of learning rate $\eta_t$ is to guarantee the convergence of $Q$, MFC-K-Q directly conducts the fixed point iteration for the approximated Bellman operator $B_K$ on the sampled data set, and sets the learning rate as 1, to take full advantage of the deterministic dynamics. Consequently, the complexity analysis of this algorithm is reduced significantly, again because of the deterministic systems for which it suffices to visit the $\epsilon$-neighborhood of each component in the $\epsilon$-net only **once**. By comparison, for stochastic systems each component in the $\epsilon$-net has to be visited sufficiently many times for a decent estimate in Q-learning.

The convergence and sample complexity analysis for this proposed MFC-K-Q algorithm is based on several assumptions.

**Assumption 3.1 (Continuity and boundedness of $\tilde{r}$)** *There exists $\tilde{R} > 0, L_{\tilde{r}} > 0$, such that for all $x \in \mathcal{X}, u \in \mathcal{U}$, $\mu_1, \mu_2 \in \mathcal{P}(\mathcal{X}), \nu_1, \nu_2 \in \mathcal{P}(\mathcal{U})$,*

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

**Assumption 3.2 (Continuity of $P$)** *There exists $L_P > 0$ such that for all $x \in \mathcal{X}, u \in \mathcal{U}, \mu_1, \mu_2 \in \mathcal{P}(\mathcal{X}), \nu_1, \nu_2 \in \mathcal{P}(\mathcal{U})$, $||P(x, \mu_1, u, \nu_1) - P(x, \mu_2, u, \nu_2)||_1 \leq L_P \cdot (||\mu_1 - \mu_2||_1 + ||\nu_1 - \nu_2||_1)$.*

**Assumption 3.3 (Controllability of the dynamics)** *For all $\epsilon$, there exists $M_\epsilon \in \mathbb{N}$ such that for any $\epsilon$-net $\mathcal{H}_\epsilon$ on $\mathcal{H}$ and $\mu, \mu' \in \mathcal{P}(\mathcal{X})$, there exists an action sequence $(h^1, \ldots . h^m), h^i \in \mathcal{H}_\epsilon, m < M_\epsilon$, that will drive the state from $\mu$ to an $\epsilon$-neighborhood of $\mu'$ by taking $(h^1, \ldots . h^m)$.*

**Assumption 3.4 (Regularity of kernels)** *For any point $c \in \mathcal{C}$, there exist at most $N_K$ points $c^i$'s in $\mathcal{C}_\epsilon$ such that $K(c^i, c) > 0$. Moreover, there exists $L_K > 0$, such that for all $c \in \mathcal{C}_\epsilon, c', c'' \in \mathcal{C}, |K(c, c') - K(c, c'')| \leq L_K \cdot d_{\mathcal{C}}(c', c'')$.*

Assumption 3.1 and 3.2 are standard regularity assumptions for the MFC problems [2]. Assumption 3.3 ensures the dynamics to be controllable. (See Section 4.3 for a detailed example for Assumption 3.3). Assumption 3.4 is easy to be satisfied: take a uniform grid as the $\epsilon-$net, then $N_K$ is roughly bounded from above by $2^{\dim(\mathcal{C})}$; meanwhile, many commonly used kernels, including the triangular kernel in Section 5, satisfy the Lipschitz assumption.

**Theorem 3.1** *Assume Assumptions 3.1, 3.2, 3.3, 3.4 and assume $\gamma \cdot (2L_P + 1) < 1$. For any $\epsilon' > 0$, under the $\epsilon'$-greedy policy, with probability $1 - \delta$, for any initial state distribution $\mu$, after $\frac{(M_\epsilon+1)\cdot(N_{\epsilon,\mathcal{H}})^{M_\epsilon+1}}{(\epsilon')^{M_\epsilon+1}} \cdot \log(N_\epsilon) \cdot e \cdot \log(1/\delta)$ samples, Algorithm 1 converges linearly to some function $\hat{Q}_{\mathcal{C}_\epsilon}$; and the sup distance between $\Gamma_K \hat{Q}_{\mathcal{C}_\epsilon}$ in (3.5) and $Q_{\mathcal{C}}$ in (2.3) is*

$$||\Gamma_K \hat{Q}_{\mathcal{C}_\epsilon} - Q_{\mathcal{C}}||_\infty \leq \frac{(1-\gamma)(\tilde{R} + 2L_{\tilde{r}}) + 2\gamma N_K L_K \tilde{R}(2L_P + 1)}{(1-\gamma)^2} \cdot \epsilon + \frac{2\tilde{R} + 4L_{\tilde{r}}}{(1 - \gamma \cdot (2L_P + 1))(1-\gamma)} \cdot \epsilon,$$

*where $\log(N_\epsilon) = \Theta(|\mathcal{X}||\mathcal{U}| \log(1/\epsilon))$, and $N_{\epsilon,\mathcal{H}} = \Theta((\frac{1}{\epsilon})^{(|\mathcal{U}|-1)|\mathcal{X}|})$.*

Theorem 3.1 shows that the sample complexity for learning MFC is $\Omega(\text{poly}((1/\epsilon) \cdot \log(1/\delta)))$, instead of the exponential rate in $N$ by existing algorithms for N-agent cooperative games in Proposition 2.1.

# 4 Proofs of Theorems

As stated in Proposition 2.2, learning MFC can be reformulated in a general MDP framework with continuous state-action space and deterministic dynamics, by lifting the finite state-action space $\mathcal{X}$ and $\mathcal{U}$ to a compact continuous state-action space $\mathcal{P}(\mathcal{X})$ and $\mathcal{H} = \{h : \mathcal{X} \mapsto \mathcal{P}(\mathcal{U})\}$.

Based on this idea, this section is organized as follows:

- First, we consider a general MDP set-up with continuous state-action space and deterministic dynamics, denoted as **MDP-CDD**, in Section 4.1; we propose a kernel-based Q-learning algorithm CDD-K-Q for this **MDP-CDD** problem. This CDD-K-Q algorithm corresponds to the MFC-K-Q algorithm for learning MFC problem.

- We then provide the convergence and sample complexity analysis of CDD-K-Q algorithm in Section 4.2 and Section 4.3, respectively. Their proofs can be found in Section 4.4.1 and Section 4.4.2, respectively.

- Finally, we apply the general theory for **MDP-CDD** to learning MFC problem in Section 4.4.3, with the complete proof of our main result Theorem 3.1.

## 4.1 MDP framework with continuous state-action space and deterministic dynamics (MDP-CDD)

Let $\mathcal{S}$ (resp. $\mathcal{A}$) be continuous state (resp. action) space which is a complete compact metric space with metric $d_{\mathcal{S}}$ (resp. $d_{\mathcal{A}}$). Let $\mathcal{C} := \mathcal{S} \times \mathcal{A}$ be a complete metric space with the metric given by

$$d_{\mathcal{C}}(c, c') = d_{\mathcal{S}}(s, s') + d_{\mathcal{A}}(a, a'), \text{ with } c = (s, a) \text{ and } c' = (s', a'). \tag{4.9}$$

At time $t$, let $s_t \in \mathcal{S}$ be the state of the representative agent. Once the agent takes the action $a_t \in \mathcal{A}$ according to a policy $\pi$, the agent moves to the next state $s_{t+1}$ according to the deterministic dynamics $s_{t+1} = \Phi(s_t, a_t)$ and receives an immediate reward $r(s_t, a_t)$. Here the policy $\pi = \{\pi_t\}_{t=0}^{\infty}$ is Markovian so that at each stage $t$, $\pi_t : \mathcal{S} \to \mathcal{P}(\mathcal{A})$ maps the state $s_t$ to $\pi_t(s_t)$, a distribution over the action space.

The agent's objective is to maximize the expected cumulative reward starting from an arbitrary state $s \in \mathcal{S}$,

$$V_{\mathcal{C}}(s) = \sup_{\pi} V_{\mathcal{C}}^{\pi}(s) := \sup_{\pi} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \Big| s_0 = s\right], \tag{4.10}$$

as well as to maximize the expected cumulative reward starting from arbitrary state-action pair $(s, a) \in \mathcal{C}$

$$Q_{\mathcal{C}}(s, a) = \sup_{\pi} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \Big| s_0 = s, a_0 = a\right]. \tag{4.11}$$

We will call this problem **MDP-CDD**. By Theorem 2.2, our learning MFC is a special case of MDP-CDD with $\mathcal{S} = \mathcal{P}(\mathcal{X})$, $\mathcal{A} = \mathcal{H}$, $\Phi$ and $r$ defined in Theorem 2.2. Meanwhile, for the general MDP-CDD problems, the kernel operator $\Gamma_K$ and the approximated Bellman operator $B_K$ can be defined in the same way, on $\mathcal{S}$ and $\mathcal{A}$, as in (3.5) and (3.8). Algorithm 1 can also be viewed as a general algorithm for solving any MDP-CDD problem. For clarity, we rewrite the algorithm in its general formulation as Algorithm 2.

Therefore, in the following discussion, we will give the convergence and sample complexity results for Algorithm 2, and apply those results back to the MFC setting, to complete the proof for Theorem 3.1.

## 4.2 Convergence of CDD-K-Q Algorithm

First, we start with some assumptions.

**Assumption 4.5 (Continuity of $\Phi$)** *There exists $L_\Phi > 0$, such that for all $c, c' \in \mathcal{C}$, $d_{\mathcal{S}}(\Phi(c), \Phi(c')) \leq L_\Phi d_{\mathcal{C}}(c, c')$.*

**Assumption 4.6 (Continuity and boundedness of $r$)** *There exists $L_r, R > 0$, such that for all $c, c' \in \mathcal{C}$, $|r(c) - r(c')| \leq L_r d_{\mathcal{C}}(c, c'), |r(c)| \leq R$.*

**Assumption 4.7 (Discounted factor $\gamma$)** $\gamma \cdot L_\Phi < 1$.

Assumptions 4.5 and 4.6 are standard for deterministic dynamics (see [1] and [29]). The essence of the assumptions is to guarantee that $Q$ or $V$ is Lipschitz continuous (shown in Theorem 4.3), in order to establish the convergence and bounds on sample complexities of the algorithm. If Assumption 4.7 fails, $V$ may not be Lipschitz, as shown below.

**Example.** Let $S = \mathbb{R}$, $A$ is a singleton set. The dynamic $\Phi(s) = 10s$. The reward $r(s) = 1$ when $s > 1$; $r(s) = -1$ when $s < -1$; $r(s) = s$ otherwise. $\gamma = 0.5$. In this case, one can compute directly that the value function is $V(0) = 0$, and $V(10^{-k}) \geq 2^{-k}$, which is not Lipschitz.

Indeed, the approximated Bellman operator $B_K$ contains two layers of approximations and the approximation error may propagate during the iteration. Therefore, the Lipschitz continuity of $V$ or $Q$ is to control the error propagation. In the case of stochastic dynamics, the Lipschitz continuity of $Q$ or $V$ is either assumed directly [29] or guaranteed with sufficient regularity of the transition kernel [23].

**Theorem 4.2** *Given Assumptions 4.5, 4.6, 4.7, $B_K$ (3.8) has a unique fixed point $Q_{\mathcal{C}_\epsilon}$ in $\{f \in \mathbb{R}^{\mathcal{C}_\epsilon} : \|f\|_\infty \leq V_{\max}\}$, and $B$ has a unique fixed point $Q_{\mathcal{C}}$ in $\{f \in \mathbb{R}^{\mathcal{C}} : \|f\|_\infty \leq V_{\max}\}$ with $V_{\max} := \frac{R}{1-\gamma}$. Moreover, the sup distance between $\Gamma_K Q_{\mathcal{C}_\epsilon}$ in (3.5) and $Q_{\mathcal{C}}$ in (4.11) is*

$$||\Gamma_K Q_{\mathcal{C}_\epsilon} - Q_{\mathcal{C}}||_\infty \leq \frac{(1+\gamma)L_r}{(1-\gamma L_\Phi)(1-\gamma)} \cdot \epsilon. \tag{4.12}$$

Theorem 4.2 shows that the error introduced by $B_K$ (3.8) is proportional to the size of the $\epsilon$-net; it suggests that as long as one has enough samples to form an $\epsilon$-net, one can run the fixed point iteration defined by $B_K$ (3.8) to get $Q_{\mathcal{C}_\epsilon}$, which is shown to be an accurate estimation for the true $Q_{\mathcal{C}}$. However, in practice, instead of the accurate data on a predefined $\epsilon$-net, we may only have access to their noisy estimates. For example, in Algorithm 2, we get samples from the neighborhood of each component on the $\epsilon$-net, and estimate the data by averaging samples. In this case, another source of error is introduced, namely, the estimation error based on samples. To control this error, we need some mild assumptions on kernels, which is Assumption 3.4. In fact, it can be proved that the error bound of Algorithm 2 is still linear in $\epsilon$ and the convergence rate is also linear, by the $\gamma-$contraction of the operator $\hat{B}_K$ defined in (4.13). The upper bound may not be tight as certain kernels appear to perform better than the upper bound, as shown in the experiment section.

The proof of Theorem 4.2 is deferred to Section 4.4.1.

**Algorithm 2 Kernel-based Q-learning Algorithm for Continuous Space with Deterministic Dynamic (CDD-K-Q)**

**Input**: Initial state $s_0$, $\epsilon > 0$, $\epsilon$-net on $\mathcal{C} : \mathcal{C}_\epsilon = \{c^i = (s^i, a^i)\}_{i=1}^{N_\epsilon}$, exploration policy $\pi$ taking actions from $\mathcal{A}_\epsilon$ induced from $\mathcal{C}_\epsilon$, regression kernel $K$ on $\mathcal{C}_\epsilon$.
**Initialize**: $\hat{r}(c^i) = 0$, $\hat{\Phi}(c^i) = 0$, $N(c^i) = 0, \forall i$.
**repeat**
  At the current state $s_t$, act $a_t$ according to $\pi$, observe $s_{t+1} = \Phi(s_t, a_t)$ and $r_t = r(s_t, a_t)$.
  **for** $1 \le i \le N_\epsilon$ **do**
    **if** $d_{\mathcal{C}}(c^i, (s_t, a_t)) < \epsilon$ **then**
      $N(c^i) \leftarrow N(c^i) + 1$.
      $\hat{r}(c^i) \leftarrow \frac{N(c^i)-1}{N(c^i)} \cdot \hat{r}(c^i) + \frac{1}{N(c^i)} \cdot r_t$
      $\hat{\Phi}(c^i) \leftarrow \frac{N(c^i)-1}{N(c^i)} \cdot \hat{\Phi}(c^i) + \frac{1}{N(c^i)} \cdot s_t$
    **end if**
  **end for**
**until** $N(c^i) > 0, \forall i$.
**Initialize**: $\hat{q}_0(c^i) = 0, \forall c^i \in \mathcal{C}_\epsilon$, $l = 0$.
**repeat**
  **for** $c^i \in \mathcal{C}_\epsilon$ **do**
    $\hat{q}_{l+1}(c^i) \leftarrow \Big(\hat{r}(c^i) + \gamma \max_{\tilde{a} \in \mathcal{A}_\epsilon} \Gamma_K \hat{q}_l(\hat{\Phi}(c^i), \tilde{a})\Big)$.
  **end for**
  $l = l + 1$.
**until** converge

**Theorem 4.3** *Assume Assumptions 4.5, 4.6, 4.7, 3.4. Let $\hat{B}_K : \mathbb{R}^{\mathcal{C}_\epsilon} \to \mathbb{R}^{\mathcal{C}_\epsilon}$ be the operator defined by*

$$(\hat{B}_K\, q)(c^i) = \hat{r}(c^i) + \gamma \max_{\tilde{a} \in \mathcal{A}_\epsilon} \Gamma_K q(\hat{\Phi}(c^i), \tilde{a}), \tag{4.13}$$

*where $\hat{r}(c)$ and $\hat{\Phi}(c)$ are sampled from an $\epsilon$-neighborhood of $c$, then it has a unique fixed point $\hat{Q}_{\mathcal{C}_\epsilon}$ in $\{f \in \mathbb{R}^{\mathcal{C}_\epsilon} : ||f||_\infty \le V_{\max}\}$. Moreover, the sup distance between $\Gamma_K \hat{Q}_{\mathcal{C}_\epsilon}$ in (3.5) and $Q_{\mathcal{C}}$ in (4.11) is*

$$||Q_{\mathcal{C}} - \Gamma_K \hat{Q}_{\mathcal{C}_\epsilon}||_\infty \le \frac{L_r + \gamma 2 N_K L_K V_{\max} L_\Phi}{1-\gamma} \cdot \epsilon + \frac{2L_r}{(1-\gamma L_\Phi)(1-\gamma)} \cdot \epsilon.$$

*And for a fixed $\epsilon$, Algorithm 2 converges linearly to $\hat{Q}_{\mathcal{C}_\epsilon}$.*

The proof of Theorem 4.3 is deferred to Section 4.4.1.

**Remark 4.1 (Comparison to [23])** *Our idea of kernel-based Q-learning method is motivated by [23]. However, our work is different from theirs in both the problem setting and the techniques for error bound analysis. In particular, Theorem 4.2 has two layers of approximations: action space approximation and state space approximation; whereas the action space in [23] has only state space approximation as their action space is finite. Secondly, the error control in [23] is guaranteed by Martingale concentration inequalities where as the error bound analysis in Theorem 4.2 is via the Lipschitz continuity of the dynamics.*

## 4.3 Complexity Analysis for CDD-K-Q Algorithm

Note in the classical Q-learning for stochastic environment, it is necessary that every component in the $\epsilon$-net be visited sufficiently many times for a good estimate. The terminology **covering time** refers to the expected time step for a certain exploration policy to visit every component in the $\epsilon$-net at least once. The complexity analysis would then focus on how many rounds of covering time is needed. In a deterministic dynamics, however, visiting each component in the $\epsilon$-net once is sufficient, thus reducing the complexity analysis to designing an exploration scheme to guarantee the boundedness of the covering time with high probability. To this end, the following assumption on the dynamics is needed.

**Assumption 4.8 (Controllability of the dynamics)** *For all $\epsilon > 0$, there exists $M_\epsilon \in \mathbb{N}$, such that for any $\epsilon$-net $\mathcal{A}_\epsilon$ on $\mathcal{A}$ and $s, s' \in \mathcal{S}$, there always exists an action sequence $(a^1, \ldots a^m), a^i \in \mathcal{A}_\epsilon, m < M_\epsilon$, that will drive the state from $s$ to to an $\epsilon$-neighborhood of $s'$ by taking $(a^1, \ldots a^m)$.*

**Example** Here is an example where Assumption 4.8 holds with $M_\epsilon$ independent of $\epsilon$. If for any fixed $s$, $\Phi(s, \cdot)$ is a surjective mapping from $A$ to $S$, then the assumption holds with $M_\epsilon = 1$, as long as $\Phi(s, \cdot)$ is 1-Lipschitz, which holds for a wide class of linear control problems [8].

Let us denote $T_{\mathcal{C},\pi}$ as the covering time of the $\epsilon$-net under policy $\pi \in \mathcal{P}(\mathcal{A})$, such that

$$T_{\mathcal{C},\pi} := \sup_{s\in\mathcal{S}} \inf\{t > 0 : s_0 = s, \forall c^i \in \mathcal{C}_\epsilon, \exists t_i \leq t, \\ (s_{t_i}, a_{t_i}) \text{ in the } \epsilon\text{-neighborhood of } c^i, \text{ under the policy } \pi\}.$$

Recall that an $\epsilon'$-greedy policy on $\mathcal{A}_\epsilon$ is a policy which with probability at least $\epsilon'$ will uniformly explore the actions on $\mathcal{A}_\epsilon$. Note that this type of policy always exists. Then we have the following sample complexity result with proof given in Section 4.4.2.

**Theorem 4.4** *(Bound for $T_c$) Given Assumption 4.8, for any $\epsilon' > 0$, let $\pi_{\epsilon'}$ be an $\epsilon'$-greedy policy on $\mathcal{A}_\epsilon$. Then*

$$\mathbb{E}[T_{\mathcal{C},\pi_{\epsilon'}}] \leq \frac{(M_\epsilon + 1) \cdot (N_{\epsilon,\mathcal{A}})^{M_\epsilon+1}}{(\epsilon')^{M_\epsilon+1}} \cdot \log(N_\epsilon). \tag{4.14}$$

*Moreover, with probability $1 - \delta$, for any initial state $s$, under the $\epsilon'$-greedy policy, the dynamics will visit each $\epsilon$-neighborhood of elements in $\mathcal{C}_\epsilon$ at least once, after*

$$\frac{(M_\epsilon + 1) \cdot (N_{\epsilon,\mathcal{A}})^{M_\epsilon+1}}{(\epsilon')^{M_\epsilon+1}} \cdot \log(N_\epsilon) \cdot e \cdot \log(1/\delta). \tag{4.15}$$

*time steps.*

Theorem 4.4 provides an upper bound for the covering time under the $\epsilon'$-greedy policy. This upper bound is $\Omega(\text{poly}((1/\epsilon)\cdot\log(1/\delta)))$ in terms of the size of the $\epsilon$-net and the accuracy $1/\delta$. Our theoretical analysis can be adapted for other exploration schemes as well: Gaussian exploration and Boltzmann exploration. This does not affect the sample complexity, as long

as the probability to explore every action on $\mathcal{A}_\epsilon$ is lower bounded by some constant. The proof of Theorem 4.4 is deferred to Section 4.4.2.

Combining Theorem 4.3 and Theorem 4.4 yields the following convergence and sample complexity results for Algorithm 2.

**Theorem 4.5** *Assume Assumptions 4.5, 4.6, 4.7, 4.8, 3.4, then under the $\epsilon'$-greedy policy, with probability $1-\delta$, for any initial state $s$, after $\frac{(M_\epsilon+1)\cdot(N_{\epsilon,\mathcal{A}})^{M_\epsilon+1}}{(\epsilon')^{M_\epsilon+1}} \cdot \log(N_\epsilon) \cdot \log(1/\delta) \cdot e$ samples, Algorithm 2 converges linearly to the unique fixed point $\hat{Q}_{\mathcal{C}_\epsilon}$ of (4.13); and the sup distance between $\Gamma_K \hat{Q}_{\mathcal{C}_\epsilon}$ in (3.5) and $Q_{\mathcal{C}}$ in (2.3) is*

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

## 4.4 Additional Proofs

### 4.4.1 Proof of Theorems 4.2 and 4.3

As discussed before, in total there are 3 sources of the approximation error in Algorithm 2: kernel regression, discretized action space, and estimated data (for both dynamics and rewards). The core idea in the convergence analysis is to decompose the error according the sources and to analyze each part one by one. To facilitate the error decomposition, there are several different types of Bellman operators we will consider along the proofs:

- the operator $B : \mathbb{R}^{\mathcal{C}} \to \mathbb{R}^{\mathcal{C}}$ for the MDP-CDD problem

$$(B\,q)(c^i) = r(c^i) + \gamma \max_{\tilde{a} \in \mathcal{A}} q(\Phi(c^i), \tilde{a}); \tag{4.16}$$

- the operator $B_{\mathcal{A}_\epsilon} : \mathbb{R}^{\mathcal{C}} \to \mathbb{R}^{\mathcal{C}}$ for the MDP-CDD problem with discretized action space

$$B_{\mathcal{A}_\epsilon} q(c) = r(c) + \gamma \max_{\tilde{a} \in \mathcal{A}_\epsilon} q(\Phi(c), \tilde{a}); \tag{4.17}$$

- the operator $B_K$ in (3.8), involving discretized action space **and** kernel approximation;
- the operator $\hat{B}_K$ in (4.13), involving discretized action space, kernel approximation **and** estimated data.

Under mild assumptions, each of those 4 operators admits a unique fixed point, which will be stated formally in the next lemma.

**Lemma 4.1** *Given Assumption 4.6, let $V_{\max} := \frac{R}{1-\gamma}$. Then*

- *$B$ has a unique fixed point, $Q_{\mathcal{C}}$, in $\{f \in \mathbb{R}^{\mathcal{C}} : \|f\|_\infty \leq V_{\max}\}$;*
- *$B_{\mathcal{A}_\epsilon}$ has a unique fixed point, $\tilde{Q}_{\mathcal{C}}$, in $\{f \in \mathbb{R}^{\mathcal{C}} : \|f\|_\infty \leq V_{\max}\}$;*
- *$B_K$ has a unique fixed point, $Q_{\mathcal{C}_\epsilon}$, in $\{f \in \mathbb{R}^{\mathcal{C}_\epsilon} : \|f\|_\infty \leq V_{\max}\}$;*
- *$\hat{B}_K$ has a unique fixed point, $\hat{Q}_{\mathcal{C}_\epsilon}$, in $\{f \in \mathbb{R}^{\mathcal{C}_\epsilon} : \|f\|_\infty \leq V_{\max}\}$.*

*Proof.* By definition, it is easy to show that $B$ and $B_{\mathcal{A}_\epsilon}$ map $\{f \in \mathbb{R}^{\mathcal{C}} : \|f\|_\infty \le V_{\max}\}$ to itself, and that $B_K$ and $\hat{B}_K$ maps $\{f \in \mathbb{R}^{\mathcal{C}_\epsilon} : \|f\|_\infty \le V_{\max}\}$ to itself.

For $B_K$, we have

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

where we use the definition of kernel function $K(c^i, c) \ge 0$ and $\sum_{i=1}^{N_\epsilon} K(c^i, c) = 1$.

Therefore, $B_K$ is a contraction mapping with modulus $\gamma < 1$ under the sup norm on $\{f \in \mathbb{R}^{\mathcal{C}_\epsilon} : \|f\|_\infty \le V_{\max}\}$. By Banach fixed point Theorem, the statement for $B_K$ holds. Similar arguments prove the statements for the other 3 operators. □

The quantity of interests is the sup distance between $Q_{\mathcal{C}}$ and $\Gamma_K \hat{Q}_{\mathcal{C}_\epsilon}$. It can be decomposed and upper bounded by

$$||Q_{\mathcal{C}} - \Gamma_K \hat{Q}_{\mathcal{C}_\epsilon}||_\infty \le ||Q_{\mathcal{C}} - \tilde{Q}_{\mathcal{C}}||_\infty + ||\tilde{Q}_{\mathcal{C}} - \Gamma_K Q_{\mathcal{C}_\epsilon}||_\infty + ||\Gamma_K Q_{\mathcal{C}_\epsilon} - \Gamma_K \hat{Q}_{\mathcal{C}_\epsilon}||_\infty. \tag{4.18}$$

The first 2 terms in the decomposition will be handled in the proof of Theorem 4.2, while the last term will be analyzed in the proof of Theorem 4.3.

There are several facts we need before we can state the proof of Theorem 4.2. The first is a characterization for $\tilde{Q}_{\mathcal{C}}$. We can imagine from the formulation of $B_{\mathcal{A}_\epsilon}$ that, $\tilde{Q}_{\mathcal{C}}$ is related to the value function of the MDP-CDD problem defined on the continuous state space and discretized action space. The following lemma make this intuition formal.

**Lemma 4.2** *Given Assumption 4.6, consider the operator $T$ mapping from $\{f \in \mathbb{R}^{\mathcal{S}} : \|f\|_\infty \le V_{\max}\}$ to itself,*

$$Tv(s) = \max_{a\in\mathcal{A}^\epsilon}(r(s,a) + \gamma v(\Phi(s,a))).$$

*Then it has a unique fixed point $\tilde{V}_{\mathcal{C}}$ and $\tilde{Q}_{\mathcal{C}}(s,a) = r(s,a) + \gamma \tilde{V}_{\mathcal{C}}(\Phi(s,a))$.*

**Proof of Lemma 4.2** Similar as in Lemma 4.1, it is easy to show that $T$ is a contraction mapping with modulus $\gamma$ with the supremum norm on $\{f \in \mathbb{R}^{\mathcal{S}} : \|f\|_\infty \le V_{\max}\}$. So it has a fixed point $\tilde{V}_{\mathcal{C}}$. In fact, it is the value function of the MDP given by restricting the action space of the original MDP to $\mathcal{A}^\epsilon$. Moreover, define $\tilde{Q}(s,a) := r(s,a) + \gamma \tilde{V}_{\mathcal{C}}(\Phi(s,a))$.

$$\begin{aligned}
&\tilde{Q}(s,a) \\
=&r(s,a) + \gamma \tilde{V}_{\mathcal{C}}(\Phi(s,a)) \\
=&r(s,a) + \gamma \max_{a'\in\mathcal{A}^\epsilon}(r(\Phi(s,a),a') + \gamma \tilde{V}_{\mathcal{C}}(\Phi(\Phi(s,a),a'))) \\
=&r(s,a) + \gamma \max_{a'\in\mathcal{A}^\epsilon} \tilde{Q}(\Phi(s,a),a').
\end{aligned}$$

So $\tilde{Q} \in \{f \in \mathbb{R}^{\mathcal{C}} : \|f\|_\infty \le V_{\max}\}$ is a fixed point of $B_{\mathcal{A}_\epsilon}$. By Lemma 4.1, $\tilde{Q} = \tilde{Q}_{\mathcal{C}}$. □

The second thing we need to check is the existence of a stationary optimal policy. In the case of finite state and action space, the existence is a well-known fact, while it becomes more tricky under the continuous setting. The reason why we care about the existence is that, when we analyze the term $||Q_{\mathcal{C}} - \tilde{Q}_{\mathcal{C}}||_\infty$ in (4.18), as shown in Lemma 4.2, we are actually compare the optimal values of 2 MDPs, in which one has a larger action space than the other. By checking the performance of the optimal policy from one problem applied to the other, we are able to bound the gap between 2 value functions. In the next lemma, we prove the existence under mild assumptions.

**Lemma 4.3** *Given Assumptions 4.5, 4.6, 4.7, there exists an optimal stationary policy* $\pi^* : \mathcal{S} \to \mathcal{A}$ *which attains the optimal state-action function, i.e.* $Q_{\mathcal{C}}^{\pi^*} = Q_{\mathcal{C}}$.

**Proof of Lemma 4.3** First, we claim that under the problem setting of **MDP-CDD** in Section 4.1, the optimal Q function in (4.11) or value function in (4.10) will not change if we restrict the policy class to be the deterministic policies. In other words, denote $\Pi_s := \{\pi = \{\pi_t\}_{t=0}^\infty \,|\, \pi_t : \mathcal{S} \to \mathcal{P}(\mathcal{A})\}$ and $\Pi_d := \{\tilde{\pi} = \{\tilde{\pi}_t\}_{t=0}^\infty \,|\, \tilde{\pi}_t : \mathcal{S} \to \mathcal{A}\}$, and

$$V_{\mathcal{C}}(s) = \sup_{\pi \in \Pi_s} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \Big| s_0 = s, a_t \sim \pi_t(s_t)\right],$$

$$\tilde{V}_{\mathcal{C}}(s) = \sup_{\tilde{\pi} \in \Pi_d} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \Big| s_0 = s, a_t = \tilde{\pi}_t(s_t)\right],$$

then we will show that $V_{\mathcal{C}} = \tilde{V}_{\mathcal{C}}$. To complete the proof, we need to define the dynamic versions of $V_{\mathcal{C}}(s)$ and $\tilde{V}_{\mathcal{C}}(s)$

$$V_{\mathcal{C},t}^{\pi}(s) := \mathbb{E}\left[\sum_{\tau=t}^{\infty} \gamma^\tau r(s_\tau, a_\tau) \Big| s_t = s, a_\tau \sim \pi_\tau(s_\tau)\right],$$

$$V_{\mathcal{C},t}^{\tilde{\pi}}(s) := \left[\sum_{\tau=t}^{\infty} \gamma^\tau r(s_\tau, a_\tau) \Big| s_t = s, a_\tau = \tilde{\pi}_\tau(s_\tau)\right].$$

Iteratively, we have

$$V_{\mathcal{C},t}^{\pi}(s) = \int_{\mathcal{A}} \left[\gamma^t r(s,a) + V_{\mathcal{C},t+1}^{\pi}(\Phi(s,a))\right] \pi_t(s, da)$$

$$V_{\mathcal{C},t}^{\tilde{\pi}}(s) = \gamma^t r(s, \tilde{\pi}_t(s)) + V_{\mathcal{C},t+1}^{\tilde{\pi}}(\Phi(s, \tilde{\pi}_t(s)))$$

It suffices to show that for any $\pi \in \Pi_s$, there exists $\tilde{\pi} \in \Pi_d$, such that for any $s \in \mathcal{S}$, $V_{\mathcal{C},0}^{\pi}(s) - \epsilon \le V_{\mathcal{C},0}^{\tilde{\pi}}(s)$.

Now fix $\pi \in \Pi_s$ and arbitrarily initialize $\tilde{\pi} \in \Pi_d$. We start with a sufficiently large $T$ such that $\frac{2\gamma^T R}{1-\gamma} < \frac{\epsilon}{2}$. From time $t = T-1$ to $t = 0$, we choose $\tilde{\pi}_t(s)$ to be $a_t \in \mathcal{A}$ such that

$$\begin{aligned}
&V_{\mathcal{C},t}^{\pi}(s) - (1-\gamma)\gamma^t \epsilon/2 \\
\le &\sup_{a \in \mathcal{A}} [\gamma^t r(s,a) + V_{\mathcal{C},t+1}^{\pi}(\Phi(s,a))] - (1-\gamma)\gamma^t \epsilon/2 \\
\le &\gamma^t r(s, a_t) + V_{\mathcal{C},t+1}^{\pi}(\Phi(s, a_t)).
\end{aligned}$$

In this case, $V^{\pi}_{\mathcal{C},t}(s) - (1-\gamma)\gamma^t\epsilon/2 \le \gamma^t r(s, \tilde{\pi}_t(s)) + V^{\pi}_{\mathcal{C},t+1}(\Phi(s, \tilde{\pi}_t(s)))$. Now we claim that $\tilde{\pi}$ constructed in this way satisfies $V^{\pi}_{\mathcal{C},0}(s) - \epsilon \le V^{\tilde{\pi}}_{\mathcal{C},0}(s), \forall s \in \mathcal{S}$.

First, at time $T$, $||V^{\pi}_{\mathcal{C},T}||_\infty \le \frac{\gamma^T R}{1-\gamma}$, $||V^{\tilde{\pi}}_{\mathcal{C},T}||_\infty \le \frac{\gamma^T R}{1-\gamma}$, by Assumption 4.6. Hence $V^{\pi}_{\mathcal{C},T}(s) - V^{\tilde{\pi}}_{\mathcal{C},T}(s) \le \frac{2\gamma^T R}{1-\gamma} < \epsilon/2, \forall s \in \mathcal{S}$. Denote $e_t = \sup_{s\in\mathcal{S}}(V^{\pi}_{\mathcal{C},t}(s) - V^{\tilde{\pi}}_{\mathcal{C},t}(s))$, then $\forall s \in \mathcal{S}$,

$$
\begin{aligned}
&V^{\pi}_{\mathcal{C},t-1}(s) - V^{\tilde{\pi}}_{\mathcal{C},t-1}(s)\\
=&V^{\pi}_{\mathcal{C},t-1}(s) - (\gamma^{t-1} r(s, \tilde{\pi}_{t-1}(s)) + V^{\pi}_{\mathcal{C},t}(\Phi(s, \tilde{\pi}_{t-1}(s))))\\
&+ (\gamma^{t-1} r(s, \tilde{\pi}_{t-1}(s)) + V^{\pi}_{\mathcal{C},t}(\Phi(s, \tilde{\pi}_{t-1}(s))))\\
&- (\gamma^{t-1} r(s, \pi_{t-1}(s)) + V^{\tilde{\pi}}_{\mathcal{C},t}(\Phi(s, \tilde{\pi}_{t-1}(s))))\\
\le&(1-\gamma)\gamma^{t-1}\epsilon/2 + V^{\pi}_{\mathcal{C},t}(\Phi(s, \tilde{\pi}_{t-1}(s))) - V^{\tilde{\pi}}_{\mathcal{C},t}(\Phi(s, \tilde{\pi}_{t-1}(s)))\\
\le&(1-\gamma)\gamma^{t-1}\epsilon/2 + e_t
\end{aligned}
$$

Therefore, $e_{t-1} \le e_t + (1-\gamma)\gamma^{t-1}\epsilon/2$, and $e_0 \le e_T + \sum_{t=1}^{T}(1-\gamma)\gamma^{t-1}\epsilon/2 < \epsilon$. Thus $V_{\mathcal{C}} = \tilde{V}_{\mathcal{C}}$.

In order to show the optimal stationary policy exists, it suffices to prove that $Q_{\mathcal{C}}$ is continuous. Then since $\mathcal{A}$ is compact, for any fixed state $s \in \mathcal{S}$, there exists $\pi^*(s) \in \mathcal{A}$ such that $\pi^*(s) = \arg\max_{a\in\mathcal{A}} Q_{\mathcal{C}}(s,a)$. In this case, it can be checked easily that $Q_{\mathcal{C}}$ satisfies the Bellman equation $Q_{\mathcal{C}} = B^{\pi^*}Q_{\mathcal{C}}$, where the operator $B^{\pi^*}$ is defined to be $B^{\pi^*}q(s,a) = r(s,a) + \gamma q(\Phi(s,a), \pi^*(\Phi(s,a)))$ on $\mathbb{R}^{\mathcal{C}}$. Moreover, one can show that this operator is a contraction with modulus $\gamma$ under the infinity-norm, and $Q^{\pi^*}_{\mathcal{C}}$ is the unique fixed point of $B^{\pi^*}$, therefore, $Q^{\pi^*}_{\mathcal{C}} = Q_{\mathcal{C}}$.

To prove the continuity of $Q_{\mathcal{C}}$, first fix $(s,a) \in \mathcal{C}$ and $(s',a') \in \mathcal{C}$ to be two state-action pairs. Then there exists some policy $\pi$ such that $Q_{\mathcal{C}}(s,a) - Q^{\pi}_{\mathcal{C}}(s,a) < \frac{\epsilon}{2}$. Let $(s,a) = (s_0,a_0),(s_1,a_1),(s_2,a_2),\ldots,(s_t,a_t),\ldots$ be the trajectory of the system starting from arbitrary state-action pair $(s,a) \in \mathcal{C}$ and then under the policy $\pi$. Then $Q^{\pi}_{\mathcal{C}}(s,a) = \sum_{t=0}^{\infty}\gamma^t r(s_t,a_t)$.

Now consider the trajectory of the system starting from $(s',a')$ and then taking $a_1,\ldots,a_n,\ldots$, denoted by $(s',a') = (s'_0,a'_0),(s'_1,a_1),(s'_2,a_2),\ldots,(s'_t,a_t),\ldots$. Note that this trajectory starting from $(s',a')$ may not be the optimal trajectory, therefore, $Q_{\mathcal{C}}(s',a') \ge \sum_{t=0}^{\infty}\gamma^t r(s'_t,a'_t)$. By Assumption (4.5) and (4.6)

$$
\begin{aligned}
&|r(s'_t,a_t) - r(s_t,a_t)|\\
\le&L_r \cdot d_{\mathcal{S}}(s'_t,s_t) = L_r \cdot d_{\mathcal{S}}(\Phi(s'_{t-1},a_{t-1}),\Phi(s_{t-1},a_{t-1}))\\
\le&L_r \cdot L_\Phi \cdot d_{\mathcal{S}}(s'_{t-1},s_{t-1}) \le \cdots \le L_r \cdot L^t_\Phi \cdot d_{\mathcal{C}}((s,a),(s',a')),
\end{aligned}
$$

implying that

$$
\begin{aligned}
&Q_{\mathcal{C}}(s,a) - Q_{\mathcal{C}}(s',a')\\
\le&\frac{\epsilon}{2} + Q^{\pi}_{\mathcal{C}}(s,a) - Q_{\mathcal{C}}(s',a') \le \frac{\epsilon}{2} + \sum_{t=0}^{\infty}\gamma^t(r(s_t,a_t) - r(s'_t,a_t))\\
\le&\frac{\epsilon}{2} + \sum_{t=0}^{\infty}\gamma^t \cdot L^t_\Phi \cdot L_r \cdot d_{\mathcal{C}}((s,a),(s',a')) = \frac{\epsilon}{2} + \frac{L_r}{1-\gamma\cdot L_\Phi}\cdot d_{\mathcal{C}}((s,a),(s',a')).
\end{aligned}
$$

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

The last technical fact we will need in the proof of Theorem 4.2 is the Lipschitz continuity of both $Q_{\mathcal{C}}$ and $\tilde{Q}_{\mathcal{C}}$. The Lipschitz continuity is crucial to control the error introduced by the kernel regression.

**Proposition 4.3** *Given Assumptions 4.5, 4.6, 4.7, $\tilde{Q}_{\mathcal{C}}$ and $Q_{\mathcal{C}}$ are Lipschitz continuous on $\mathcal{C}$.*

**Proof of Proposition 4.3** . Let $\pi^* : \mathcal{S} \to \mathcal{A}$ be the optimal policy of the deterministic MDP on $\mathcal{S}$ and $\mathcal{A}$. Let $c = (s,a)$ and $c' = (s',a')$ be two state-action pairs. Let $(s,a) = (s_0,a_0),(s_1,a_1),(s_2,a_2),\ldots,(s_t,a_t),\ldots$ be the trajectory of the system under the optimal policy $\pi^*$, starting from state $s$ and firstly taking action $a$. Since $Q_{\mathcal{C}}$ is the Q function of this MDP, we have $Q_{\mathcal{C}}(s,a) = \sum_{t=0}^{\infty}\gamma^t r(s_t,a_t)$.

Now consider the trajectory of the system starting from $(s',a')$ and then taking $a_1,\ldots,a_n,\ldots$, denoted by $(s',a') = (s'_0,a'_0),(s'_1,a_1),(s'_2,a_2),\ldots,(s'_t,a_t),\ldots$. Note that since this trajectory starting from $(s',a')$ may not be the optimal trajectory, $Q_{\mathcal{C}}(s',a') \ge \sum_{t=0}^{\infty}\gamma^t r(s'_t,a_t)$

$$
\begin{aligned}
&|r(s'_t,a_t) - r(s_t,a_t)| \\
\le& L_r\cdot d_{\mathcal{S}}(s'_t,s_t) = L_r\cdot d_{\mathcal{S}}(\Phi(s'_{t-1},a_{t-1}),\Phi(s_{t-1},a_{t-1})) \\
\le& L_r\cdot L_\Phi\cdot d_{\mathcal{S}}(s'_{t-1},s_{t-1}) \le L_r\cdot L_\Phi^{t-1}\cdot d_{\mathcal{S}}(s'_1,s_1) \\
\le& L_r\cdot L_\Phi^{t}\cdot d_{\mathcal{C}}((s,a),(s',a')),
\end{aligned}
$$

implying

$$
\begin{aligned}
&Q_{\mathcal{C}}(s,a) - Q_{\mathcal{C}}(s',a') \\
\le&\sum_{t=0}^{\infty}\gamma^t(r(s_t,a_t) - r(s'_t,a_t)) \le \sum_{t=0}^{\infty}\gamma^t\cdot L_\Phi^t\cdot L_r\cdot d_{\mathcal{C}}((s,a),(s',a')) \\
=&\frac{L_r}{1-\gamma\cdot L_\Phi}\cdot d_{\mathcal{C}}((s,a),(s',a')).
\end{aligned}
$$

Similarly, one can show $Q_{\mathcal{C}}(s',a') - Q_{\mathcal{C}}(s,a) \le \frac{L_r}{1-\gamma\cdot L_\Phi}\cdot d_{\mathcal{C}}((s,a),(s',a'))$. This proves that $Q_{\mathcal{C}}$ is Lipschitz.

From Lemma 4.2, it is clear that in order to prove $\tilde{Q}_{\mathcal{C}}$ is Lipschitz, it suffices to show that $\tilde{V}_{\mathcal{C}}$ is Lipschitz, where $\tilde{V}_{\mathcal{C}}$ is the optimal value function of the MDP with restricted action space $\mathcal{A}_\epsilon$. This can be proved by using exactly the same argument as in the $Q_{\mathcal{C}}$ case. □

Based on Lemma 4.1 to 4.3 and Proposition 4.3, we are now ready to prove Theorem 4.2.

**Proof of Theorem 4.2** We aim to show $\|\Gamma_K Q_{\mathcal{C}_\epsilon} - \tilde{Q}_{\mathcal{C}}\|_\infty \le \frac{L_r}{(1-\gamma L_\Phi)(1-\gamma)}\cdot\epsilon$ and $\|\tilde{Q}_{\mathcal{C}} -$

$Q_C\|_\infty \le \frac{L_r\gamma}{(1-\gamma L_\Phi)(1-\gamma)} \cdot \epsilon$. To prove $||\Gamma_K Q_{\mathcal{C}_\epsilon} - \tilde{Q}_\mathcal{C}||_\infty \le \frac{L_r}{(1-\gamma L_\Phi)(1-\gamma)} \cdot \epsilon$, note that

$$\begin{aligned}
&||\Gamma_K Q_{\mathcal{C}_\epsilon} - \tilde{Q}_\mathcal{C}||_\infty \\
=&||\Gamma_K B_K Q_{\mathcal{C}_\epsilon} - \tilde{Q}_\mathcal{C}||_\infty = ||\Gamma_K B_{\mathcal{A}_\epsilon}\Gamma_K Q_{\mathcal{C}_\epsilon} - \tilde{Q}_\mathcal{C}||_\infty \\
\le&||\Gamma_K B_{\mathcal{A}_\epsilon}\Gamma_K Q_{\mathcal{C}_\epsilon} - \Gamma_K B_{\mathcal{A}_\epsilon}\tilde{Q}_\mathcal{C}||_\infty + ||\Gamma_K B_{\mathcal{A}_\epsilon}\tilde{Q}_\mathcal{C} - \tilde{Q}_\mathcal{C}||_\infty \\
=&||\Gamma_K B_{\mathcal{A}_\epsilon}\Gamma_K Q_{\mathcal{C}_\epsilon} - \Gamma_K B_{\mathcal{A}_\epsilon}\tilde{Q}_\mathcal{C}||_\infty + ||\Gamma_K \tilde{Q}_\mathcal{C} - \tilde{Q}_\mathcal{C}||_\infty \\
\le&\gamma||\Gamma_K Q_{\mathcal{C}_\epsilon} - \tilde{Q}_\mathcal{C}||_\infty + ||\Gamma_K \tilde{Q}_\mathcal{C} - \tilde{Q}_\mathcal{C}||_\infty.
\end{aligned}$$

Here the first and the third equalities come from the fact that $Q_{\mathcal{C}_\epsilon}$ is the fixed point of $B_K$ and $\tilde{Q}_\mathcal{C}$ is the fixed point of $B_{\mathcal{A}_\epsilon}$. The second inequality is by the fact that $\Gamma_K$ is a non-expansion mapping, i.e., $\|\Gamma_K f\|_\infty \le \|f\|_\infty$, and that $B_{\mathcal{A}_\epsilon}$ is a contraction with modulus $\gamma$ with the supremum norm. Meanwhile, for any Lipschitz function $f \in \mathbb{R}^\mathcal{C}$ with Lipschitz constant $L$, we have for all $c \in \mathcal{C}$,

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

Note here the inequality follows from $K(c, c^i) = 0$ for all $d_\mathcal{C}(c, c^i) \ge \epsilon$. Therefore,

$$||\Gamma_K Q_{\mathcal{C}_\epsilon} - \tilde{Q}_\mathcal{C}||_\infty \le \frac{L_{\tilde{Q}_\mathcal{C}}}{1-\gamma}\epsilon.$$

where $L_{\tilde{Q}_\mathcal{C}} = \frac{L_r}{1-\gamma L_\Phi}$ is the Lipschitz constant for $\tilde{Q}_\mathcal{C}$.

In order to prove the second part, first note that $Q_\mathcal{C}(s,a) - \tilde{Q}_\mathcal{C}(s,a) = \gamma\big(V_\mathcal{C}(\Phi(s,a)) - \tilde{V}_\mathcal{C}(\Phi(s,a))\big)$, where $V_\mathcal{C}$ is the optimal value function of the MDP on $\mathcal{S}$ and $\mathcal{A}$, and $\tilde{V}_\mathcal{C}$ is the optimal value function of the MDP on $\mathcal{S}$ and $\mathcal{A}_\epsilon$. Hence it suffices to prove that $||V_\mathcal{C} - \tilde{V}_\mathcal{C}||_\infty \le \frac{L_r}{(1-\gamma L_\Phi)(1-\gamma)} \cdot \epsilon$. We adopt the similar strategy as in the proof of Theorem 4.3.

Let $\pi^*$ be the optimal policy of the deterministic MDP on $\mathcal{S}$ and $\mathcal{A}$, whose existence is shown in Lemma 4.3. For any $s \in \mathcal{S}$, let $(s,a) = (s_0,a_0),(s_1,a_1),(s_2,a_2),\ldots,(s_t,a_t),\ldots$ be the trajectory of the system under the optimal policy $\pi^*$, starting from state $s$. We have $V_\mathcal{C}(s) = \sum_{t=0}^\infty \gamma^t r(s_t,a_t)$.

Now let $a^{i_t}$ be the nearest neighbor of $a_t$ in $\mathcal{A}_\epsilon$. $d_\mathcal{A}(a^{i_t}, a_t) \le \epsilon$. Consider the trajectory of the system starting from $s$ and then taking $a^{i_0},\ldots,a^{i_t},\ldots$, denote the corresponding state by $s'_t$. We have $\tilde{V}_\mathcal{C}(s) \ge \sum_{t=0}^\infty \gamma^t r(s'_t, a^{i_t})$, since $\tilde{V}_\mathcal{C}$ is the optimal value function.

$$d_\mathcal{S}(s'_t, s_t) = d_\mathcal{S}\big(\Phi(s'_{t-1}, a^{i_{t-1}}), \Phi(s_{t-1}, a_t)\big) \le L_\Phi \cdot \big(d_\mathcal{S}(s'_{t-1}, s_{t-1}) + \epsilon\big)$$

By the iteration, we have $d_\mathcal{S}(s'_t, s_t) \le \frac{L_\Phi - L_\Phi^{t+1}}{1-L_\Phi} \cdot \epsilon$.

$$|r(s'_t, a^{i_t}) - r(s_t, a_t)| \le L_r \cdot \big(d_\mathcal{S}(s'_t, s_t) + \epsilon\big) \le L_r \cdot \frac{L_\Phi^{t+1} - 1}{L_\Phi - 1} \cdot \epsilon,$$

which implies

$$\begin{aligned}
0 \le& V_\mathcal{C}(s) - \tilde{V}_\mathcal{C}(s) \le \sum_{t=0}^\infty \gamma^t (r(s_t,a_t) - r(s'_t, a'_t)) \\
\le& \sum_{t=0}^\infty \gamma^t \cdot L_r \cdot \frac{L_\Phi^{t+1} - 1}{L_\Phi - 1} \cdot \epsilon = \frac{L_r}{(1-\gamma L_\Phi)(1-\gamma)} \cdot \epsilon.
\end{aligned}$$

Here $0 \leq V_{\mathcal{C}}(s) - \tilde{V}_{\mathcal{C}}(s)$ is by the optimality of $V_{\mathcal{C}}$. □

The final building block we need before formally proving Theorem 4.3 is the following lemma.

**Lemma 4.4** *For any $g \in \{f \in \mathbb{R}^{\mathcal{C}_\epsilon} : ||f||_\infty \leq V_{\max}\}$, $\Gamma_K g \in \{f \in \mathbb{R}^{\mathcal{C}} : ||f||_\infty \leq V_{\max}\}$ and it is Lipschitz continuous with the Lipschitz constant bounded by $2N_K L_K V_{\max}$.*

**Proof of Lemma 4.4** $\Gamma_K g \in \{f \in \mathbb{R}^{\mathcal{C}} : ||f||_\infty \leq V_{\max}\}$ is trivial. $\forall c, c' \in \mathcal{C}, |\Gamma_K g(c) - \Gamma_K g(c')| = |\sum_{i=1}^{N_\epsilon}(K(c^i, c) - K(c^i, c'))g(c^i)|$. By Assumption 3.4, we know that $K(c^i, c) - K(c^i, c')$ is nonzero for at most $2N_K$ $1 \leq i \leq N_\epsilon$, and for all$i$, $|K(c^i, c) - K(c^i, c')| \leq L_K \cdot d_{\mathcal{C}}(c, c')$. Therefore, $|\Gamma_K g(c) - \Gamma_K g(c')| \leq 2N_K L_K V_{\max} \cdot d_{\mathcal{C}}(c, c')$. □

**Proof of Theorem 4.3** The existence and uniqueness of $\hat{Q}_{\mathcal{C}_\epsilon}$ is proved in Lemma 4.1. Let $q_0$ denote the zero function on $\mathcal{C}_\epsilon$. From the previous discussion, we know that $Q_{\mathcal{C}_\epsilon} = \lim_{N\to\infty} B_K^N q_0$, and $\hat{Q}_{\mathcal{C}_\epsilon} = \lim_{N\to\infty} \hat{B}_K^N q_0$. Denote $q_n := B_K^n q_0$, $\hat{q}_n := \hat{B}_K^n q_0$, and $e_n := ||q_n - \hat{q}_n||_\infty$. For any $(s, a) \in \mathcal{C}_\epsilon$,

$$
\begin{aligned}
&e_{n+1}(s,a)\\
=&\left|\hat{r}(s,a) + \gamma \max_{\tilde{a}\in\mathcal{A}_\epsilon} \Gamma_K \hat{q}_n(\hat{\Phi}(s,a),\tilde{a}) - r(s,a) - \gamma \max_{\tilde{a}\in\mathcal{A}_\epsilon} \Gamma_K q_n(\Phi(s,a),\tilde{a})\right|\\
\leq&|\hat{r}(s,a) - r(s,a)| + \gamma \max_{\tilde{a}\in\mathcal{A}_\epsilon} \left|\Gamma_K \hat{q}_n(\hat{\Phi}(s,a),\tilde{a}) - \Gamma_K q_n(\Phi(s,a),\tilde{a})\right|\\
\leq&\epsilon L_r + \gamma \max_{\tilde{a}\in\mathcal{A}_\epsilon} \left[|\Gamma_K \hat{q}_n(\hat{\Phi}(s,a),\tilde{a}) - \Gamma_K \hat{q}_n(\Phi(s,a),\tilde{a})| + |\Gamma_K \hat{q}_n(\Phi(s,a),\tilde{a}) - \Gamma_K q_n(\Phi(s,a),\tilde{a})|\right].
\end{aligned}
$$

Here $|\hat{r}(s,a) - r(s,a)| \leq \epsilon L_r$ because $\hat{r}(s,a)$ is sampled from an $\epsilon$-neighborhood of $(s,a)$ and by Assumption 4.6. Moreover, for any fixed $\tilde{a}$,

$$|\Gamma_K \hat{q}_n(\hat{\Phi}(s,a),\tilde{a}) - \Gamma_K \hat{q}_n(\Phi(s,a),\tilde{a})| \leq 2N_K L_K V_{\max} \cdot d_{\mathcal{S}}(\hat{\Phi}(s,a), \Phi(s,a)) \leq 2N_K L_K V_{\max} L_\Phi \epsilon.$$

The first inequality comes from Lemma 4.4 and the second inequality comes from the fact that $\hat{\Phi}(s,a)$ is sampled from an $\epsilon$-neighborhood of $(s,a)$ and by Assumption 4.5. Meanwhile,

$$|\Gamma_K \hat{q}_n(\Phi(s,a),\tilde{a}) - \Gamma_K q_n(\Phi(s,a),\tilde{a})| \leq ||q_n - \hat{q}_n||_\infty = e_n,$$

since $\Gamma$ is non-expansion. Putting these pieces together, we have

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

In this case, elementary algebra shows that $e_n \leq \epsilon \cdot \frac{L_r + \gamma 2N_K L_K V_{\max} L_\Phi}{1-\gamma}, \forall n$. Then the desired inequality comes directly from combining Theorem 4.2 with the bound on $e_n$, and using the fact that $\Gamma_K$ is non-expansion. The claim regarding the convergence rate follows from the $\gamma-$contraction of the operator $\hat{B}_K$. This completes the proof. □

### 4.4.2 Proof of Theorem 4.4

**Lemma 4.5** *Assume for some policy $\pi$, $\mathbb{E}[T_{\mathcal{C},\pi}] \leq T < \infty$. Then with probability $1-\delta$, for any initial state $s$, under the policy $\pi$, the dynamics will visit each $\epsilon$-neighborhood of elements in $\mathcal{C}_\epsilon$ at least once, after $T \cdot e \cdot \log(1/\delta)$ time steps, i.e. $\mathbb{P}(T_{\mathcal{C},\pi} \leq T \cdot e \cdot \log(1/\delta)) \geq 1-\delta$.*

**Proof of Lemma 4.5** By Markov's inequality,

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

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

**Proof of Theorem 4.4.** Recall there are $N_\epsilon$ different state-action pairs in the $\epsilon$-net. Denote the $\epsilon$-neighborhoods of those pairs by $B_\epsilon = \{B^i\}_{i=1}^{N_\epsilon}$. Without loss of generality, we may assume that $B^i$ are disjoint, since the covering time will only become smaller if they overlap with each other. Let $T_k := \min\{t > 1 : \text{k of } B_\epsilon \text{ is visited}\}$. $T_k - T_{k-1}$ is the time to visit a new neighborhood after $k-1$ neighborhoods are visited. By Assumption 3.3, for any $B^i \in B_\epsilon$ with center $(s^i, a^i)$, $s \in \mathcal{S}$, there exists a sequence of actions in $A_\epsilon$, whose length is at most $M_\epsilon$, such that starting from $s$ and taking that sequence of actions will let the agent visit the $\epsilon$-neighborhood of $s^i$. Then, at that point, taking $a^i$ will let the agent visit $B^i$. Hence $\forall B^i \in B_\epsilon$, $s \in \mathcal{S}$,

$$\mathbb{P}(B^i \text{ is visited in } M_\epsilon + 1 \text{ steps} \,|\, s_{T_{k-1}} = s) \geq (\frac{\epsilon'}{N_{\epsilon,\mathcal{A}}})^{M_\epsilon+1}.$$

$$\mathbb{P}(\text{a new neighborhood is visited in } M_\epsilon + 1 \text{ steps} \,|s_{T_{k-1}} = s, k-1 \text{neighborhoods are visited})$$

$$\geq (N_\epsilon - k + 1) \cdot (\frac{\epsilon'}{N_{\epsilon,\mathcal{A}}})^{M_\epsilon+1}.$$

This implies $\mathbb{E}[T_k - T_{k-1}] \leq \frac{M_\epsilon+1}{N_\epsilon-k+1} \cdot (\frac{N_{\epsilon,\mathcal{A}}}{\epsilon'})^{M_\epsilon+1}$. Summing $\mathbb{E}[T_k - T_{k-1}]$ from $k=1$ to $k = N_\epsilon$ yields the desired result. The second part follows directly from Lemma 4.5. □

### 4.4.3 Proof of Theorem 3.1

The proof of Theorem 3.1 relies on several lemmas.

**Lemma 4.6 (Continuity of $\nu$)**

$$\|\nu(\mu, h) - \nu(\mu', h')\|_1 \leq d_{\mathcal{C}}((\mu, h), (\mu', h')). \tag{4.19}$$

**Proof of Lemma 4.6.**

$$
\begin{aligned}
&\|\nu(\mu,h)-\nu(\mu',h')\|_1\\
\le&\|\nu(\mu,h)-\nu(\mu,h')\|_1+\|\nu(\mu,h'),\nu(\mu',h')\|_1\\
\le&\Big|\Big|\sum_{x\in\mathcal{X}}(h(x)-h'(x))\mu(x)\Big|\Big|_1+\Big|\Big|\sum_{x\in\mathcal{X}}(\mu(x)-\mu'(x))h'(x)\Big|\Big|_1\\
\le&\sum_{x\in\mathcal{X}}\mu(x)\Big|\Big|h(x)-h'(x)\Big|\Big|_1+\Big|\Big|\sum_{x\in\mathcal{X}}(\mu(x)-\mu'(x))h'(x)\Big|\Big|_1\\
\le&\max_{x\in\mathcal{X}}\Big|\Big|h(x)-h'(x)\Big|\Big|_1+\sum_{u\in\mathcal{U}}\sum_{x\in\mathcal{X}}|\mu(x)-\mu'(x)|h'(x)(u)\\
=&d_{\mathcal{H}}(h,h')+\sum_{x\in\mathcal{X}}|\mu(x)-\mu'(x)|\\
=&d_{\mathcal{H}}(h,h')+\|\mu-\mu'\|_1=d_{\mathcal{C}}((\mu,h),(\mu',h')).
\end{aligned}
$$

□

**Lemma 4.7 (Continuity of $r$)** *Under Assumption 3.1,*

$$
|r(\mu,h)-r(\mu',h')|\le(\tilde{R}+2L_{\tilde{r}})d_{\mathcal{C}}((\mu,h),(\mu',h')). \tag{4.20}
$$

**Proof of Lemma 4.7.**

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

By Assumption 3.1 and Lemma 4.6, for any $x\in\mathcal{X},u\in\mathcal{U}$,

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

Meanwhile,

$$
\begin{aligned}
&\sum_{x\in\mathcal{X}}\sum_{u\in\mathcal{U}}|\mu(x)h(x)(u)-\mu'(x)h'(x)(u)|\\
\le&\sum_{x\in\mathcal{X}}\sum_{u\in\mathcal{U}}|\mu(x)-\mu'(x)|h(x)(u)+\sum_{x\in\mathcal{X}}\sum_{u\in\mathcal{U}}\mu'(x)|h(x)(u)-h'(x)(u)|\\
=&\sum_{x\in\mathcal{X}}|\mu(x)-\mu'(x)|+\sum_{x\in\mathcal{X}}\mu'(x)\|h(x)-h'(x)\|_1\\
\le&\|\mu-\mu'\|_1+\max_{x\in\mathcal{X}}\|h_1(x)-h_2(x)\|_1=d_{\mathcal{C}}((\mu,h),(\mu',h')).
\end{aligned}
$$

Combining all these results, we have

$$\begin{aligned}
&|r(\mu,h)-r(\mu',h')| \\
\leq&\sum_{x\in\mathcal{X}}\sum_{u\in\mathcal{U}}|\tilde{r}_{x,u}-\tilde{r}'_{x,u}|\mu(x)h(x)(u)+\tilde{R}\sum_{x\in\mathcal{X}}\sum_{u\in\mathcal{U}}|\mu(x)h(x)(u)-\mu'(x)h'(x)(u)| \\
\leq&(\tilde{R}+2L_{\tilde{r}})d_{\mathcal{C}}((\mu,h),(\mu',h')).
\end{aligned}$$

□

**Lemma 4.8 (Continuity of $\Phi$)** *Under Assumption 3.2,*

$$\|\Phi(\mu,h)-\Phi(\mu',h')\|_1\leq(2L_P+1)d_{\mathcal{C}}((\mu,h),(\mu',h')). \tag{4.21}$$

**Proof of Lemma 4.8.**

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

By Assumption 3.2 and Lemma 4.6, $\forall x,u$,

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

Meanwhile, from the proof of 4.7, we show

$$\sum_{x\in\mathcal{X}}\sum_{u\in\mathcal{U}}|\mu(x)h(x)(u)-\mu'(x)h'(x)(u)|\leq d_{\mathcal{C}}((\mu,h),(\mu',h')).$$

Combining all these results, we have

$$\begin{aligned}
&\|\Phi(\mu,h)-\Phi(\mu',h')\|_1 \\
\leq&\sum_{x\in\mathcal{X}}\sum_{u\in\mathcal{U}}||P_{x,u}-P'_{x,u}||_1\mu(x)h(x,u)+\sum_{x\in\mathcal{X}}\sum_{u\in\mathcal{U}}||P'_{x,u}||_1|\mu(x)h(x)(u)-\mu'(x)h'(x)(u))| \\
\leq&(2L_P+1)d_{\mathcal{C}}((\mu,h),(\mu',h')).
\end{aligned}$$

□

**Proof of Theorem 3.1** Now by Lemma 4.7 and Lemma 4.8, Assumption 4.5, 4.6 and 4.7 hold with $L_r=\tilde{R}+2L_{\tilde{r}}$, $R=\tilde{R}$ and $L_\Phi=2L_P+1$. Meanwhile, $N_{\epsilon,\mathcal{A}}$, the size of the $\epsilon$-net in $\mathcal{A}$ is $\Theta((\frac{1}{\epsilon})^{(|\mathcal{U}|-1)|\mathcal{X}|})$, because $\mathcal{A}=\mathcal{H}$ is a compact $(|\mathcal{U}|-1)|\mathcal{X}|$ dimensional manifold. Similarly, $N_\epsilon=\Theta((\frac{1}{\epsilon})^{|\mathcal{U}||\mathcal{X}|-1})$ as $\mathcal{C}$ is a compact $|\mathcal{U}||\mathcal{X}|-1$ dimensional manifold. Theorem 3.1 follows directly from plugging those constants into Theorem 4.5. □

# 5 Experiments

We will test the MFC-K-Q algorithm on a network traffic congestion control problem. In the network there are senders and receivers. Multiple senders share a single communication link which has an unknown and limited bandwidth. When the total sending rates from these senders exceed the shared bandwidth, packages may be lost. Sender streams data packets to the receiver and receives feedback from the receiver on success or failure in the form of packet acknowledgements (ACKs). (See Figure 1 for illustration and [12] for a similar set-up). The control problem for each sender is to send the packets as fast as possible and with the risk of packet loss as little as possible. Given a large interactive population of senders, the exact dynamics of the system and the rewards are unknown, thus it is natural to formulate this control problem in the framework of learning MFC.

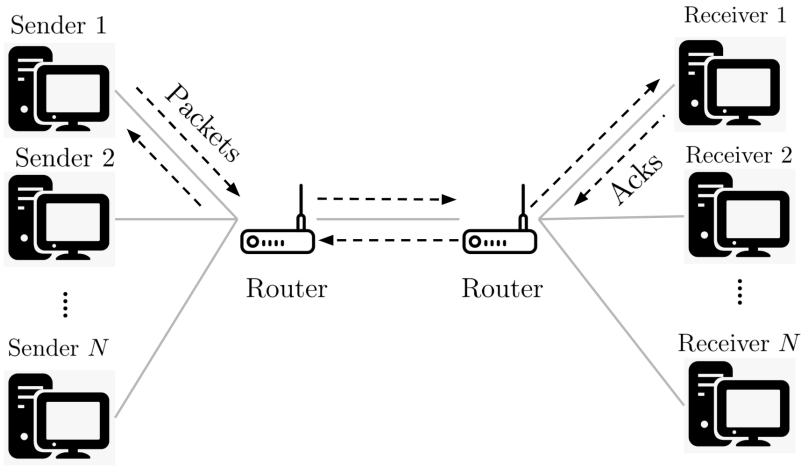



Figure 1: Multiple network traffic flows sharing the same link.

## 5.1 Set-up

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

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

**Limited bandwidth and packet loss.** A system with $N$ agents has a shared link of unknown bandwidth $cN$ ($c > 0$). In the mean-field limit with $N \to \infty$, $F_t = \sum_{x \in \mathcal{X}, u \in \mathcal{U}} u h_t(x)(u) \mu_t(x)$ is the average sending rate at time $t$. If $F_t > c$, with probability $\frac{(F_t - c)}{F_t}$, each agent's packet will be lost.

**MFC dynamics.** At time $t+1$, the state of the representative agent moves from $x_t$ to $x_t - u_t$. Overshooting is not allowed: $u_t \leq x_t$. Meanwhile, at the end of each round, there are some packets added to each agent's packet sending queue. The packet fulfillment consists

of two scenarios. First, a lost package will be added to the original queue. Second, once the inventory hits zero, a random fulfillment with uniform distribution Unif($\mathcal{X}$) will be added to her queue. That is,

$$x_{t+1} = x_t - u_t + u_t 1_t(L) + (1 - 1_t(L)\mathbf{I}(u_t = x_t) \cdot U_t,$$

where $1_t(L) = \mathbf{I}$(packet is lost in round t), with $\mathbf{I}$ an indicator function and $U_t \sim$ Unif($\mathcal{X}$).

**Evolution of population state distribution $\mu_t$.** Define, for $x \in \mathcal{X}$,

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

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

**Rewards.** Consistent with [5] and [12], the reward function depending on throughput, latency, with loss penalty is defined as $\tilde{r} = a * \text{throughput} - b * \text{latency}^2 - d * \text{loss}$, with $a, b, d \geq 0$.

## 5.2 Performance of MFC-K-Q Algorithm

We first test the convergence property and performance of MFC-K-Q (Algorithm 1) for this traffic control problem with different kernel choices and with varying $N$. We then compare MFC-K-Q with MFQ Algorithm [4] on MFC, Deep PPQ [12], and PCC-VIVACE [5] on MARL.

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

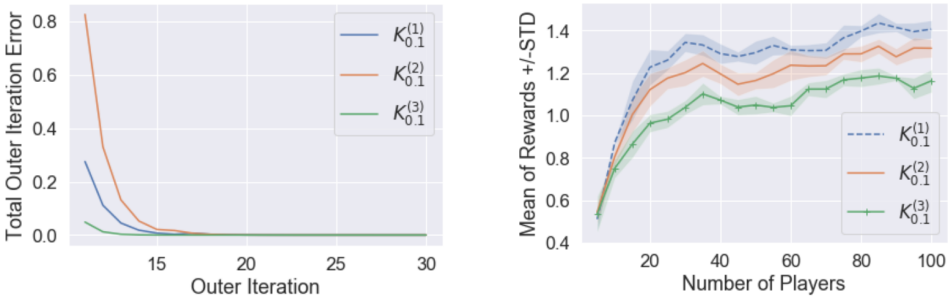
**Applying MFC policy to $N$-agent game.** To measure the performance of the MFC policy $\pi$ for an $N$-agent set-up, we apply $\pi$ to the empirical state distribution of $N$ agents.

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

We randomly sample $K$ initial states $\{\boldsymbol{x}_0^k \in \mathcal{X}^N\}_{k=1}^K$ and apply policy $\pi$ to each initial state $\boldsymbol{x}_0^k$ and collect the continuum rewards in each path for $T_0$ rounds $\{\bar{r}_{k,t}^\pi\}_{t=1}^{T_0}$. Here $\bar{r}_{k,t}^\pi = \frac{\sum_{i=1}^N r_k^{\pi,i}}{N}$ is the average reward from $N$ agents in round $t$ under policy $\pi$. Then $R_N^\pi(\boldsymbol{x}_0^k) := \sum_{t=1}^{T_0} \gamma^t \bar{r}_{k,t}^\pi$ is used to approximate the value function $V_{\mathcal{C}}^\pi$ with policy $\pi$, when $T_0$ is large.

Two performance criteria are used: the first one $C_N^{(1)}(\pi) = \frac{1}{K}\sum_{k=1}^{K} R_N^{\pi}(\boldsymbol{x}_0^k)$ measures the average reward from policy $\pi$; and the second criterion $C_N^{(2)}(\pi^1, \pi^2) = \frac{1}{K}\sum_{k=1}^{K} \frac{R_N^{\pi_1}(\boldsymbol{x}_0^k) - R_N^{\pi_2}(\boldsymbol{x}_0^k)}{R_N^{\pi_1}(\boldsymbol{x}_0^k)}$ measures the relative improvements of using policy $\pi^1$ instead of policy $\pi^2$.

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



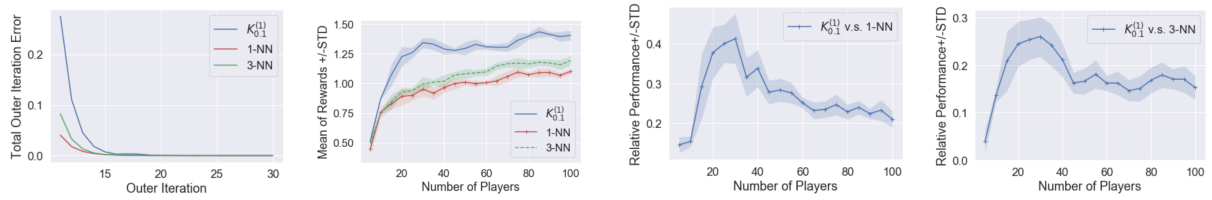

(a) Convergence of Q function. (b) $C_N^{(1)}$: Average reward.

Figure 2: Performance comparison among different kernels.

**Results with different kernels.** We use the following kernels with hyper-parameter $\epsilon$: triangular, (truncated) Gaussian, and (truncated) constant kernels. That is, $\phi_\epsilon^{(1)}(x, y) = \mathbf{1}_{\{\|x-y\|_2 \le \epsilon\}} \big|\epsilon - \|x - y\|_2\big|$, $\phi_\epsilon^{(2)}(x, y) = \mathbf{1}_{\{\|x-y\|_2 \le \epsilon\}} \frac{1}{\sqrt{2\pi}} \exp(-|\epsilon - \|x - y\|_2|^2)$, and $\phi_\epsilon^{(3)}(x, y) = \mathbf{1}_{\{\|x-y\|_2 \le \epsilon\}}$. We run the experiments for $K_\epsilon^{(j)}(c^i, c) = \frac{\phi_\epsilon^{(j)}(c^i, c)}{\sum_{i=1}^{N_\epsilon} \phi_\epsilon^{(j)}(c^i, c)}$. with $j = 1, 2, 3$ and $\epsilon = 0.1$.

All kernels lead to the convergence of Q functions within 15 outer iterations (Figure 2a). When $N \le 10$, the performances of all kernels are similar since $\epsilon$-net is accurate for games with $N = \frac{1}{\epsilon}$ agents. When $N \ge 15$, $K_{0.1}^{(1)}$ performs the best and $K_{0.1}^{(3)}$ does the worst (Figure 2b): implying that treating all nearby $\epsilon$-net points with equal weights yields relatively poor performance.

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




(a) Convergence of Q function (b) $C_N^{(1)}$: Average reward. (c) $C_N^{(2)}$: Improvement of $K_{0.1}^{(1)}$ from 1-NN. (d) $C_N^{(2)}$: Improvement of $K_{0.1}^{(1)}$ from 3-NN.

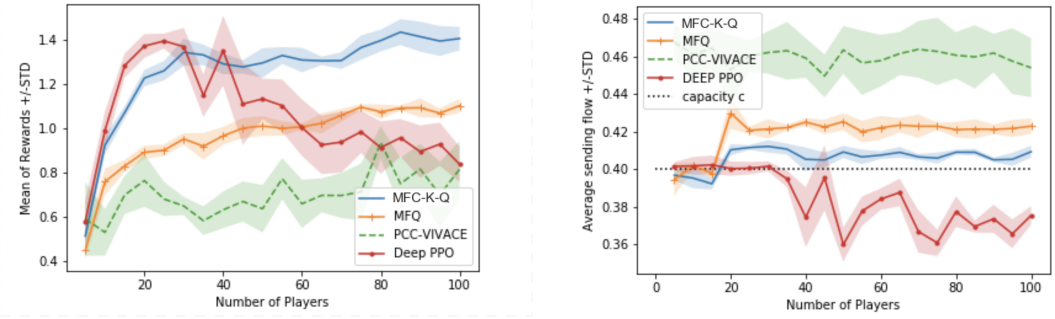
Figure 3: Comparison between $K_{0.1}^{1}(x, y)$ and $k$-NN ($k = 1, 3$).

**Results with different $k$-nearest neighbors.** We compare kernel $K^1_{0.1}(x,y)$ with the $k$-nearest-neighbor ($k$-NN) method ($k=1,3$), with 1-NN the projection approach by which each point is projected onto the closest point in $\mathcal{C}_\epsilon$, a simple method for continuous state and action spaces [21, 27].

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

**Comparison with other algorithms.** We compare MFC-K-Q with $K^{(1)}_{0.1}$ with three representative algorithms, MFQ from [4] on MFC, Deep PPQ from [12], and PCC-VIVACE from [5] on MARL. Our experiment demonstrates superior performances of MFC-K-Q.

- When N>40, MFC-K-Q dominates all these three algorithms (Figure 4a) and it learns the bandwidth parameter $c$ most accurately (Figure 4b). Despite being the best performer when N<35, Deep PPQ suffers from the "curse of dimensionality" and the performance gets increasingly worse when N increases;
- MFC-K-Q with $K^{(1)}_{0.1}$ dominates MFQ, which is similar to our worst performer MFC-K-Q with 1-NN. In general, kernel regression performs better than simple projection (adopted in MFQ) where only one point is used to estimate $Q$;
- the decentralized PCC-VIVACE has the worst performance. Moreover, it is insensitive to the bandwidth parameter $c$. See Figure 4b.




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

Figure 4: Performance comparison among different algorithms.

# References


[1] Andrey Bernstein and Nahum Shimkin. Adaptive-resolution reinforcement learning with polynomial exploration in deterministic domains. *Machine learning*, 81(3):359–397, 2010.

[2] René Carmona and François Delarue. *Probabilistic Theory of Mean Field Games with Applications I-II*. Springer, 2018.

[3] René Carmona, Mathieu Laurière, and Zongjun Tan. Linear-quadratic mean-field reinforcement learning: Convergence of policy gradient methods. *arXiv preprint arXiv:1910.04295*, 2019.

[4] René Carmona, Mathieu Laurière, and Zongjun Tan. Model-free mean-field reinforcement learning: Mean-field MDP and mean-field Q-learning. *arXiv preprint arXiv:1910.12802*, 2019.

[5] Mo Dong, Tong Meng, Doron Zarchy, Engin Arslan, Yossi Gilad, Brighten Godfrey, and Michael Schapira. {PCC} vivace: Online-learning congestion control. In *15th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 18)*, pages 343–356, 2018.

[6] Samah El-Tantawy, Baher Abdulhai, and Hossam Abdelgawad. Multiagent reinforcement learning for integrated network of adaptive traffic signal controllers (MARLIN-ATSC): Methodology and large-scale application on downtown Toronto. *IEEE Transactions on Intelligent Transportation Systems*, 14(3):1140–1150, 2013.

[7] Eyal Even-Dar and Yishay Mansour. Learning rates for Q-learning. *Journal of machine learning Research*, 5(Dec):1–25, 2003.

[8] Wendell H Fleming and Raymond W Rishel. *Deterministic and stochastic optimal control*, volume 1. Springer Science & Business Media, 2012.

[9] Jürgen Gärtner. On the McKean-Vlasov limit for interacting diffusions. *Mathematische Nachrichten*, 137(1):197–248, 1988.

[10] Haotian Gu, Xin Guo, Xiaoli Wei, and Renyuan Xu. Dynamic programming principles for learning MFCs. *arXiv preprint arXiv:1911.07314*, 2019.

[11] Pablo Hernandez-Leal, Bilal Kartal, and Matthew E Taylor. A survey and critique of multiagent deep reinforcement learning. *Autonomous Agents and Multi-Agent Systems*, 33(6):750–797, 2019.

[12] Nathan Jay, Noga Rotman, Brighten Godfrey, Michael Schapira, and Aviv Tamar. A deep reinforcement learning perspective on internet congestion control. In *International Conference on Machine Learning*, pages 3050–3059, 2019.

[13] Junqi Jin, Chengru Song, Han Li, Kun Gai, Jun Wang, and Weinan Zhang. Real-time bidding with multi-agent reinforcement learning in display advertising. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 2193–2201, 2018.

[14] Mark Kac. Foundations of kinetic theory. In *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability*, volume 3, pages 171–197. University of California Press Berkeley and Los Angeles, California, 1956.

[15] Daniel Lacker. Limit theory for controlled McKean–Vlasov dynamics. *SIAM Journal on Control and Optimization*, 55(3):1641–1672, 2017.

[16] Martin Lauer and Martin Riedmiller. An algorithm for distributed reinforcement learning in cooperative multi-agent systems. In *In Proceedings of the Seventeenth International Conference on Machine Learning*. Citeseer, 2000.

[17] Minne Li, Zhiwei Qin, Yan Jiao, Yaodong Yang, Jun Wang, Chenxi Wang, Guobin Wu, and Jieping Ye. Efficient ridesharing order dispatching with mean field multi-agent reinforcement learning. In *The World Wide Web Conference*, pages 983–994, 2019.

[18] Yuwei Luo, Zhuoran Yang, Zhaoran Wang, and Mladen Kolar. Natural actor-critic converges globally for hierarchical linear quadratic regulator. *arXiv preprint arXiv:1912.06875*, 2019.

[19] Henry P McKean. Propagation of chaos for a class of non-linear parabolic equations. *Stochastic Differential Equations (Lecture Series in Differential Equations, Session 7, Catholic Univ., 1967)*, pages 41–57, 1967.

[20] Médéric Motte and Huyên Pham. Mean-field Markov decision processes with common noise and open-loop controls. *arXiv preprint arXiv:1912.07883*, 2019.

[21] Rémi Munos and Andrew Moore. Variable resolution discretization in optimal control. *Machine learning*, 49(2-3):291–323, 2002.

[22] Guannan Qu, Adam Wierman, and Na Li. Scalable reinforcement learning of localized policies for multi-agent networked systems. *arXiv preprint arXiv:1912.02906*, 2019.

[23] Devavrat Shah and Qiaomin Xie. Q-learning with nearest neighbors. In *Advances in Neural Information Processing Systems*, pages 3111–3121, 2018.

[24] Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. Safe, multi-agent, reinforcement learning for autonomous driving. *arXiv preprint arXiv:1610.03295*, 2016.

[25] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, and Marc Lanctot. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484, 2016.

[26] Alain-Sol Sznitman. Topics in propagation of chaos. In *Ecole d'été de Probabilités de Saint-Flour XIX-1989*, pages 165–251. Springer, 1991.

[27] Hado Van Hasselt. Reinforcement learning in continuous state and action spaces. In *Reinforcement Learning*, pages 207–251. Springer, 2012.

[28] Oriol Vinyals, Igor Babuschkin, Junyoung Chung, Michael Mathieu, Max Jaderberg, Wojciech M Czarnecki, Andrew Dudzik, Aja Huang, Petko Georgiev, and Richard Powell. Alphastar: Mastering the real-time strategy game starcraft II. *DeepMind Blog*, page 2, 2019.

[29] Lin F Yang, Chengzhuo Ni, and Mengdi Wang. Learning to control in metric space with optimal regret. *arXiv preprint arXiv:1905.01576*, 2019.

[30] Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *arXiv preprint arXiv:1911.10635*, 2019.