Scalable spectral representations for multiagent reinforcement learning in network MDPs

Zhaolin Ren[*]
Harvard University

Runyu (Cathy) Zhang[*]
Harvard University

Bo Dai
Georgia Institute of Technology

Na Li
Harvard University

Abstract

Network Markov Decision Processes (MDPs), which are the de-facto model for multi-agent control, pose a significant challenge to efficient learning caused by the exponential growth of the global state-action space with the number of agents. In this work, utilizing the exponential decay property of network dynamics, we first derive scalable spectral local representations for multiagent reinforcement learning in network MDPs, which induces a network linear subspace for the local Q-function of each agent. Building on these local spectral representations, we design a scalable algorithmic framework for multiagent reinforcement learning in continuous state-action network MDPs, and provide end-to-end guarantees for the convergence of our algorithm. Empirically, we validate the effectiveness of our scalable representation-based approach on two benchmark problems, and demonstrate the advantages of our approach over generic function approximation approaches to representing the local Q-functions.

1 Introduction

Multi-agent network systems have found applications in various societal infrastructures, such as power systems, traffic networks, and smart cities [McArthur et al., 2007, Burmeister et al., 1997, Roscia et al., 2013]. One particularly important class of such problems is the cooperative multi-agent network MDP setting, where agents are embedded in a graph, and each agent has its own local state [Qu et al., 2020b]. In network MDPs, the local state transition probabilities and rewards only depend on the states and actions of the agent's *direct neighbors* in the graph. Such a property has been observed in a great variety of cooperative network control problems, ranging from thermal control of multizone buildings [Zhang et al., 2016], wireless access control [Zocca, 2019] to phase synchronization in electrical grids [Blaabjerg et al., 2006], where agents typically only need to act and learn based on information within a local neighborhood due to constraints on the information and communication infrastructure. However, despite many efforts (c.f. [Qu et al., 2021, Lin et al., 2021a, Zhang et al., 2023c, Abdallah and Lesser, 2007, Du et al., 2022, Ma et al., 2024]), efficiently finding effective local policies for networks remains an open challenge.

Reinforcement Learning (RL) [Sutton, 2018] has emerged as a promising tool for addressing the complex dynamics of these systems [Chen et al., 2024, Nezamoddini and Gholami, 2022, Yan and Xu, 2020]. There are several pioneering works on designing scalable RL algorithms for network systems [Qu et al., 2021, Lin et al., 2021a, Zhang et al., 2023c]. To facilitate scalable control in network control, in [Qu et al., 2021], the authors introduced a key insight, referred to as the *exponential decay property* of the Q-function. This property suggests that each agent's local Q-function can be well-approximated using only information from its κ-hop neighborhood. We note that a similar property has also been proposed in [Gu et al., 2022] which focuses on reinforcement learning in the mean field multi-agent setting. Leveraging this property, the proposed algorithm concentrates on learning truncated Q-functions and then applying either policy gradient [Qu et al., 2021, Lin et al., 2021a] or policy iteration [Zhang et al., 2023c]. However, although these methods are scalable with respect to the network size, they are limited to the tabular setting, where each agent must store a local Q-table that scales with the state and action spaces of its neighborhood, making it inefficient for large state and action spaces. In fact, due to the inherent complexity in network MDPs, *i.e.*, *network size* is large and the *state and action spaces* of each agent are large or continuous, designing efficient

[*]: Equal contribution. Proceedings of the 28th International Conference on Artificial Intelligence and Statistics (AISTATS) 2025, Mai Khao, Thailand. PMLR: Volume 258. Copyright 2025 by the author(s).

and scalable RL algorithms for such systems remains a long-standing challenge.

There have been several works aimed at addressing scalability in the context of large state and action spaces. A common approach is to use function approximation to find an efficient representation of the Q-function. For instance, [Stankovic and Stankovic, 2016] explores function approximation to solve network RL problems. However, their setting is simpler than the network MDP considered here, as they assume fully decoupled agent dynamics, whereas we allow an agent's dynamics to depend on the states of neighboring agents. In the broader context of multi-agent learning, function approximation has also been widely studied [Zhang et al., 2018, Dubey and Pentland, 2021]. However, these works differ from ours: [Zhang et al., 2018] focuses on the stochastic game setting, where agents share a common global state, while [Dubey and Pentland, 2021] examines the parallel MDP setting. Additionally, outside the RL domain, there are works on network representation learning [Dong et al., 2020, Li and Pi, 2020, Zhang et al., 2020]. However, it remains unclear whether these techniques can be applied to control and RL in network systems, which presents an interesting open question for future research.

Finding a suitable representation for the Q-function is not a unique problem in network RL. It is also a central challenge in classical centralized or single-agent RL. However, picking the right class of function approximators that can represent the Q function while being sample efficient to learn is challenging. A natural approach is to use deep neural network (NN) architectures, which have great representational capacities. However, challenges to using deep NNs include sample-inefficiency (deep NNs often require huge amounts of data to train), hyperparameter sensitivity, stability of training (it is known that TD learning with nonlinear function approximation may fail to converge), and it can be difficult to pick an appropriate architecture for the problem setting at hand. One promising approach arises in the (low-rank) linear MDP setting [Jin et al., 2020a], where the transition kernel of the MDP can be represented as a linear combination of low-rank features. By applying the Bellman equation, the Q-value function can then be represented as a linear combination of these low-rank features. It has been shown in [Jin et al., 2020a] in this setting, efficient RL can be achieved, with sample complexity depending on the dimension of the feature space rather than the size of the state and action spaces. Moreover, computationally, to realize the theoretical promise of linear MDPs, there has been a line of work [Ren et al., 2022c, Ren et al., 2022a, Zhang et al., 2022] that show that the transition can be effectively approximated by a linear decomposition of nonlinear features, with strong empirical performances. Notably, [Ren et al., 2022c] explore the connection between stochastic nonlinear dynamical systems and linear MDPs, showing that under certain noise assumptions, stochastic nonlinear dynamics can be well-approximated by a linear decomposition of finite-dimensional (nonlinear) spectral features through an approach called *spectral dynamic embedding*. Building on these spectral features, [Ren et al., 2023] developed RL algorithms, with strong theoretical guarantees and empirical performances. Given the existing literature, the following question remains open:

Can we identify an appropriate representation for network MDPs and leverage it for scalability in both the size of the network and state-action space?

Our contribution Building on the existing literature, this paper addresses the critical gap by proposing a spectral dynamic embedding-based representation and developing a multi-agent RL algorithm for network systems that *scales efficiently with both network size and the complexity of state and action spaces*, while also providing provable convergence guarantees.

Our approach integrates insights from both network RL and linear MDP/representation-based approaches in centralized RL. Specifically, utilizing the exponential decay property and local nature of the transition dynamics, we show how we can approximate the local Q_i-value function linearly via *network κ-local spectral features* that factorize the κ-hop transition dynamics. Leveraging this property, we develop a scalable sample-efficient method to learn local Q-functions in continuous network MDPs, followed by policy optimization based on the learned Q-functions.

We provide rigorous sample complexity guarantees for our framework, and to the best of our knowledge, this is the first work to propose a provably efficient multi-agent RL algorithm for network systems that is scalable with respect to both network size and the size of the state and action spaces of individual agents. Finally, we validate our approach with numerical experiments on network thermal control and Kuramoto oscillator synchronization. In both cases, we find that our approach provides benefits over generic neural network function approximations, demonstrating the advantages of our spectral representation-based framework.

Notations For any vectors $v_1, \ldots, v_n \in \mathbb{R}^d$, the notation $\otimes_{i=1}^n v_i \in \mathbb{R}^{nd}$ denotes their tensor product. The inner product of two tensor products is defined as follows. Consider another set of vectors $w_1, \ldots, w_n \in \mathbb{R}^d$. Then, we denote $\langle \otimes_{i=1}^n v_i, \otimes_{i=1}^n w_i \rangle := \prod_{i=1}^n \langle v_i, w_i \rangle$. We also use the notation $[n]$ to denote the set $\{1, \ldots, n\}$ for a positive integer n. In addition, when the context is clear, for notational convenience, we may drop the time indices and denote $(s(t), a(t), s(t+1))$ as (s, a, s').

2 Problem Setup and Preliminaries

Network Markov Decision Process (MDP) We consider the network MDP model, where there are n agents associated with an underlying undirected graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N} = \{1, \ldots, n\}$ is the set of agents and $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$ is the set of edges. Each agent i is associated with state $s_i \in \mathcal{S}_i$, $a_i \in \mathcal{A}_i$ where $\mathcal{S}_i \subset \mathbb{R}^S$ and $\mathcal{A}_i \subset \mathbb{R}^A$ are bounded compact sets. At each time $t \in \mathbb{N}$, the global state of the network is denoted as $s(t) = (s_1(t), \ldots, s_n(t)) \in \mathcal{S} := \mathcal{S}_1 \times \ldots \mathcal{S}_n$. Similarly, the global actuation of the network at each time t is denoted as $a(t) = (a_1(t), \ldots, a_n(t)) \in \mathcal{A} := \mathcal{A}_1 \times \ldots \mathcal{A}_n$. We also introduce the following notations related to *κ-hop neighborhoods*. Let N_i^κ denote the set of κ-hop neighborhood of node i and define $N_{-i}^\kappa = \mathcal{N} \setminus N_i^\kappa$, i.e., the set of agents that are outside of i'th agent's κ-hop neighborhood. We write state s as $(s_{N_i^\kappa}, s_{N_{-i}^\kappa})$, i.e., the states of agents that are in the κ-hop neighborhood of i and outside of κ-hop neighborhood respectively. Similarly, we write a as $(a_{N_i^\kappa}, a_{N_{-i}^\kappa})$. When $\kappa = 1$, for simplicity we denote $N_i := N_i^1$.

We assume that the next state of each agent i depends only on the current states and actions of its neighbors, so that the probability transition admits the following factorization

$$\mathbb{P}\left(s(t+1) \mid s(t), a(t)\right) = \textstyle\prod_{i=1}^n \mathbb{P}\left(s_i(t+1) \mid s_{N_i}(t), a_{N_i}(t)\right),$$

where N_i indicates the neighbors of agent i, and $s_{N_i}(t)$ denotes the states of the neighbors of agent i at time t. Further, each agent is associated with a stage reward function $r_i(s_{N_i}, a_{N_i})$ that depends on the local state and action, and the global stage reward is $r(s,a) = \frac{1}{n}\sum_{i=1}^n r_i(s_{N_i}, a_{N_i})$; for simplicity, in the rest of our paper, we will assume that r_i depends only on (s_i, a_i), but we note that our analysis carries with minimal changes when r_i depends on (s_{N_i}, a_{N_i}). The objective is to find a (localized) policy tuple $\pi = (\pi_1, \ldots, \pi_n)$, where each $\pi_i(\cdot \mid s) \equiv \pi_i(\cdot \mid s_{N_i^{\kappa_\pi}})$ depends only on a κ_π-hop neighborhood, such that the discounted global stage reward is maximized, starting from some initial state distribution μ_0,

$$\max_\pi J(\pi) := \mathbb{E}_{s \sim \mu_0} \mathbb{E}_{a(t) \sim \pi(\cdot|s(t))} \left[\textstyle\sum_{t=0}^\infty \gamma^t r(s(t), a(t)) | s(0) = s\right].$$

Next, we give the Kuramoto oscillator synchronization problem as an example of continuous state-action network MDPs. This example will be used in our simulations in Section 5 later. For space reasons, we defer another example, that of thermal control of multi-zone buildings, to Appendix 7.2.

Example 1 (Kuramoto oscillator synchronization)**.** *The Kuramoto model [Acebrón et al., 2005, Dorfler and Bullo, 2012] is a well-known model of non-linear coupled oscillators, and has been widely applied in various fields, ranging from synchronization of neurons in the brain [Cumin and Unsworth, 2007], to synchronization of frequency of the alternating current (AC) generators or oscillators [Filatrella et al., 2008]. Concretely, we consider here a Kuramoto system with n agents, with an underlying graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N} = \{1, \ldots, n\}$ is the set of agents and $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$ is the set of edges. The state of each agent i is its phase $\theta_i \in [-\pi, \pi]$, and the action of each agent is a scalar $a_i \in \mathcal{A}_i \subset \mathbb{R}$ in a bounded subset of $\mathbb{R}$. The dynamics of each agent is influenced only by the states of its neighbors as well as its own action, satisfying the following form in discrete time [Mozafari et al., 2012]:*

$$\theta_i(t+1) = \theta_i(t) + dt \underbrace{\left(\omega_i(t) + a_i(t) + \left(\sum_{j \in N_i} K_{ij} \sin(\theta_j - \theta_i)\right)\right)}_{:=\dot{\theta}_i(t)} + \epsilon_i(t).$$

Above, ω_i denotes the natural frequency of agent i, dt is the discretization time-step, K_{ij} denotes the coupling strength between agents i and j, $a_i(t)$ is the action of agent i at time t, and $\epsilon_i(t) \sim N(0, \sigma^2)$ is a noise term faced by agent i at time t. We note that this fits into the localized transition considered in network MDPs. For the reward, we consider frequency synchronization to a fixed target ω_{target}. In this case, the local reward of each agent can be described as $r_i(\theta_{N_i}, a_i) = -\left|\dot{\theta}_i - \omega_{\text{target}}\right|$. □

To provide context for what follows, we review a few key concepts in RL. First, fixing a localized policy tuple $\pi = (\pi_1, \ldots, \pi_n)$, the Q-function for this policy π is:

$$\begin{aligned} &Q^\pi(s,a) \\ &= \frac{1}{n}\sum_{i=1}^n \mathbb{E}_{a(t) \sim \pi(\cdot|s(t))}\left[\sum_{t=0}^\infty \gamma^t r_i(s_i(t), a_i(t)) | (s(0), a(0) = (s,a)\right] \\ &:= \frac{1}{n}\sum_{i=1}^n Q_i^\pi(s,a). \end{aligned}$$

In the last step, we defined the local Q-functions $Q_i^\pi(s,a)$ which represent the Q functions for the individual reward r_i. Correspondingly, we can also define the local value function $V_i^\pi(s) = \int_a \pi(a \mid s) Q_i^\pi(s,a) da$. We note that the global $Q(s,a)$ function can be obtained by averaging n local $Q_i(s,a)$ functions. This plays an important role due to the following result known as the policy gradient theorem, which states that the policy gradient can be computed with knowledge of the $Q(s,a)$ function.

Lemma 0 ([Sutton et al., 1999])**.** *Let $d^\theta(s) = (1-\gamma)\sum_{t=0}^\infty \gamma^t \Pr(s_t = s)$ Then, we have*

$$\nabla_\theta J(\pi^\theta) = \mathbb{E}_{s \sim d^\theta, a \sim \pi^\theta(\cdot|s)}\left[Q^{\pi^\theta}(s,a) \nabla \log \pi^\theta(a|s)\right].$$

□

A natural approach to learning the $Q(s, a)$ function in the networked case is for each agent to learn its local $Q_i(s, a)$ function and share it across the network to form a global average. However, this poses a significant challenge when (i) the network size n is large, and (ii) the individual state and action spaces $\mathcal{S}_i$ and $\mathcal{A}_i$ are continuous. Even if $\mathcal{S}_i$ and $\mathcal{A}_i$ are finite, representing $Q_i(s, a)$ requires $(|\mathcal{S}_i| \times |\mathcal{A}_i|)^n$ entries, which grows exponentially with n. This challenge worsens with continuous spaces, which have infinite cardinality. To address this, we first explore the *exponential decay property* from prior work, which improves scalability with network size. We then present our main contribution: integrating the exponential decay property with spectral representations from single-agent RL to derive scalable local Q_i-value function representations for continuous state-action network MDPs. We begin by discussing the exponential decay property.

Exponential decay property. The exponential decay property [Qu et al., 2020b, Qu et al., 2020a, Lin et al., 2021b] is defined as follows.

Definition 1. *Given any $c > 0$ and $0 < \rho < 1$, the (c, ρ)-exponential decay property holds for a policy π if given any natural number κ, for any $i \in \mathcal{N}$, $s_{N_i^\kappa} \in \mathcal{S}_{N_i^\kappa}$, $s_{N_{-i}^\kappa} \in \mathcal{S}_{N_{-i}^\kappa}$, $a_{N_i^\kappa} \in \mathcal{A}_{N_i^\kappa}$, $a_{N_{-i}^\kappa} \in \mathcal{A}_{N_{-i}^\kappa}$, the local value function Q_i^π satisfies,*

$$\left|Q_i^\pi(s_{N_i^\kappa}, s_{N_{-i}^\kappa}, a_{N_i^\kappa}, a_{N_{-i}^\kappa}) - Q_i^\pi(s_{N_i^\kappa}, s'_{N_{-i}^\kappa}, a_{N_i^\kappa}, a'_{N_{-i}^\kappa})\right| \leqslant c\rho^{\kappa+1}.$$

As an immediate corollary, it follows that

$$\left|V_i^\pi(s_{N_i^\kappa}, s_{N_{-i}^\kappa}) - V_i^\pi(s_{N_i^\kappa}, s'_{N_{-i}^\kappa})\right| \leqslant c\rho^{\kappa+1}. \quad \square$$

We defer discussion about when the exponential decay property holds to Appendix 7.3. The power of the exponential decay property is that it immediately guarantees that the dependence of Q_i^π on other agents shrinks quickly as the distance between them grows, such that the true local $Q_i(s, a)$-functions can be approximated by truncated $\hat{Q}_i(s_{N_i^\kappa}, a_{N_i^\kappa})$-functions up to an error that decays exponentially with κ. The truncated $\hat{Q}_i$ function is significantly easier to represent in the finite state-action setting since each agent only needs to keep track of $(|\mathcal{S}_i| \times |\mathcal{A}_i|)^\kappa$ entries. However, continuous state and action space problems still pose a significant challenge. To overcome this, we will use the idea of spectral representations from linear MDPs and show how this can be adapted to the networked setting to yield truncated functions.

3 Spectral representations for truncated approximations of local Q_i-value functions

To recap, the key question we face is this: how can we derive scalable local Q_i-value function representations in network problems with continuous state-action spaces, and integrate them into a scalable control framework? This forms the main contribution of our work. In this section, we tackle this question by demonstrating that the spectral representation of local transition kernels provides an effective representation for the local Q_i-value functions (see Lemma 3 below).

We first motivate our analysis by reviewing representation learning in centralized RL via *spectral decompositions* [Jin et al., 2020b, Ren et al., 2022b]. From such works, we know that if the global $P(s' \mid s, a)$ admits a linear decomposition in terms of some spectral features $\phi(s, a)$ and $\mu(s')$, then the $Q(s, a)$-value function can be linearly represented in terms of the spectral features $\phi(s, a)$. In the case of representing local Q_i-functions, this property can be stated as follows.

Lemma 1 (Representing local Q_i-value functions via spectral decomposition of P (Linear MDP in [Jin et al., 2020b])). *Suppose the probability transition $P(s' \mid s, a)$ of the next state s' given the current (s, a) pair can be linearly decomposed as $P(s' \mid s, a) = \phi(s, a)^\top \mu(s')$ for some features $\phi(s, a) \in \mathbb{R}^D$ and $\mu(s') \in \mathbb{R}^D$, which we also refer to as spectral representations. Then, the local Q_i-value function admits the linear representation*

$$Q_i^\pi(s, a) = \tilde{\phi}_i(s, a)^\top w_i^\pi,$$

where

$$\tilde{\phi}_i(s, a) := [r_i(s_i, a_i), \phi(s, a)],$$
$$w_i^\pi = [1, \gamma \int_{s'} \mu(s') V_i^\pi(s') ds']^\top. \quad \square$$

Remark 1. *We note that Lemma 1 requires the assumption of the existence of a linear decomposition of the transition kernel. One significant such example of where a linear decomposition of the transition kernel is possible was discussed in [Ren et al., 2023]. The authors in [Ren et al., 2023] showed that for a wide class of stochastic control setting with Gaussian noise (or more generally, noise which take the form of a positive-definite kernel), the transition kernel admits an exact (but infinite-dimensional) linear decomposition. By considering a finite-dimensional truncation of these infinite-dimensional features, it can be shown that the linear decomposition holds approximately with finite-dimensional features, and rigorous approximation error bounds can be shown for these finite-dimensional truncations, as shown later in Lemma 5 of our paper as well as [Ren et al., 2023].*

More generally, a linear decomposition of the kernel also exists when the transition displays a particular latent variable structure [Ren et al., 2022a]. □

The benefit of the spectral decomposition property is that the Q_i-value functions can be represented by a $(D+$

1)-dimensional representation $\tilde{\phi}_i(s,a)$ comprising the spectral representation $\phi(s,a) \in \mathbb{R}^D$ and local reward $r_i(s_i, a_i) \in \mathbb{R}$; as demonstrated in [Jin et al., 2020a], under appropriate normalization conditions on the norm of the features, the sample complexity of using RL using such features will only depend polynomially on the feature dimension D, rather than the number of the states and actions.. However, applying this result directly in the networked case poses significant challenges, since the required feature dimension $D+1$ may be high. To see why this is the case, recall that the probability transition in our networked setting admits the following factorization:

$$\mathbb{P}(s' \mid s, a) = \prod_{i=1}^n \mathbb{P}(s_i' \mid s_{N_i}, a_{N_i}).$$

Suppose for the sake of discussion that each agent's transition probability satisfies the following Property 1, which states that the local transition has an exact d-dimensional spectral/linear decomposition. As discussed earlier in Remark 1, such a property can be expected to hold (at least approximately) for many problems, in particular stochastic dynamics under Gaussian noise.

Property 1. *For any $i \in [n]$ and any state-action-next state tuple (s, a, s'), there exist features $\bar{\phi}_i(s_{N_i}, a_{N_i}) \in \mathbb{R}^d$ and $\bar{\mu}_i(s_i') \in \mathbb{R}^d$ such that*

$$\mathbb{P}\left(s_i' \mid s_{N_i}, a_{N_i}\right) = \langle \bar{\phi}_i(s_{N_i}, a_{N_i}), \bar{\mu}_i(s_i') \rangle. \qquad \square$$

Given the factorization of the dynamics, this implies that

$$\mathbb{P}\left(s' \mid s, a\right) = \prod_{i=1}^n \langle \bar{\phi}_i(s_{N_i}, a_{N_i}), \bar{\mu}_i(s_i') \rangle$$
$$= \left\langle \bigotimes_{i=1}^n \bar{\phi}_i(s_{N_i}, a_{N_i}), \bigotimes_{i=1}^n \bar{\mu}_i(s_i') \right\rangle := \left\langle \bar{\phi}(s,a), \bar{\mu}(s') \right\rangle.$$

We note that the above expression of $P(s' \mid s, a)$ as an inner product between two tensor products follows from the definition of the tensor product introduced earlier in our notations; in particular, note that $\bar{\phi}(s,a)$ and $\bar{\mu}(s')$ are both in the space $\mathbb{R}^{d^n}$. Intuitively, to interpret why the tensor product arises, we note that in the absence of any structure and correlations between the agents, the features factorizing the transition of the entire network are essentially the outer/tensor product of the local transition-factorization features of the n individual agents. Agnostically, Property 1 means that representing the global network dynamics may require using the d^n-dimensional features $\bar{\phi}(s,a) := \bigotimes_{i=1}^n \bar{\phi}_i(s_{N_i}, a_{N_i})$, which even for small d is undesirable due to an exponential dependence on the network size n.

While the exponential decay property suggests that the $\hat{Q}_i$-function can be approximated by considering a κ-hop neighborhood of agent i, it is unclear how we can combine this with the spectral decomposition property to derive scalable representations for the local Q_i-value functions.

To resolve this, we combine insights from both the exponential decay and spectral decomposition property, which intuitively, suggests that what matters in determining $Q_i^\pi(s,a)$ is the probability transition dynamics within a κ-hop neighborhood of agent i. In fact, due to the local factorization property of the dynamics, the evolution of κ-hop neighborhood only depends on the $\kappa+1$-hop neighborhood, which, when Property 1 holds, admits the following spectral decomposition.

Property 2 (Network κ-local spectral features)**.** *For any $i \in [n]$ and any state-action-next state tuple (s, a, s'), there exist some positive integer $d_{i,\kappa}$ and features $\phi_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}}) \in \mathbb{R}^{d_{i,\kappa}}$ and $\mu_{i,\kappa}(s'_{N_i^\kappa}) \in \mathbb{R}^{d_{i,\kappa}}$ such that*

$$\mathbb{P}\left(s'_{N_i^\kappa} \mid s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}}\right) = \langle \phi_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}}), \mu_{i,\kappa}(s'_{N_i^\kappa}) \rangle. \qquad \square$$

Property 2 is a statement that a linear decomposition of the κ-hop neighborhood transition kernel for any agent i is possible. As we see in Lemma 3 later, the features $\phi_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}}) \in \mathbb{R}^{d_{i,\kappa}}$ that arise in this decomposition can represent the local Q_i-value function with error decaying exponentially in κ. While Property 2 can be viewed as a standalone property independent of Property 1, the following lemma shows that when Property 1 is true, Property 2 automatically holds, with $\phi_{i,\kappa}$ and $\mu_{i,\kappa}$ given by appropriate tensor products of the original $\bar{\phi}_i$ and $\bar{\mu}_i$ from the factorization of the local dynamics. We defer the proof to Appendix 7.4.

Lemma 2. *Property 2 holds whenever Property 1 holds, by setting*

$$\phi_{i,\kappa}(s_{N_i^{\kappa+1}}(t), a_{N_i^{\kappa+1}}(t)) := \bigotimes_{j \in N_i^\kappa} \bar{\phi}_j(s_{N_j}(t), a_{N_j}(t)),$$
$$\mu_{i,\kappa}(s_{N_i^\kappa}(t+1)) := \bigotimes_{j \in N_i^\kappa} \bar{\mu}_j(s_j(t+1)). \qquad \square$$

Remark 2. *While the tensor product representation in Lemma 2 can be used to give a factorization of the κ-transition dynamics in Property 2, for specific problems, there may exist problem-specific alternative $\phi_{i,\kappa}$ and $\mu_{i,\kappa}$ features that may be lower-dimensional and thus more tractable to use.* □

Property 2 presents us with a path towards scalable representation of the local Q_i via factorization of the local κ-hop neighborhood dynamics and approximating $Q_i(s,a)$ by network local representations. We first formalize this in the case when the spectral decomposition is exact and error-free. When this holds, we have the following lemma which shows how $Q_i^\pi(s,a)$ can be approximated by network local representations. We defer the proof to Appendix 7.5.1.

Lemma 3 (Local Q_i approximation via network κ-local spectral features)**.** *Suppose the (c, ρ)-exponential decay property holds. Suppose Property 2 also holds. Then, for any (s, a) pair, agent i, and natural number κ, there exists an approximation $\bar{Q}_i^\pi$ which depends linearly* *on network κ-local spectral features $\tilde{\phi}_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})$, such that*

$$\left|Q_i^\pi(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}}, s_{N_{-i}^{\kappa+1}}, a_{N_i^{\kappa+1}}) - \bar{Q}_i^\pi(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})\right| \leqslant 2c\gamma\rho^{\kappa+1},$$

where $\bar{Q}_i^\pi(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}}) = \left\langle \tilde{\phi}_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}}), w_{i,\kappa}^\pi(s'_{N_i^\kappa})\right\rangle$, *with the definitions*

$$\tilde{\phi}_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}}) := [r_i(s_i, a_i), \phi_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})]^\top,$$

$$w_{i,\kappa}^\pi(s'_{N_i^\kappa}) := [1, \gamma \int_{s'_{N_i^\kappa}} ds'_{N_i^\kappa} \mu_{i,\kappa}(s'_{N_i^\kappa}) \bar{V}_i^\pi(s'_{N_i^\kappa})]^\top,$$

where $\bar{V}_i^\pi(s'_{N_i^\kappa}) := \int_{s'_{N_{-i}^\kappa}} \frac{ds'_{N_{-i}^\kappa}}{\mathrm{Vol}(\mathcal{S}_{N_{-i}^\kappa})} V_i^\pi(s'_{N_i^\kappa}, s'_{N_{-i}^\kappa})$. □

Approximation. In general, it may be impossible to find $\phi_{i,\kappa}$ and $\mu_{i,\kappa}$ that can exactly factorize the transition kernel. However, in both the unknown-model and the known-model cases, there exist ways to approximate the kernel. In the model-free case, we may leverage representation-learning techniques to approximate the spectrum of the κ-hop transition kernel, such as the spectral decomposition in [Ren et al., 2022b] which seeks to approximate the SVD of the kernel. In the model-based case, in the case when the local transition evolves according to a known dynamics function subject to Gaussian noise, we may approximate the kernel by random or Nystrom features [Ren et al., 2023]. We provide below a unified analysis for the error bound of approximating $Q_i(s, a)$ in terms of network κ-local representations $\hat{\phi}_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})$ that approximately factorize $P(s_{N_i^\kappa} \mid s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})$.

Lemma 4. *For any distribution ν^o over the space $\mathcal{S}_{N_i^{\kappa+1}} \times \mathcal{A}_{N_i^{\kappa+1}}$, suppose there exists a network κ-local representation $\hat{\phi}_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}}) \in \mathbb{R}^m$ for which there exists $\hat{\mu}(s'_{N_i^\kappa}) \in \mathbb{R}^m$ such that for every $i \in [n]$, the following holds for some approximation error $\epsilon_P > 0$:*

$$\mathbb{E}_{\nu^o}\left[\int_{s'_{N_i^\kappa}} \left|\Delta_{i,\kappa}\left((s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}}), s'_{N_i^\kappa}\right)\right| ds'_{N_i^\kappa}\right] \leqslant \epsilon_P, \quad (1)$$

where

$$\Delta_{i,\kappa}\left((s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}}), s'_{N_i^\kappa}\right) := P(s'_{N_i^\kappa} | s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}}) - \hat{\phi}_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})^\top \hat{\mu}_{i,\kappa}(s'_{N_i^\kappa}).$$

Then, by setting $\tilde{\phi}_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}}) := [r_i(s_i, a_i), \hat{\phi}_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})]^\top$, *for every* $i \in [n]$,

$$\min_{w \in \mathbb{R}^{m+1}} \mathbb{E}_{\nu^o}\left[\left|\bar{Q}_i^\pi(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}}) - \tilde{\phi}_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})^\top w\right|\right] \leqslant \frac{\epsilon_P \gamma \bar{r}}{1-\gamma} \quad (2)$$

□

Proof. Suppose (1) holds. Then, define $w^* := [1, \gamma \int_{\mathcal{S}_{N_i^\kappa}} \hat{\mu}_{i,\kappa}(s'_{N_i^\kappa}) \bar{V}_i^\pi(s'_{N_i^\kappa}) ds'_{N_i^\kappa}]^\top \in \mathbb{R}^{m+1}$. Then, by using the upper bound $\left|\bar{V}_i^\pi(s'_{N_i^\kappa})\right| \leqslant \frac{\bar{r}}{1-\gamma}$, we have

$$\min_{w \in \mathbb{R}^{m+1}} \mathbb{E}_{\nu^o}\left[\left|\bar{Q}_i^\pi(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}}) - \tilde{\phi}_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})^\top w\right|\right] \leqslant \frac{\epsilon_P \gamma \bar{r}}{1-\gamma} \quad (3)$$

□

The approximation error in the bound above relies on the condition in (1) to hold. In the case when the local transition evolves according to a known dynamics function subject to a positive-definite kernel noise (e.g. Gaussian noise), we may approximate the κ-hop transition kernel with random features such that (1) holds with high probability. For clarity of exposition, we focus on the approximation error of random features for Gaussian kernels [Rahimi and Recht, 2007][1]. In this case, our error bound is shown in the following result, whose proof we defer to Appendix 7.5.2.

Lemma 5. *Fix any $i \in [n]$. Suppose the local dynamics take the form $s'_i = f_i(s_{N_i}, a_{N_i}) + \epsilon_i$ where $\epsilon_i \sim N(0, \sigma^2 I_S)$, such that for any κ, $s'_{N_i^\kappa} = f_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}}) + \epsilon_{N_i^\kappa}$ where $\epsilon_{N_i^\kappa} \sim N(0, \sigma^2 I_{|N_i^\kappa|S})$ and $f_{i,\kappa}$ is concatenation of f_j for $j \in N_i^\kappa$. Fix any $0 \leqslant \alpha < 1$. Then, for a positive integer m, define the m-dimensional features $\hat{\phi}_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}}) \in \mathbb{R}^m$, where*

$$\hat{\phi}_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}}) := \frac{g_\alpha(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})}{\alpha^{|N_i^\kappa|S}} \times \left\{\sqrt{\frac{2}{m}} \cos\left(\frac{\omega_\ell^\top f_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})}{\sqrt{1-\alpha^2}} + b_\ell\right)\right\}_{\ell=1}^m,$$

with $\{\omega_\ell\}_{\ell=1}^m$ being i.i.d draws from $N(0, \sigma^{-2} I_{|N_i^\kappa|S})$, $\{b_\ell\}_{\ell=1}^m$ being i.i.d draws from $\mathrm{Unif}([0, 2\pi])$, and

$$g_\alpha(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}}) := \exp\left(\frac{\alpha^2 \left\|f_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})\right\|^2}{2(1-\alpha^2)\sigma^2}\right).$$

In addition, define

$$\tilde{g}_\alpha := \max_{i \in [n]}\left(\sup_{x \in f_{i,\kappa}(\mathcal{S}_{N_i^{\kappa+1}}, \mathcal{A}_{N_i^{\kappa+1}})} \frac{g_\alpha(x)}{\alpha^{|N_i^\kappa|S}}\right).$$

Suppose

$$m = \Omega\left(\max_{i \in [n]}\left[\log\left(\frac{(|N_i^\kappa|S(diam(\mathcal{S}_{N_i^\kappa}))^2)}{\sigma^2(\delta/n)(\epsilon_P/\tilde{g}_\alpha)}\right) \frac{|N_i^\kappa|S\tilde{g}_\alpha^2}{\epsilon_P^2}\right]\right)$$

[1] We note that our result easily generalizes to any positive-definite transition kernel noise (e.g. Laplacian, Cauchy, Matérn, etc; see Table 1 in [Dai et al., 2014] for more examples)

for some $\delta > 0$. Then, with probability at least $1-\delta$, the condition in (1) holds for every $i \in [n]$ and any distribution ν^o over $\mathcal{S} \times \mathcal{A}$, with

$$\hat{\mu}_{i,\kappa}(s'_{N_i^\kappa}) := \left\{ \sqrt{\frac{2}{m}} p_\alpha(s'_{N_i^\kappa}) \cos(\sqrt{1-\alpha^2}\omega_\ell^\top s'_{N_i^\kappa} + b_\ell) \right\}_{\ell=1}^m ,$$

where $p_\alpha(s'_{N_i^\kappa}) := \frac{\alpha^{|N_i^\kappa|S}}{(2\pi\sigma^2)^{|N_i^\kappa|S}} \exp(-\frac{\left\|\alpha s'_{N_i^\kappa}\right\|^2}{2\sigma^2})$ *is a Gaussian distribution with standard deviation* $\frac{\sigma}{\alpha}$. □

The key takeaway from the above result is that under Gaussian noise and known reward and dynamics function, there exists finite-dimensional features that can, with high probability, approximately factorize the local κ-transition kernels, satisfying the condition in (1) with high probability. Moreover, from this result, we note that the required number of features to achieve this is $\tilde{O}\left(\frac{\max_{i \in [n]} |N_i^\kappa| S \tilde{g}_\alpha^2}{\epsilon_P^2}\right)$, which only depends on the dimension of states in the largest κ-hop neighborhood. We note that the tunable α in Lemma 5 allows greater flexibility and may be tuned to improve empirical performance [Ren et al., 2023].

4 Algorithms

As suggested in Lemma 3, $\widetilde{\phi}_{i,\kappa}$ serves as a good representation for the local Q_i-functions. Based upon this observation, this section focuses on how the local Q_i-function and subsequently a good localized policy can be learned. The algorithm contains three major steps: **feature generation**, **policy evaluation** and **policy gradient**.

The first step is **feature generation** (Lines 1 through 3), where we generate the appropriate features $\widetilde{\phi}_{i,\kappa}$. This comprises the local reward function as well as the spectral features $\hat{\phi}_{i,\kappa}(s_{N_i^\kappa}, a_{N_i^\kappa})$ coming from the factorization of the local κ-hop dynamics. In the case of known dynamics and Gaussian noise, we know from Lemma 5 that $\hat{\phi}_{i,\kappa}(s_{N_i^\kappa}, a_{N_i^\kappa})$ can be derived by random features which factorize the local κ-hop dynamics with high probability. In this case, we note that our spectral features are scalable with respect to both the network size and the continuous state-action space, since the required number of features only depend on the dimensions of the κ-hop neighborhoods.

The second step is **policy evaluation**, where we use the feature $\widetilde{\phi}_{i,\kappa}$ and apply LSTD to find a set of weights w_i to approximate the local Q_i-functions by $\hat{Q}_i = \widetilde{\phi}_{i,\kappa}^\top \hat{w}_i$. At each round $k \in [K]$, we first sample M_s samples from the stationary distribution of $\pi^{(k)}$ (Line 5), and then perform a LSTD update for each agent $i \in [n]$ to learn the appropriate weights for the local Q_i-functions (Line 6).

Finally, the last step is updating policy using **policy gradient** (Lines 6 to 7). For each agent $i \in [n]$, with the learned $\{\hat{Q}_j\}_{j \in N_i^{\kappa_\pi + \kappa}}$, we perform a gradient step to update the local policy weights θ_i, and update to the new policy. We note that this update is scalable since from the perspective of each agent i, it only requires knowledge of the local $\hat{Q}_j$ for agents j in a $(\kappa_\pi + \kappa)$-hop neighborhood of agent i. In practice, the κ-hop spectral representation we introduce can be combined flexibly with any actor in a distributed cooperative actor-critic framework that requires knowledge of the local Q_i-functions.

4.1 Policy evaluation error

We have the following result on the policy evaluation error with our network κ-local features. We defer the details of the proof (including preliminary results required for the proof) to Appendix 7.6.

Lemma 6 (Policy Evaluation Error). *Suppose equation (1) in Lemma 4 holds. Suppose the sample size $M_s \geqslant \log\left(\frac{2(m+1)}{\delta/(Kn)}\right)$. Then, with probability at least $1-2\delta$, for every $i \in [n]$ and $k \in [K]$, the ground truth Q function $Q_i^{\pi(k)}(s,a)$ and the truncated Q function learnt in Algorithm 1 $\hat{Q}_i(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})$ satisfies, for any distribution $\bar{\nu}$ on $\mathcal{S} \times \mathcal{A}$,*

$$\begin{aligned}
&\mathbb{E}_{\bar{\nu}}\left[\left|Q_i^{\pi(k)}(s,a) - \hat{Q}_i^{(k)}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})\right|\right] \\
&\leqslant O\Bigg(\underbrace{c\rho L^2 D \rho^{\kappa+1}}_{\text{truncation error}} + \underbrace{\log\left(\frac{(m+1)}{\delta/(Kn)}\right)\frac{D^2 L^5}{\sqrt{M_s}}}_{\text{statistical error}} \\
&\qquad + \underbrace{L\epsilon_P\left(\left\|\frac{\bar{\nu}}{\nu^o}\right\|_\infty + \left\|\frac{\nu_{\pi(k)}}{\nu^o}\right\|_\infty\right)}_{\text{approximation error}}\Bigg),
\end{aligned}$$

where $D := \max_{i\in[n],k\in[K]} \left\|(M_i^{(k)})^{-1}\right\|, \quad L := \max_{i\in[n]} \|\tilde{\varphi}_{i,\kappa}\|$

and $M_i^{(k)}$ is defined as in equation (4). □

From the above result, we note that the policy evaluation error comprises three components, with one being the statistical error due to using finite samples, which decays with the square root of the sample size M_s, and the truncation error from considering a truncated κ-hop neighborhood (this decays exponentially in κ), as well as the approximation error of the spectral features in approximating the κ-hop transition (ϵ_P).

4.2 Policy optimization error and main convergence result

Theorem 1. *Suppose the sample size $M_s \geqslant \log\left(\frac{2(m+1)}{(\delta/Kn)}\right)$. Suppose with probability at least $1-\delta$, for all $i \in [n]$, the following holds for some features $\hat{\phi}_{i,\kappa} \in \mathbb{R}^m$ and $\hat{\mu}_{i,\kappa} \in \mathbb{R}^m$:*

$$\mathbb{E}_{\nu^o}\left[\int_{s'_{N_i^\kappa}} \left|\Delta_{i,\kappa}\left((s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}}), s'_{N_i^\kappa}\right)\right| ds'_{N_i^\kappa}\right] \leqslant \epsilon_P,$$

Algorithm 1: Networked control with spectral embedding

```
Data: $Q$-value truncation radius $\kappa$, Policy truncation radius $\kappa_\pi$, Reward Function $r(s,a)$, Number of features $m$, Number of samples/round $M_s$, Learning Rate $\eta$, Number of rounds K
Result: $\pi^{(K+1)} = (\pi_1^{(K+1)}, \dots, \pi_n^{(K+1)})$
Spectral dynamic embedding generation
for $i \in [n]$ do
    Generate features $\tilde{\phi}_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}}) := [r_i(s_i,a_i), \hat{\phi}_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})] \in \mathbb{R}^{m+1}$, where $\hat{\phi}_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})$ satisfies the condition in (1).
end
Policy evaluation and update
for $k = 1, 2, \cdots, K$ do
    Least squares policy evaluation
    Set $\pi_i^{(k)} := \pi_i^{\theta_i^{(k+1)}}$. Sample i.i.d. $D_k = \{(s(j), a(j), s'(j)), a'(j)\}_{j \in [M_s]}$ where $(s(j), a(j)) \sim \nu_{\pi^{(k)}}$, $s'(j) \sim P(\cdot \mid s(j), a(j))$ where $\nu_{\pi^{(k)}}$ is the stationary distribution of $\pi^{(k)}$, and $\forall j \in [M_s], \forall i \in [n] : a_i'(j) \sim \pi_i^{(k)}(\cdot \mid s'_{N_i^{\kappa_\pi}}(j))$
    for $i \in [n]$ do
        Solve $\hat{w}_i^{(k)}$ using least square temporal difference (LSTD) as follows:
            $\hat{w}_i^{(k)} = (M_i^{(k)})^{-1} H_i^{(k)} r_i$
            $M_i^{(k)} := \frac{1}{|D_k|} \sum_{s,a,s',a' \in D_k} \tilde{\varphi}_{i,\kappa} (\tilde{\varphi}_{i,\kappa} - \gamma \tilde{\varphi}'_{i,\kappa})^\top$   (4)
            $H_i^{(k)} := \frac{1}{|D_k|} \sum_{s,a,s',a' \in D_k} \tilde{\varphi}_{i,\kappa} \tilde{\varphi}_{i,\kappa}^\top$
            $r_i := [1, 0, 0, \dots, 0]^\top \in \mathbb{R}^{m+1}$
        where
            $\tilde{\varphi}_{i,\kappa}(j) := \tilde{\phi}_{i,\kappa}(s_{N_i^{\kappa+1}}(j), a_{N_i^{\kappa+1}}(j))$,
            $\tilde{\varphi}'_{i,\kappa}(j) := \tilde{\phi}_{i,\kappa}(s'_{N_i^{\kappa+1}}(j), a'_{N_i^{\kappa+1}}(j))$.
        Update approximate $\hat{Q}_i^{(k)}$-value function as $\hat{Q}_i(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}}) := \tilde{\phi}_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})^\top \hat{w}_i^{(k)}$.
    end
    Policy gradient for control
    for $i \in [n]$ do
        Calculate
            $\hat{g}_i^{(k)} = \frac{1}{M_s} \sum_{j=1}^{M_s} \sum_{\ell \in N_i^{\kappa+\kappa_\pi}} \frac{\hat{Q}_\ell^{(k)}(s_{N_\ell^\kappa}(j), a_{N_\ell^\kappa}(j))}{n} \times \nabla_{\theta_i} \log \pi_i^{(\theta_i^{(k)})}(a_i(j) \mid s_{N_i^{\kappa_\pi}}(j))$
        Take gradient step $\theta_i^{(k+1)} = \theta_i^{(m)} + \eta \hat{g}_i^{(k)}$
    end
end
```

for some $\epsilon_P > 0$ and distribution ν^o over $\mathcal{S} \times \mathcal{A}$, where

$$\Delta_{i,\kappa}\left((s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}}), s'_{N_i^\kappa}\right) := P(s'_{N_i^\kappa} | s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}}) - \hat{\phi}_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})^\top \hat{\mu}_{i,\kappa}(s'_{N_i^\kappa}).$$

Then, if $\eta = O(1/\sqrt{K})$, with probability at least $1 - 4\delta$,

$$\frac{1}{K} \sum_{k=1}^{K} \left\| \nabla J(\theta^{(k)}) \right\|^2 \leqslant O\left(\frac{\bar{r}/(1-\gamma)}{\sqrt{K}} + \frac{L_\pi \bar{r} \epsilon_J}{1-\gamma} + \frac{L'}{\sqrt{K}} \left(\epsilon_J^2 + \left(\frac{L_\pi \bar{r}}{1-\gamma} \right)^2 \right) \right)$$

where $\epsilon_J := 2cL_\pi \rho^\kappa + \frac{2\bar{r}L_\pi}{1-\gamma} \sqrt{\frac{1}{M_s} \log\left(\frac{d_\theta + 1}{\delta/K}\right)} + \epsilon_Q L_\pi$, and

$$\epsilon_Q := \max_{k \in [K]} O\left(c\rho L^2 D \rho^{\kappa+1} + \log\left(\frac{(m+1)}{\delta/(Kn)}\right) \frac{D^2 L^5}{\sqrt{M_s}} + L\epsilon_P \left(\left\| \frac{\hat{\nu}^{(k)}}{\nu^o} \right\|_\infty + \left\| \frac{\nu_{\pi^{(k)}}}{\nu^o} \right\|_\infty \right) \right).$$

where L' is the Lipschitz continuity parameter of $\nabla J(\theta)$, $L_{i,\pi}$ is a bound on $\left\| \nabla_{\theta_i} \log \pi_i^{\theta_i}(\cdot \mid \cdot) \right\|$, and $L_\pi := \sqrt{\sum_{i=1}^n L_{i,\pi}^2}$. □

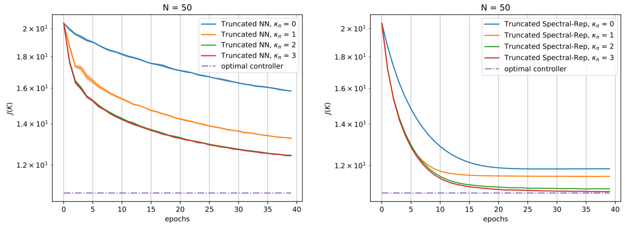
From the above result, we see that our algorithm can achieve convergence to an approximate stationary point of the global objective J as the number of rounds K increases, up to an error term depending on ϵ_J, which depends on the policy evaluation error ϵ_Q from Lemma 6. As we observed before, the policy evaluation error comprises a statistical error, a truncation error decaying exponentially as κ increases, and a feature approximation error term ϵ_P. Consequently, the convergence error to an approximation stationary point also depends on these three terms.

5 Simulations

5.1 Thermal control of multi-zone buildings

We consider a stochastic linear dynamical system modeling the thermal control of a 50-zone building. We assume that the network is connected, with each agent having 2 neighbors. The dynamics of each agent is only affected by its neighbors, and subject to Gaussian noise. We also assume access to the model dynamics and reward function. In this problem, to implement our algorithm, we utilize random features that factorize the κ-hop Gaussian transition (cf. Lemma 5), and perform least squares, followed by normalized gradient descent. The controller is parameterized to be linear. More details on our experimental setup can be found in Appendix 7.9.

Since the dynamics are assumed to be linear, we have access to the cost of the optimal controller, making this a good way to benchmark the performance of our algorithm. The performance of our algorithm is shown in Figure 1 below. As we can see, as κ_π increases, our algorithm is indeed able to approximate the performance of the optimal controller. Moreover, the speed at which it converges is faster than $\hat{Q}_i$ approximations that leverage a generic two-hidden layer neural network (NN) to represent the (truncated) local $\hat{Q}_i$ value functions; we note that in both cases, the algorithms utilize the same learning rate for the policy gradient step, and have access to the rewards and dynamics function.



(a) NN critic (b) Spectral features critic

Figure 1: Learning trajectories of cost (lower is better) using Algorithm 1 + random features and NN critics on a 50-dimensional stochastic linear dynamical system for varying κ_π. Average and 1 std confidence intervals over 5 seeds.

5.2 Kuramoto oscillator control

Due to the nonlinearity in this problem, we adopt the Soft-Actor-Critic (SAC) framework for this problem, and compare the performance of a generic deep NN critic with our network spectral local-κ critic. In this problem, we consider the more realistic and difficult setting where the dynamics is unknown. In this problem, the network has 40 agents in total, and the network graph is connected, with each agent having 2 neighbors. We set the goal for the agents to synchronize to a target frequency of 0.2.

In both the generic SAC and our spectral SAC implementation, the local critic for Q_i-value function considers a κ-hop neighborhood, i.e. approximate Q_i by $\hat{Q}_i(s_{N_i^\kappa}, a_{N_i^\kappa}) = \hat{\phi}_i(s_{N_i^\kappa}, a_{N_i^\kappa})^\top w_i$, where $\hat{\phi}_i(s_{N_i^\kappa}, a_{N_i^\kappa})$ is a two-hidden layer neural network. However, for our approach (spectral + SAC), we add a feature step that regularizes the feature $\hat{\phi}_i(s_{N_i^\kappa}, a_{N_i^\kappa})$ towards factorizing the local dynamics, i.e. minimizing the objective in Condition 1 in Lemma 4. We defer more details on the problem setup as well as experimental details to Appendix 7.9.

In Figure 2, we compare the performance of our approach (Spectral + SAC) with a generic SAC with two-hidden layer NN critic. As we can see, our approach leads to significantly higher rewards. Moreover, we observe that our approach leads to qualitatively better synchronization behavior when starting from the same initial condition, as indicated in Figure 3. Finally, we note that in the model-based setting, our algorithm (utilizing random features) achieves a performance comparable to that of generic NN approaches. The model-based results are deferred to Appendix 7.9.

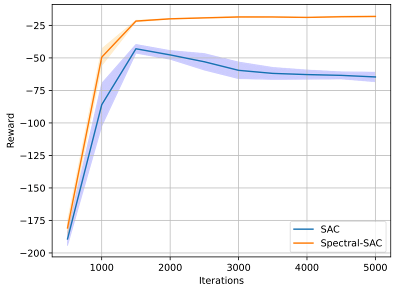
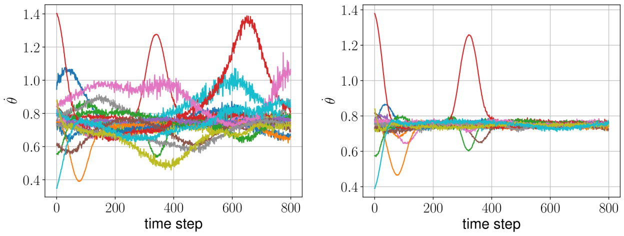


Figure 2: Change in reward during training for Kuramoto oscillator control, $N = 20$, $\kappa_\pi = 1, \kappa = 2$. The performance for each algorithm is averaged over 5 seeds.



(a) SAC Controller (b) Spectral+ SAC Controller

Figure 3: Synchronization of frequency ($\dot{\theta}$) under SAC and Spectral + SAC controller, for 800 time steps with time interval $dt = 0.01$.

6 Conclusion

In this work, utilizing local spectral representations, we provide the first provably efficient algorithm for scalable network control in continuous state-action spaces. We validate our results numerically, where we find that utilizing κ-local spectral features can achieve effective control on a thermal network control problem as well as a Kuramoto nonlinear coupled oscillator control problem. Moreover, in both cases, we demonstrate that our approach has benefits over generic neural network approximations for local Q_i-value functions. Collectively, our theoretical and empirical results demonstrate the validity and importance of a representation-based viewpoint in achieving more effective and scalable control in continuous state-action network MDPs.

Acknowledgements

We thank the reviewers for their helpful comments. Zhaolin Ren, Runyu (Cathy) Zhang and Na Li are supported by NSF ASCENT 2328241, NSF AI institute: 2112085, and NSF ECCS-2401390. Bo Dai is supported by NSF ECCS 2401391, NSF IIS 2403240, as well as Dolby.

References

[Abdallah and Lesser, 2007] Abdallah, S. and Lesser, V. (2007). Multiagent reinforcement learning and self-organization in a network of agents. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, pages 1–8, Honolulu Hawaii. ACM.

[Acebrón et al., 2005] Acebrón, J. A., Bonilla, L. L., Pérez Vicente, C. J., Ritort, F., and Spigler, R. (2005). The kuramoto model: A simple paradigm for synchronization phenomena. *Reviews of modern physics*, 77(1):137–185.

[Bamieh et al., 2002] Bamieh, B., Paganini, F., and Dahleh, M. (2002). Distributed control of spatially invariant systems. *IEEE Transactions on Automatic Control*, 47(7):1091–1107.

[Blaabjerg et al., 2006] Blaabjerg, F., Teodorescu, R., Liserre, M., and Timbus, A. V. (2006). Overview of control and grid synchronization for distributed power generation systems. *IEEE Transactions on industrial electronics*, 53(5):1398–1409.

[Burmeister et al., 1997] Burmeister, B., Haddadi, A., and Matylis, G. (1997). Application of multi-agent systems in traffic and transportation. *IEE Proceedings-Software*, 144(1):51–60.

[Chen et al., 2024] Chen, D., Zhang, K., Wang, Y., Yin, X., Li, Z., and Filev, D. (2024). Communication-Efficient Decentralized Multi-Agent Reinforcement Learning for Cooperative Adaptive Cruise Control. *IEEE Transactions on Intelligent Vehicles*, pages 1–14. Conference Name: IEEE Transactions on Intelligent Vehicles.

[Claus and Boutilier, 1998] Claus, C. and Boutilier, C. (1998). The dynamics of reinforcement learning in cooperative multiagent systems. *AAAI/IAAI*, 1998(746-752):2.

[Cumin and Unsworth, 2007] Cumin, D. and Unsworth, C. (2007). Generalising the kuramoto model for the study of neuronal synchronisation in the brain. *Physica D: Nonlinear Phenomena*, 226(2):181–196.

[Dai et al., 2014] Dai, B., Xie, B., He, N., Liang, Y., Raj, A., Balcan, M.-F. F., and Song, L. (2014). Scalable kernel methods via doubly stochastic gradients. *Advances in neural information processing systems*, 27.

[Dong et al., 2020] Dong, Y., Hu, Z., Wang, K., Sun, Y., and Tang, J. (2020). Heterogeneous Network Representation Learning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, pages 4861–4867, Yokohama, Japan. International Joint Conferences on Artificial Intelligence Organization.

[Dorfler and Bullo, 2012] Dorfler, F. and Bullo, F. (2012). Synchronization and transient stability in power networks and nonuniform kuramoto oscillators. *SIAM Journal on Control and Optimization*, 50(3):1616–1642.

[Du et al., 2022] Du, Y., Ma, C., Liu, Y., Lin, R., Dong, H., Wang, J., and Yang, Y. (2022). Scalable Model-based Policy Optimization for Decentralized Networked Systems. arXiv:2207.06559 [cs, math, stat].

[Dubey and Pentland, 2021] Dubey, A. and Pentland, A. (2021). Provably Efficient Cooperative Multi-Agent Reinforcement Learning with Function Approximation. arXiv:2103.04972 [cs, stat].

[Filatrella et al., 2008] Filatrella, G., Nielsen, A. H., and Pedersen, N. F. (2008). Analysis of a power grid using a kuramoto-like model. *The European Physical Journal B*, 61:485–491.

[Gu et al., 2022] Gu, H., Guo, X., Wei, X., and Xu, R. (2022). Mean-Field Multi-Agent Reinforcement Learning: A Decentralized Network Approach. arXiv:2108.02731 [cs, math].

[Guestrin et al., 2003] Guestrin, C., Koller, D., Parr, R., and Venkataraman, S. (2003). Efficient solution algorithms for factored mdps. *Journal of Artificial Intelligence Research*, 19:399–468.

[Gutmann and Hyvärinen, 2010] Gutmann, M. and Hyvärinen, A. (2010). Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 297–304. JMLR Workshop and Conference Proceedings.

[Haarnoja et al., 2018] Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR.

[Hu and Wellman, 2003] Hu, J. and Wellman, M. P. (2003). Nash Q-learning for general-sum stochastic games. *Journal of machine learning research*, 4(Nov):1039–1069.

[Jin et al., 2020a] Jin, C., Yang, Z., Wang, Z., and Jordan, M. I. (2020a). Provably efficient reinforcement learning with linear function approximation. In *Proceedings of Thirty Third Conference on Learning Theory*, pages 2137–2143. PMLR. ISSN: 2640-3498.

[Jin et al., 2020b] Jin, C., Yang, Z., Wang, Z., and Jordan, M. I. (2020b). Provably efficient reinforcement learning with linear function approximation. In *Conference on learning theory*, pages 2137–2143. PMLR.

[Kearns and Koller, 1999] Kearns, M. and Koller, D. (1999). Efficient reinforcement learning in factored mdps. In *IJCAI*, volume 16, pages 740–747.

[Leonardos et al., 2022] Leonardos, S., Overman, W., Panageas, I., and Piliouras, G. (2022). Global convergence of multi-agent policy gradient in Markov potential games. In *International Conference on Learning Representations.*

[Li and Pi, 2020] Li, B. and Pi, D. (2020). Network representation learning: a systematic literature review. *Neural Computing and Applications*, 32(21):16647–16679.

[Li et al., 2021] Li, Y., Tang, Y., Zhang, R., and Li, N. (2021). Distributed reinforcement learning for decentralized linear quadratic control: A derivative-free policy optimization approach. *IEEE Transactions on Automatic Control*, 67(12):6429–6444.

[Lin et al., 2021a] Lin, Y., Qu, G., Huang, L., and Wierman, A. (2021a). Multi-Agent Reinforcement Learning in Stochastic Networked Systems. arXiv:2006.06555 [cs, stat].

[Lin et al., 2021b] Lin, Y., Qu, G., Huang, L., and Wierman, A. (2021b). Multi-agent reinforcement learning in stochastic networked systems. *Advances in neural information processing systems*, 34:7825–7837.

[Littman, 1994] Littman, M. L. (1994). Markov games as a framework for multi-agent reinforcement learning. In *Machine learning proceedings 1994*, pages 157–163. Elsevier.

[Ma et al., 2024] Ma, C., Li, A., Du, Y., Dong, H., and Yang, Y. (2024). Efficient and scalable reinforcement learning for large-scale network control. *Nature Machine Intelligence.*

[Ma and Collins, 2018] Ma, Z. and Collins, M. (2018). Noise contrastive estimation and negative sampling for conditional models: Consistency and statistical efficiency. *arXiv preprint arXiv:1809.01812.*

[McArthur et al., 2007] McArthur, S. D., Davidson, E. M., Catterson, V. M., Dimeas, A. L., Hatziargyriou, N. D., Ponci, F., and Funabashi, T. (2007). Multi-agent systems for power engineering applications—part i: Concepts, approaches, and technical challenges. *IEEE Transactions on Power systems*, 22(4):1743–1752.

[Meuleau et al., 1998] Meuleau, N., Hauskrecht, M., Kim, K.-E., Peshkin, L., Kaelbling, L. P., Dean, T. L., and Boutilier, C. (1998). Solving very large weakly coupled markov decision processes. *AAAI/IAAI*, 8:2.

[Mozafari et al., 2012] Mozafari, Y., Kiani, A., and Hirche, S. (2012). Oscillator network synchronization by distributed control. In *2012 IEEE International Conference on Control Applications*, pages 621–626. IEEE.

[Nezamoddini and Gholami, 2022] Nezamoddini, N. and Gholami, A. (2022). A Survey of Adaptive Multi-Agent Networks and Their Applications in Smart Cities. *Smart Cities*, 5(1):318–347. Number: 1 Publisher: Multidisciplinary Digital Publishing Institute.

[Olsson et al., 2024] Olsson, J., Zhang, R. C., Tegling, E., and Li, N. (2024). Scalable reinforcement learning for linear-quadratic control of networks. In *2024 American Control Conference (ACC)*, pages 1813–1818.

[Osband and Van Roy, 2014] Osband, I. and Van Roy, B. (2014). Near-optimal reinforcement learning in factored mdps. *Advances in Neural Information Processing Systems*, 27.

[Qu et al., 2020a] Qu, G., Lin, Y., Wierman, A., and Li, N. (2020a). Scalable multi-agent reinforcement learning for networked systems with average reward. *Advances in Neural Information Processing Systems*, 33:2074–2086.

[Qu et al., 2020b] Qu, G., Wierman, A., and Li, N. (2020b). Scalable reinforcement learning of localized policies for multi-agent networked systems. In Bayen, A. M., Jadbabaie, A., Pappas, G., Parrilo, P. A., Recht, B., Tomlin, C., and Zeilinger, M., editors, *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, volume 120 of *Proceedings of Machine Learning Research*, pages 256–266. PMLR.

[Qu et al., 2021] Qu, G., Wierman, A., and Li, N. (2021). Scalable Reinforcement Learning for Multi-Agent Networked Systems. arXiv:1912.02906 [cs, math].

[Rahimi and Recht, 2007] Rahimi, A. and Recht, B. (2007). Random features for large-scale kernel machines. *Advances in neural information processing systems*, 20.

[Rantzer, 2011] Rantzer, A. (2011). Distributed control of positive systems. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 6608–6611. IEEE.

[Ren et al., 2023] Ren, T., Ren, Z., Li, N., and Dai, B. (2023). Stochastic Nonlinear Control via Finite-dimensional Spectral Dynamic Embedding. In *2023 62nd IEEE Conference on Decision and Control (CDC)*, pages 795–800. ISSN: 2576-2370.

[Ren et al., 2022a] Ren, T., Xiao, C., Zhang, T., Li, N., Wang, Z., Sanghavi, S., Schuurmans, D., and Dai, B. (2022a). Latent variable representation for reinforcement learning. *arXiv preprint arXiv:2212.08765*.

[Ren et al., 2022b] Ren, T., Zhang, T., Lee, L., Gonzalez, J. E., Schuurmans, D., and Dai, B. (2022b). Spectral decomposition representation for reinforcement learning. *arXiv preprint arXiv:2208.09515*.

[Ren et al., 2022c] Ren, T., Zhang, T., Szepesvári, C., and Dai, B. (2022c). A Free Lunch from the Noise: Provable and Practical Exploration for Representation Learning.

[Roscia et al., 2013] Roscia, M., Longo, M., and Lazaroiu, G. C. (2013). Smart city by multi-agent systems. In *2013 International Conference on Renewable Energy Research and Applications (ICRERA)*, pages 371–376. IEEE.

[Rotkowitz and Lall, 2005] Rotkowitz, M. and Lall, S. (2005). A characterization of convex problems in decentralized control. *IEEE transactions on Automatic Control*, 50(12):1984–1996.

[Saeks, 1979] Saeks, R. (1979). On the decentralized control of interconnected dynamical systems. *IEEE Transactions on Automatic Control*, 24(2):269–271.

[Shapley, 1953] Shapley, L. S. (1953). Stochastic games. *Proceedings of the national academy of sciences*, 39(10):1095–1100.

[Shin et al., 2023] Shin, S., Lin, Y., Qu, G., Wierman, A., and Anitescu, M. (2023). Near-Optimal Distributed Linear-Quadratic Regulator for Networked Systems. *SIAM Journal on Control and Optimization*, 61(3):1113–1135. Publisher: Society for Industrial and Applied Mathematics.

[Shribak et al., 2024] Shribak, D., Gao, C.-X., Li, Y., Xiao, C., and Dai, B. (2024). Diffusion spectral representation for reinforcement learning. *arXiv preprint arXiv:2406.16121*.

[Song et al., 2021] Song, Z., Mei, S., and Bai, Y. (2021). When can we learn general-sum Markov games with a large number of players sample-efficiently?

[Stankovic and Stankovic, 2016] Stankovic, M. S. and Stankovic, S. S. (2016). Multi-agent temporal-difference learning with linear function approximation: Weak convergence under time-varying network topologies. In *2016 American Control Conference (ACC)*, pages 167–172. ISSN: 2378-5861.

[Sutton, 2018] Sutton, R. S. (2018). Reinforcement learning: An introduction. *A Bradford Book*.

[Sutton et al., 1999] Sutton, R. S., McAllester, D., Singh, S., and Mansour, Y. (1999). Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12.

[Tropp et al., 2015] Tropp, J. A. et al. (2015). An introduction to matrix concentration inequalities. *Foundations and Trends® in Machine Learning*, 8(1-2):1–230.

[Williams and Seeger, 2000] Williams, C. and Seeger, M. (2000). Using the nyström method to speed up kernel machines. *Advances in neural information processing systems*, 13.

[Yan and Xu, 2020] Yan, Z. and Xu, Y. (2020). A Multi-Agent Deep Reinforcement Learning Method for Cooperative Load Frequency Control of a Multi-Area Power System. *IEEE Transactions on Power Systems*, 35(6):4599–4608. Conference Name: IEEE Transactions on Power Systems.

[Zhang et al., 2020] Zhang, D., Yin, J., Zhu, X., and Zhang, C. (2020). Network Representation Learning: A Survey. *IEEE Transactions on Big Data*, 6(1):3–28. Conference Name: IEEE Transactions on Big Data.

[Zhang et al., 2018] Zhang, K., Yang, Z., Liu, H., Zhang, T., and Basar, T. (2018). Fully Decentralized Multi-Agent Reinforcement Learning with Networked Agents. In *Proceedings of the 35th International Conference on Machine Learning*, pages 5872–5881. PMLR. ISSN: 2640-3498.

[Zhang et al., 2024] Zhang, R., Ren, Z., and Li, N. (2024). Gradient play in stochastic games: Stationary points, convergence, and sample complexity. *IEEE Transactions on Automatic Control*, 69(10):6499–6514.

[Zhang et al., 2023a] Zhang, R., Zhang, Y., Konda, R., Ferguson, B., Marden, J., and Li, N. (2023a). Markov games with decoupled dynamics: Price of anarchy and sample complexity. In *2023 62nd IEEE Conference on Decision and Control (CDC)*, pages 8100–8107. IEEE.

[Zhang et al., 2023b] Zhang, R. C., Li, W., and Li, N. (2023b). On the Optimal Control of Network LQR with Spatially-Exponential Decaying Structure. In *2023 American Control Conference (ACC)*, pages 1775–1780. ISSN: 2378-5861.

[Zhang et al., 2022] Zhang, T., Ren, T., Yang, M., Gonzalez, J., Schuurmans, D., and Dai, B. (2022). Making linear mdps practical via contrastive representation learning. In *International Conference on Machine Learning*, pages 26447–26466. PMLR.

[Zhang et al., 2016] Zhang, X., Shi, W., Li, X., Yan, B., Malkawi, A., and Li, N. (2016). Decentralized temperature control via hvac systems in energy efficient buildings: An approximate solution procedure. In *2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 936–940. IEEE.

[Zhang et al., 2023c] Zhang, Y., Qu, G., Xu, P., Lin, Y., Chen, Z., and Wierman, A. (2023c). Global convergence of localized policy iteration in networked multi-agent reinforcement learning. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 7(1):1–51.

[Zocca, 2019] Zocca, A. (2019). Temporal starvation in multi-channel csma networks: an analytical framework. *ACM SIGMETRICS Performance Evaluation Review*, 46(3):52–53.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. **Yes**
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. **Yes**
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. **Yes**

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. **Yes**
   (b) Complete proofs of all theoretical results. **Yes**
   (c) Clear explanations of any assumptions. **Yes**

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). **Yes**
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). **Yes**
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). **Yes**
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). **Yes**

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. **Yes**
   (b) The license information of the assets, if applicable. **Not applicable**
   (c) New assets either in the supplemental material or as a URL, if applicable. **Yes**
   (d) Information about consent from data providers/curators. **Not applicable**
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. **Not applicable**

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. **Not Applicable**
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. **Not applicable**
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. **Not applicable**

7 Appendix

7.1 More Related Works

Multi-agent reinforcement learning (MARL) for Markov games The study of MARL dates back to the early work of [Littman, 1994, Claus and Boutilier, 1998, Hu and Wellman, 2003]. One classical setting considered is the stochastic game setting [Shapley, 1953], where agents can take their own actions but they share a common global state and maximize a global reward. There are many recent works that studies reducing the computational and sample complexity in this scenario, e.g. [Zhang et al., 2018, Leonardos et al., 2022, Song et al., 2021, Zhang et al., 2024]. However, one restriction of their approach is that it assumes access to the global state s, as opposed to our setting where each agent only observes a local state $s_{\mathcal{N}_i^\kappa}$, thus their approach won't extend easily to the network setting considered in our paper.

Distributed Control for Network Linear Quadratic Regulator (LQR) Apart from reinforcement learning, there is also another line of work tackling the network system from the control perspective. Distributed control is a classical topic that is widely discussed in literature (c.f. [Saeks, 1979, Bamieh et al., 2002, Rotkowitz and Lall, 2005, Rantzer, 2011]) Notably, some recent works [Zhang et al., 2023b, Shin et al., 2023, Olsson et al., 2024] also leverage a similar exponential decaying property to address the near-global-optimality of distributed controllers. However, these works are primarily focused on linear dynamical systems, as opposed to our work which is targeted to potentially nonlinear dynamics via representation learning

Other multi-agent dynamical models There are also other settings in multi-agent/network systems such as the weakly coupled MDP [Meuleau et al., 1998, Zhang et al., 2023a], where agents' transition dynamics are fully decoupled and the only coupling is through the reward function; factored MDP [Kearns and Koller, 1999, Guestrin et al., 2003, Osband and Van Roy, 2014], where there is a global action affecting every agent's individual local state. (See the 'Related Literature' in [Qu et al., 2021] for a more detailed summary and comparison.)

Linear MDPs There has been a sequence of work on sample-efficient RL via the linear MDPs approach [Jin et al., 2020a]. In a linear MDP, the transition kernel of the MDP can be represented as a linear combination of low-rank features. By applying the Bellman equation, the Q-value function can then be represented as a linear combination of these low-rank features. It has been shown in [Jin et al., 2020a] in this setting, sample-efficient RL can be achieved, with sample complexity depending on the dimension of the feature space rather than the size of the state and action spaces. Computationally, to realize the theoretical promise of linear MDPs, there has been a line of works with strong empirical performances that represents the Q-value function using finite-dimensional features that factorize the transition kernel, where the features are learnt in different ways, e.g. a latent variable approach [Ren et al., 2022a], a noise-contrastive approach [Zhang et al., 2022] building on noise contrastive estimation [Gutmann and Hyvärinen, 2010, Ma and Collins, 2018], and a diffusion-inspired approach [Shribak et al., 2024]. A key question in the study of linear MDPs is the circumstances under which a low-rank linear factorization of possibly nonlinear features that represent the transition kernel indeed exists. Towards answering this question, the works in [Ren et al., 2022c, Ren et al., 2023] show that for a wide range of problems in stochastic nonlinear control, specifically stochastic nonlinear control problems where the transition noise takes the form of a positive-definite kernel (e.g. Gaussian noise), there exists infinite-dimensional spectral features which exactly factorizes the transition kernel. However, to enable tractable control, finite-dimensional features are required. Tn [Ren et al., 2023], the authors propose a finite-dimensional approximation of the infinite-dimensional spectral features via random features [Rahimi and Recht, 2007] and Nystrom features [Williams and Seeger, 2000], characterize the approximation error of these finite-dimensional truncation approaches, and provide end-to-end theoretical guarantees for a actor-critic framework building on these finite-dimensional features, with strong empirical performance.

7.2 Example of thermal control in buildings as network MDP

Example 2 (Thermal control in buildings)**.** *The problem of thermal control of multiple zones in a building can also be cast as a network MDP. Consider a multi-zone building with a Heating Ventilation and Air Conditioning (HVAC) system. Each zone is equipped with a sensor that can measure the local temperatures and can adjust the supply air flow rate of its associated HVAC system. For simplicity, we consider a discrete-time linear thermal* *dynamics model based on [Zhang et al., 2016, Li et al., 2021], where for any $i \in [n]$,*

$$x_i(t+1) - x_i(t) = \frac{\Delta}{v_i \zeta_i}(\theta^o(t) - x_i(t)) + \sum_{j \in N_i} \frac{\Delta}{v_i \zeta_{ij}}(x_j(t) - x_i(t)) + \frac{\Delta}{v_i} a_i(t) + \frac{\Delta}{v_i}\pi_i + \sqrt{\frac{\Delta}{v_i}} w_i(t),$$

where $x_i(t)$ denotes the temperature of zone i at time t, $a_i(t)$ denotes the control input of zone i that is related with the air flow rate of the HVAC system, $\theta^o(t)$ denotes the outdoor temperature, π_i represents a constant heat from external sources to zone i, $w_i(t)$ represents random disturbances, Δ is the time resolution, v_i is the thermal capacitance of zone i, ζ_i represents the thermal resistance of the windows and walls between zone i and the outside environment, and ζ_{ij} represents the thermal resistance of the walls between zone i and j. Again, we note that the transition dynamics of each agent depends only on its neighbors (and itself). At each zone i, there is a desired temperature θ_i^* set by the users. The local reward function is composed of the (negative) deviation from the desired temperature and the control cost, i.e.

$$r_i(t) = -\left((x_i(t) - \theta_i^*)^2 + \alpha_i a_i(t)^2\right),$$

where $\alpha_i > 0$ is a trade-off parameter.

7.3 On the exponential decay property

It may not be immediately clear when the exponential decay property holds. The following lemma (cf. Appendix A in [Qu et al., 2020b]) highlights that for a local policy where each agent's actions depend only only on its and its neighbors' states (i.e. $\pi_i(\cdot \mid s) \equiv \pi_i(\cdot \mid s_{N_i})$), the exponential decay property holds generally, with $\rho = \gamma$. We defer the proof to our appendix.

Lemma 7. *Suppose $\forall i \in [n]$, agent i adopts a localized policy, i.e. $\pi_i(\cdot \mid s) \equiv \pi_i(\cdot \mid s_{N_i})$. Suppose also that the local rewards are bounded such that $0 \leqslant r_i \leqslant \bar{r}$. Then the $\left(\frac{\bar{r}}{1-\gamma}, \gamma\right)$-exponential decay property holds.*

We note that under some mixing time assumptions on the MDP [Qu et al., 2020b], the exponential decay property may in fact hold for $\rho < \gamma$ depending on the system parameters, making it applicable to problems with large discount factors or even in the average-reward setting [Qu et al., 2020a].

We proceed now to prove Lemma 7, which shows that the exponential decay property holds for localized policies and bounded rewards. We note that this was first shown in [Qu et al., 2020b], and we provide the proof here for completeness.

Proof. Consider any i, and choose any natural number κ. For an arbitrary $(s,a) = (s_{N_i^\kappa}, s_{N_{-i}^\kappa}, a_{N_i^\kappa}, a_{N_{-i}^\kappa})$, consider any state-action pair (s', a') that differs with (s,a) only outside the N_i^κ-neighborhood, i.e. $(s',a') = (s_{N_i^\kappa}, s'_{N_{-i}^\kappa}, a_{N_i^\kappa}, a'_{N_{-i}^\kappa})$. For any natural number t, let $p_{t,i}$ denote the distribution of $s_i(t), a_i(t)$ conditional on $s(0) = s, a(0) = a$, and let $p'_{t,i}$ denote the distribution of $s_i(t), a_i(t)$ conditional on $s(0) = s', a(0) = a'$. Then,

$$\begin{aligned}
Q_i^\pi(s,a) - Q_i^\pi(s',a') &= \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_i(s_i(t), a_i(t)) \mid s(0) = s, a(0) = a\right] - \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_i(s_i(t), a_i(t)) \mid s(0) = s', a(0) = a'\right] \\
&\overset{\text{(i)}}{=} \sum_{t=0}^{\infty} \gamma^t \left(\mathbb{E}_{p_{t,i}}\left[r_i(s_i(t), a_i(t))\right] - \mathbb{E}_{p'_{t,i}}\left[r_i(s_i(t), a_i(t))\right]\right) \\
&= \sum_{t=0}^{\kappa} \gamma^t \left(\mathbb{E}_{p_{t,i}}\left[r_i(s_i(t), a_i(t))\right] - \mathbb{E}_{p'_{t,i}}\left[r_i(s_i(t), a_i(t))\right]\right) \\
&\quad + \sum_{t=\kappa+1}^{\infty} \gamma^t \left(\mathbb{E}_{p_{t,i}}\left[r_i(s_i(t), a_i(t))\right] - \mathbb{E}_{p'_{t,i}}\left[r_i(s_i(t), a_i(t))\right]\right) \\
&\overset{\text{(ii)}}{=} \sum_{t=\kappa+1}^{\infty} \gamma^t \left(\mathbb{E}_{p_{t,i}}\left[r_i(s_i(t), a_i(t))\right] - \mathbb{E}_{p'_{t,i}}\left[r_i(s_i(t), a_i(t))\right]\right) \\
&\overset{\text{(iii)}}{\leqslant} \frac{\gamma^{\kappa+1}}{1-\gamma}\bar{r}.
\end{aligned}$$

Above, (i) is a direct application of the definition of $p_{t,i}$ and $p'_{t,i}$. Meanwhile, (ii) utilizes the fact that for any $0 \leqslant t \leqslant \kappa$, we have $p_{t,i} \equiv p'_{t,i}$. This is because of (a) localized policy, such that $a_i(t)$ depends only on $s_{N_i}(t-1)$, and (b) factorized localized dynamics, that $s_{N_i^j}(t)$ depends only only on $s_{N_i^{j+1}}(t)$ for any natural number j; hence an iterative argument shows that for any t, $p_{t,i}$ and $p'_{t,i}$ both only depend on $s_{N_i^t}(0)$ and $a_{N_i^t}(0)$. Thus, since (s,a) and (s',a') share identical $s_{N_i^\kappa}(0)$ and $a_{N_i^\kappa}(0)$, it follows that $p_{t,i} \equiv p'_{t,i}$ for $t \leqslant \kappa$. Finally, (iii) uses the fact that bounded reward assumption, i.e. $0 \leqslant r_i \leqslant \bar{r}$. The proof then concludes by rerunning the argument on $Q_i^\pi(s',a') - Q_i^\pi(s,a)$. □

Next, we state and prove the following elementary technical result, which bounds any two truncated Q(or V)-functions with different weights.

Lemma 8. *Suppose the (c,ρ)-exponential decay property holds. Then, for any two different weights $w_i(s_{N_i^\kappa}, a_{N_i^\kappa}; s_{N_{-i}^\kappa}, a_{N_{-i}^\kappa})$ and $w'_i(s_{N_i^\kappa}, a_{N_i^\kappa}; s_{N_{-i}^\kappa}, a_{N_{-i}^\kappa})$ over the space $\mathcal{S}_{N_{-i}^\kappa} \times \mathcal{A}_{N_{-i}^\kappa}$, i.e.*

$$\sum_{s_{N_{-i}^\kappa}, a_{N_{-i}^\kappa}} w_i(s_{N_i^\kappa}, a_{N_i^\kappa}; s_{N_{-i}^\kappa}, a_{N_{-i}^\kappa}) = 1,$$

$$\sum_{s_{N_{-i}^\kappa}, a_{N_{-i}^\kappa}} w'_i(s_{N_i^\kappa}, a_{N_i^\kappa}; s_{N_{-i}^\kappa}, a_{N_{-i}^\kappa}) = 1,$$

we have

$$\left|\hat{Q}_i^\pi(s_{N_i^\kappa}, a_{N_i^\kappa}) - (\hat{Q}')_i^\pi(s_{N_i^\kappa}, a_{N_i^\kappa})\right| \leqslant 2c\rho^{\kappa+1},$$

where

$$\hat{Q}_i^\pi(s_{N_i^\kappa}, a_{N_i^\kappa}) = \sum_{s_{N_{-i}^\kappa}, a_{N_{-i}^\kappa}} w_i(s_{N_i^\kappa}, a_{N_i^\kappa}; s_{N_{-i}^\kappa}, a_{N_{-i}^\kappa}) Q_i^\pi(s_{N_i^\kappa}, s_{N_{-i}^\kappa}, a_{N_i^\kappa}, a_{N_{-i}^\kappa}),$$

$$(\hat{Q}')_i^\pi(s_{N_i^\kappa}, a_{N_i^\kappa}) = \sum_{s_{N_{-i}^\kappa}, a_{N_{-i}^\kappa}} w'_i(s_{N_i^\kappa}, a_{N_i^\kappa}; s_{N_{-i}^\kappa}, a_{N_{-i}^\kappa}) Q_i^\pi(s_{N_i^\kappa}, s_{N_{-i}^\kappa}, a_{N_i^\kappa}, a_{N_{-i}^\kappa})$$

Similarly, for any two different weights $w_i(s_{N_i^\kappa}; s_{N_{-i}^\kappa})$ and $w'_i(s_{N_i^\kappa}; s_{N_{-i}^\kappa})$ over the space $\mathcal{S}_{N_{-i}^\kappa}$, i.e.

$$\sum_{s_{N_{-i}^\kappa}} w_i(s_{N_i^\kappa}; s_{N_{-i}^\kappa}) = 1,$$

$$\sum_{s_{N_{-i}^\kappa}} w'_i(s_{N_i^\kappa}; s_{N_{-i}^\kappa}) = 1,$$

we have

$$\left|\hat{V}_i^\pi(s_{N_i^\kappa} - (\hat{V}')_i^\pi(s_{N_i^\kappa})\right| \leqslant 2c\rho^{\kappa+1},$$

where

$$\hat{V}_i^\pi(s_{N_i^\kappa}) = \sum_{s_{N_{-i}^\kappa}} w_i(s_{N_i^\kappa}; s_{N_{-i}^\kappa}) V_i^\pi(s_{N_i^\kappa}, s_{N_{-i}^\kappa}),$$

$$(\hat{V}')_i^\pi(s_{N_i^\kappa}) = \sum_{s_{N_{-i}^\kappa}} w'_i(s_{N_i^\kappa}; s_{N_{-i}^\kappa}) V_i^\pi(s_{N_i^\kappa}, s_{N_{-i}^\kappa})$$

Proof. Compare both truncated Q-functions to a Q-function evaluated at any specific state action pair where the states and actions of the agents in N_i^κ are $s_{N_i^\kappa}$ and $a_{N_i^\kappa}$ respectively. The desired result then follows by Definition 1. A similar argument works for the V-function. □

7.4 Helper results on factorization of network probability transition

Lemma 2. *Property 2 holds whenever Property 1 holds, by setting*

$$\phi_{i,\kappa}(s_{N_i^{\kappa+1}}(t), a_{N_i^{\kappa+1}}(t)) := \bigotimes_{j \in N_i^\kappa} \bar{\phi}_j(s_{N_j}(t), a_{N_j}(t)),$$
$$\mu_{i,\kappa}(s_{N_i^\kappa}(t+1)) := \bigotimes_{j \in N_i^\kappa} \bar{\mu}_j(s_j(t+1)). \qquad \square$$

Proof. To see that, observe that

$$\mathbb{P}\left(s_{N_i^\kappa}(t+1) \mid s_{N_i^{\kappa+1}}(t), a_{N_i^{\kappa+1}}(t)\right) = \prod_{j \in N_i^\kappa} \mathbb{P}\left(s_j(t+1) \mid s_{N_j}(t), a_{N_j}(t)\right)$$
$$\overset{\text{(iv)}}{=} \prod_{j \in N_i^\kappa} \langle \bar{\phi}_j(s_{N_j}(t), a_{N_j}(t)), \bar{\mu}_j(s_j(t+1)) \rangle \overset{\text{(v)}}{=} \left\langle \bigotimes_{j \in N_i^\kappa} \bar{\phi}_j(s_{N_j}(t), a_{N_j}(t)), \bigotimes_{j \in N_i^\kappa} \bar{\mu}_j(s_j(t+1)) \right\rangle.$$

Above, (iv) follows from Property 1, while (v) uses the definition of the inner product of two tensor products. Thus, when Property 1 holds, Property 2 holds by setting

$$\phi_{i,\kappa}(s_{N_i^{\kappa+1}}(t), a_{N_i^{\kappa+1}}(t)) := \bigotimes_{j \in N_i^\kappa} \bar{\phi}_j(s_{N_j}(t), a_{N_j}(t)), \quad \mu_{i,\kappa}(s_{N_i^\kappa}(t+1)) := \bigotimes_{j \in N_i^\kappa} \bar{\mu}_j(s_j(t+1)).$$

□

7.5 Approximation error of spectral features

7.5.1 Approximation error when spectral features exactly factorize κ-hop transition

We recall and prove this result, which bounds the approximation error of using the truncated spectral features to approximate the local Q_i-function, in the case when there is no approximation error in the spectral features in representing the κ-hop transition.

Lemma 3 (Local Q_i approximation via network κ-local spectral features)**.** *Suppose the (c, ρ)-exponential decay property holds. Suppose Property 2 also holds. Then, for any (s, a) pair, agent i, and natural number κ, there exists an approximation $\bar{Q}_i^\pi$ which depends linearly on network κ-local spectral features $\tilde{\phi}_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})$, such that*

$$\left| Q_i^\pi(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}}, s_{N_{-i}^{\kappa+1}}, a_{N_i^{\kappa+1}}) - \bar{Q}_i^\pi(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}}) \right|$$
$$\leqslant 2c\gamma\rho^{\kappa+1},$$

where $\bar{Q}_i^\pi(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}}) = \left\langle \tilde{\phi}_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}}), w_{i,\kappa}^\pi(s'_{N_i^\kappa}) \right\rangle,$
with the definitions

$$\tilde{\phi}_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}}) := [r_i(s_i, a_i), \phi_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})]^\top,$$
$$w_{i,\kappa}^\pi(s'_{N_i^\kappa}) := [1, \gamma \int_{s'_{N_i^\kappa}} ds'_{N_i^\kappa} \mu_{i,\kappa}(s'_{N_i^\kappa}) \bar{V}_i^\pi(s'_{N_i^\kappa})]^\top,$$

where

$$\bar{V}_i^\pi(s'_{N_i^\kappa}) := \int_{s'_{N_{-i}^\kappa}} \frac{ds'_{N_{-i}^\kappa}}{\mathrm{Vol}(\mathcal{S}_{N_{-i}^\kappa})} V_i^\pi(s'_{N_i^\kappa}, s'_{N_{-i}^\kappa}). \qquad \square$$

Proof. For notational convenience, we omit the t and $(t+1)$ in the parentheses of the state and action notations, and instead use a superscript $^+$ to denote $(t+1)$, e.g. s^+ to denote $s(t+1)$. Observe that

$$Q_i^\pi(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}}, s_{N_{-i}^{\kappa+1}}, a_{N_{-i}^{\kappa+1}})$$
$$= r_i(s_i, a_i) + \gamma \int_{s^+_{N_i^\kappa}} ds^+_{N_i^\kappa} \mathbb{P}\left(s^+_{N_i^\kappa} \mid s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}}\right) \left(\int_{s^+_{N_{-i}^\kappa}} ds^+_{N_{-i}^\kappa} V_i^\pi(s^+_{N_i^\kappa}, s^+_{N_{-i}^\kappa}) \mathbb{P}\left(s^+_{N_{-i}^\kappa} \mid s, a\right) \right)$$
$$\overset{\text{(vi)}}{=} r_i(s_i, a_i) + \gamma \int_{s^+_{N_i^\kappa}} ds^+_{N_i^\kappa} \mathbb{P}\left(s^+_{N_i^\kappa} \mid s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}}\right) \hat{V}_i^\pi(s^+_{N_i^\kappa})$$

$$
\begin{aligned}
&\overset{\text{(vii)}}{=} r_i(s_i, a_i) + \gamma \int_{s^+_{N_i^\kappa}} ds^+_{N_i^\kappa} \mathbb{P}\left(s^+_{N_i^\kappa} \mid s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}}\right) \bar{V}_i^\pi(s^+_{N_i^\kappa}) \\
&\qquad + \gamma \int_{s^+_{N_i^\kappa}} ds^+_{N_i^\kappa} \mathbb{P}\left(s^+_{N_i^\kappa} \mid s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}}\right) (\hat{V}_i^\pi(s^+_{N_i^\kappa}) - \bar{V}_i^\pi(s^+_{N_i^\kappa})) \\
&\overset{\text{(viii)}}{=} r_i(s_i, a_i) + \gamma \int_{s^+_{N_i^\kappa}} ds^+_{N_i^\kappa} \left\langle \phi_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}}), \mu_{i,\kappa}(s^+_{N_i^\kappa}) \bar{V}_i^\pi(s^+_{N_i^\kappa}) \right\rangle \\
&\qquad + \gamma \int_{s^+_{N_i^\kappa}} ds^+_{N_i^\kappa} \mathbb{P}\left(s^+_{N_i^\kappa} \mid s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}}\right) (\hat{V}_i^\pi(s^+_{N_i^\kappa}) - \bar{V}_i^\pi(s^+_{N_i^\kappa}))
\end{aligned}
$$

In (vi) above, we used the notation

$$
\hat{V}_i^\pi(s^+_{N_i^\kappa}) := \int_{s^+_{N_{-i}^\kappa}} ds^+_{N_{-i}^\kappa} V_i^\pi(s^+_{N_i^\kappa}, s^+_{N_{-i}^\kappa}) \mathbb{P}\left(s^+_{N_{-i}^\kappa} \mid s, a\right),
$$

and in (vii), we recall that we defined

$$
\bar{V}_i^\pi(s^+_{N_i^\kappa}) := \int_{s^+_{N_{-i}^\kappa}} \frac{ds^+_{N_{-i}^\kappa}}{\mathrm{Vol}(\mathcal{S}_{N_{-i}^\kappa})} V_i^\pi(s^+_{N_i^\kappa}, s^+_{N_{-i}^\kappa}).
$$

Since the (c, ρ)-exponential decay property holds, applying Lemma 8, we have

$$
\left|\hat{V}_i^\pi(s^+_{N_i^\kappa}) - \bar{V}_i^\pi(s^+_{N_i^\kappa})\right| \leqslant 2c\rho^{\kappa+1}.
$$

Thus our desired result holds by setting

$$
\bar{Q}_i^\pi(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}}) := r_i(s_i, a_i) + \left\langle \phi_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}}), \gamma \int_{s^+_{N_i^\kappa}} ds^+_{N_i^\kappa} \mu_{i,\kappa}(s^+_{N_i^\kappa}) \bar{V}_i^\pi(s^+_{N_i^\kappa}) \right\rangle
$$

□

7.5.2 Results on approximation error of random features

We first state the following result on uniform convergence of random Fourier features, adapted from Claim 1 in [Rahimi and Recht, 2007].

Lemma 9 (Uniform convergence of Fourier features). *Let $\mathcal{M}$ be a compact subset of $\mathbb{R}^d$ with diameter diam$(\mathcal{M})$. Let k be a positive definite shift-invariant kernel $k(x, y) = k(x - y)$. Define the mapping z, where*

$$
z = \sqrt{\frac{2}{D}} \begin{bmatrix} \cos(\omega_1^\top x + b_1) & \cdots & \cos(\omega_D^\top x + b_D) \end{bmatrix},
$$

where $\omega_1, \ldots, \omega_D \in \mathbb{R}^d$ are D iid samples from p, where p is the Fourier transform of k, i.e. $p(\omega) = \frac{1}{2\pi} \int e^{-j\omega^\top \delta} k(\delta) d\delta$, and $b_1, \ldots, b_D$ are D are iid samples from $\mathrm{Unif}(0, 2\pi)$. We assume that k is suitably scaled such that p is a probability distribution.

Then, for the mapping z defined above, we have

$$
\Pr\left[\sup_{x,y \in \mathcal{M}} |z(x)^\top z(y) - k(x, y)| \geqslant \epsilon\right] \leqslant 2^8 \left(\frac{\sigma_p \, diam(\mathcal{M})}{\epsilon}\right)^2 \exp\left(-\frac{D\epsilon^2}{4(d+2)}\right),
$$

where $\sigma_p^2 = \mathbb{E}_p[\omega^\top \omega]$ is the second moment of the Fourier transform of k.

Further,

$$
\sup_{x,y \in \mathcal{M}} |z(x)^\top z(y) - k(x, y)| \leqslant \epsilon
$$

with probability at least $1 - \delta$ when

$$
D = \Omega\left(\log\left(\frac{\sigma_p \, diam(\mathcal{M})^2}{\delta\epsilon}\right) \frac{d}{\epsilon^2}\right).
$$

Lemma 5. *Fix any $i \in [n]$. Suppose the local dynamics take the form $s_i' = f_i(s_{N_i}, a_{N_i}) + \epsilon_i$ where $\epsilon_i \sim N(0, \sigma^2 I_S)$, such that for any κ, $s'_{N_i^\kappa} = f_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}}) + \epsilon_{N_i^\kappa}$ where $\epsilon_{N_i^\kappa} \sim N(0, \sigma^2 I_{|N_i^\kappa|S})$ and $f_{i,\kappa}$ is concatenation of f_j for $j \in N_i^\kappa$. Fix any $0 \leqslant \alpha < 1$. Then, for a positive integer m, define the m-dimensional features $\hat\phi_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}}) \in \mathbb{R}^m$, where*

$$\begin{aligned}&\hat\phi_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})\\ :=\ &\frac{g_\alpha(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})}{\alpha^{|N_i^\kappa|S}}\\ &\times \left\{\sqrt{\frac{2}{m}}\cos\left(\frac{\omega_\ell^\top f_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})}{\sqrt{1-\alpha^2}} + b_\ell\right)\right\}_{\ell=1}^m,\end{aligned}$$

with $\{\omega_\ell\}_{\ell=1}^m$ being i.i.d draws from $N(0, \sigma^{-2} I_{|N_i^\kappa|S})$, $\{b_\ell\}_{\ell=1}^m$ being i.i.d draws from $\mathrm{Unif}([0, 2\pi])$, and

$$g_\alpha(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}}) := \exp\left(\frac{\alpha^2 \left\|f_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})\right\|^2}{2(1-\alpha^2)\sigma^2}\right).$$

In addition, define

$$\tilde g_\alpha := \max_{i \in [n]} \left(\sup_{x \in f_{i,\kappa}(\mathcal{S}_{N_i^{\kappa+1}}, \mathcal{A}_{N_i^{\kappa+1}})} \frac{g_\alpha(x)}{\alpha^{|N_i^\kappa|S}}\right).$$

Suppose

$$m = \Omega\left(\max_{i\in[n]}\left[\log\left(\frac{(|N_i^\kappa|S(diam(\mathcal{S}_{N_i^\kappa}))^2)}{\sigma^2(\delta/n)(\epsilon_P/\tilde g_\alpha)}\right)\frac{|N_i^\kappa|S\tilde g_\alpha^2}{\epsilon_P^2}\right]\right)$$

for some $\delta > 0$. Then, with probability at least $1-\delta$, the condition in (1) holds for every $i \in [n]$ and any distribution ν^o over $\mathcal{S}\times\mathcal{A}$, with

$$\hat\mu_{i,\kappa}(s'_{N_i^\kappa}) := \left\{\sqrt{\frac{2}{m}} p_\alpha(s'_{N_i^\kappa}) \cos(\sqrt{1-\alpha^2}\omega_\ell^\top s'_{N_i^\kappa} + b_\ell)\right\}_{\ell=1}^m,$$

where $p_\alpha(s'_{N_i^\kappa}) := \frac{\alpha^{|N_i^\kappa|S}}{(2\pi\sigma^2)^{|N_i^\kappa|S}} \exp(-\frac{\left\|\alpha s'_{N_i^\kappa}\right\|^2}{2\sigma^2})$ is a Gaussian distribution with standard deviation $\frac{\sigma}{\alpha}$. □

Proof. Observe that $P\left(s^+_{N_i^\kappa} \mid f_{i,\kappa}(s_{N_i^{\kappa+1}, a_{N_i^{\kappa+1}}})\right)$ follows the Gaussian distribution $N(0, \sigma^2 I_{|N_i^\kappa|S})$. For notational convenience, in this proof, we denote $x := f_{i,\kappa}(s_{N_i^{\kappa+1}, a_{N_i^{\kappa+1}}})$, $y := s^+_{N_i^\kappa}$. In addition, in this proof we denote $d := |N_i^\kappa|S$. For any $0 \leqslant \alpha < 1$, observe that

$$P(y \mid x) = \frac{g_\alpha(x)}{\alpha^d}\exp\left(-\frac{\left\|(1-\alpha^2)y - x\right\|^2}{2\sigma^2(1-\alpha^2)}\right)p_\alpha(y),$$

where $g_\alpha(x) := \exp(\alpha^2\|x\|^2/(2\sigma^2(1-\alpha^2)))$, and $p_\alpha(y) := \frac{\alpha^d}{(2\pi\sigma^2)^{d/2}}\exp(-\|\alpha y\|^2/(2\sigma^2))$.[2]

Define $\tilde g_\alpha := \max_{i\in[n]} \sup_{x \in f_{i,\kappa}(\mathcal{S}_{N_i^{\kappa+1}}, \mathcal{A}_{N_i^{\kappa+1}})} \frac{g_\alpha(x)}{\alpha^d}$.[3] Observe now that $k_\alpha(z, z') := \exp(-\frac{\left\|z-z'\right\|^2}{2\sigma^2(1-\alpha^2)})$ is a positive-definite shift-invariant kernel. Hence, by Lemma 9, if $m = \Omega\left(\log\left(\frac{d\sigma^{-2}diam(\mathcal{S}_{N_i^\kappa})^2}{\delta(\epsilon/\tilde g_\alpha)}\right)\frac{d\tilde g_\alpha^2}{\epsilon^2}\right)$ it follows that with probability at least $1-\delta$,

$$\sup_{x,y\in\mathcal{S}_{N_i^\kappa}} \left|\bar\phi_{i,\kappa}(x)^\top \bar\mu_{i,\kappa}(y) - k_\alpha(x, (1-\alpha^2)y)\right| \leqslant \frac{\epsilon_P}{\tilde g_\alpha},$$

[2] When $\alpha := 0$, this simplifies to $P(y \mid x) \propto \exp\left(-\frac{\|y-x\|^2}{2\sigma^2}\right)$. However, we allow a general $0 \leqslant \alpha < 1$ because it gives greater flexibility both theoretically and empirically.

[3] Here we overload notation to use $f_{i,\kappa}(\mathcal{S}_{N_i^{\kappa+1}}, \mathcal{A}_{N_i^{\kappa+1}})$ to denote $\{f_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})\}_{(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}}) \in \mathcal{S}_{N_i^{\kappa+1}} \times \mathcal{A}_{N_i^{\kappa+1}}}$

where

$$\bar{\phi}_{i,\kappa}(x) = \sqrt{\frac{2}{D}}\left\{\cos\left(\frac{w_\ell^\top x}{\sqrt{1-\alpha^2}} + b_\ell\right)\right\}_{\ell=1}^{m},$$

$$\bar{\mu}_{i,\kappa}(y) = \sqrt{\frac{2}{D}}\left\{\cos\left(\sqrt{1-\alpha^2}y + b_\ell\right)\right\}_{\ell=1}^{m},$$

and $\{\omega_\ell\}$'s are drawn iid from $N(0, \sigma^{-2}I_d)$, $\{b_\ell\}$'s are drawn iid from $\mathrm{Unif}(0, 2\pi)$. It follows then for any $x \in f_{i,\kappa}(\mathcal{S}_{N_i^{\kappa+1}}, \mathcal{A}_{N_i^{\kappa+1}})$, we have

$$\begin{aligned}\int_y \left|P(y \mid x) - \hat{\phi}_{i,\kappa}(x)^\top \hat{\mu}_{i,\kappa}(y)\right| dy &= \int_y \left|\frac{g_\alpha(x)}{\alpha^d}\right| \left|k_\alpha(x, (1-\alpha^2)y) - \bar{\phi}_{i,\kappa}(x)^\top \bar{\mu}_{i,\kappa}(y)\right| |p_\alpha(y)|\, dy \\ &\leqslant \tilde{g}_\alpha \frac{\epsilon_P}{\tilde{g}_\alpha} \int_y p_\alpha(y) dy = \epsilon_P\end{aligned}$$

where we recall that

$$\hat{\phi}_{i,\kappa}(x) := \frac{g_\alpha}{\alpha^d}\bar{\phi}_{i,\kappa}(x), \quad \hat{\mu}_{i,\kappa}(y) := p_\alpha(y)\bar{\mu}_{i,\kappa}(y).$$

The proof then follows by rescaling $\epsilon_P := \epsilon_P/n$, and taking a union bound over all $i \in [n]$.

□

7.6 Algorithm analysis - policy evaluation error

For simplicity, we assume throughout the analysis that we are solving the LSTD step in the policy evaluation exactly, i.e. we take the number of least square solves, T, to infinite. Moreover, we drop the i subscript in the notation of $\tilde{\phi}_{i,\kappa}$, and use ν to denote $\nu_{\pi^{(k)}}$. At round k, the algorithm output of the policy parameter w_i of agent i is given by

$$w_i^{(k)} = (M_i^{(k)})^{-1} H_i^{(k)} r_i$$

where

$$\begin{aligned}M_i^{(k)} &= \frac{1}{|D_k|}\sum_{s,a,s',a' \in D_k} \tilde{\phi}_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})\left(\tilde{\phi}_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}}) - \gamma\tilde{\phi}_{i,\kappa}(s'_{N_i^{\kappa+1}}, a'_{N_i^{\kappa+1}})\right)^\top, \\ H_i^{(k)} &= \frac{1}{|D_k|}\sum_{s,a,s',a' \in D_k} \tilde{\phi}_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})\tilde{\phi}_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})^\top.\end{aligned}$$

For notational convenience, when the context is clear, we drop the k-superscript indicating the current round k, and denote $w_i := w_i^{(k)}$, $M_i := M_i^{(k)}$ and $H_i := H_i^{(k)}$.

We define an intermediate variable $\tilde{w}_i$ as follows:

$$\tilde{w}_i = \widetilde{M}_i^{-1}\widetilde{H}_i r_i$$

where

$$\begin{aligned}\widetilde{M}_i &= \mathbb{E}_{s,a\sim\nu}\tilde{\phi}_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})\left(\tilde{\phi}_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}}) - \gamma\mathbb{E}_{s',a'\sim P(\cdot|s,a),\pi(\cdot|s')}\tilde{\phi}_{i,\kappa}(s'_{N_i^{\kappa+1}}, a'_{N_i^{\kappa+1}})\right)^\top \\ \widetilde{H}_i &= \mathbb{E}_{s,a\sim\nu}\tilde{\phi}_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})\tilde{\phi}_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})^\top\end{aligned}$$

and further define

$$\widetilde{Q}_i(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}}) = \tilde{\phi}_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})^\top \tilde{w}_i$$

The real Q-function is $Q_i^{\pi(k)}(s,a)$. From Lemma 3 and Lemma 4, we have that there exists

$$\hat{Q}_i(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}}) = \tilde{\phi}_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})^\top \hat{w}_i$$

such that with probability at least $1-\delta$, for every $i \in [n]$,

$$\mathbb{E}_\nu\left[|Q_i^{\pi(k)}(s,a) - \hat{Q}_i(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})|\right] \leqslant 2c\rho^{\kappa+1} + \left\|\frac{\nu}{\nu^o}\right\|_\infty \frac{\epsilon_P\gamma\bar{r}}{1-\gamma} \tag{5}$$

Assumption 1.

$$\|M_i^{-1}\| \leqslant D$$

Assumption 2.

$$\|\tilde{\phi}_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})\| \leqslant L, \quad \forall\ s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}}$$

Lemma 10 (Bellman Error)**.** *On the event that condition (1) in Lemma 4 holds, for every $i \in [n]$, we have*

$$\|\tilde{w}_i - \hat{w}_i\| \leqslant 2LD\left(c\rho^{\kappa+1} + \left\|\frac{\nu}{\nu^o}\right\|_\infty \frac{\gamma\bar{r}\epsilon_P}{1-\gamma}\right)$$

Proof. From Bellman equation we have that

$$\begin{aligned} &Q_i^{\pi(k)}(s,a) = r_i(s_i,a_i) + \gamma\mathbb{E}_{s',a'\sim P,\pi} Q_i^{\pi(k)}(s',a') \\ \Longrightarrow\quad &\hat{Q}_i(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}}) = r_i(s_i,a_i) + \gamma\mathbb{E}_{s',a'\sim P,\pi}\hat{Q}_i(s'_{N_i^{\kappa+1}}, a'_{N_i^{\kappa+1}}) + \Delta(s,a), \end{aligned}$$

where $\Delta(s,a) = -\left(Q_i^{\pi(k)}(s,a) - \hat{Q}_i(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})\right) + \gamma\mathbb{E}_{s',a'\sim P,\pi}\left(Q_i^{\pi(k)}(s',a') - \hat{Q}_i(s'_{N_i^{\kappa+1}}, a'_{N_i^{\kappa+1}})\right)$. Substituting $\hat{Q}_i(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}}) = \tilde{\phi}_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})^\top \hat{w}_i$ into the above equation we have

$$\tilde{\phi}_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})^\top \hat{w}_i = \tilde{\phi}_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})^\top r_i + \gamma\mathbb{E}_{s',a'\sim P,\pi}\tilde{\phi}_{i,\kappa}(s'_{N_i^{\kappa+1}}, a'_{N_i^{\kappa+1}})^\top \hat{w}_i + \Delta(s,a).$$

On both side multiply by $\tilde{\phi}_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})$ and take expectation over $s,a\sim\nu$, we have that

$$\begin{aligned} &\mathbb{E}_{s,a\sim\nu}\tilde{\phi}_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})\tilde{\phi}_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})^\top \hat{w}_i \\ &= \mathbb{E}_{s,a\sim\nu}\tilde{\phi}_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})\tilde{\phi}_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})^\top r_i \\ &\quad + \gamma\mathbb{E}_{s,a\sim\nu}\tilde{\phi}_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})\mathbb{E}_{s',a'\sim P,\pi}\tilde{\phi}_{i,\kappa}(s'_{N_i^{\kappa+1}}, a'_{N_i^{\kappa+1}})^\top \hat{w}_i + \mathbb{E}_{s,a\sim\nu}\tilde{\phi}_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})\Delta(s,a) \\ \Longrightarrow\quad &\widetilde{M}_i\hat{w}_i = \widetilde{H}_i r_i + \mathbb{E}_{s,a\sim\nu}\tilde{\phi}_{i,\kappa}(s_{N_i^{\kappa+1}}, a'_{N_i^{\kappa+1}})\Delta(s,a) \end{aligned}$$

Further, given that

$$\widetilde{M}_i\tilde{w}_i = \widetilde{H}_i r_i,$$

on the event that condition (1) in Lemma 4 holds.

$$\begin{aligned} \hat{w}_i - \tilde{w}_i &= \widetilde{M}_i^{-1}\mathbb{E}_{s,a\sim\nu}\tilde{\phi}_{i,\kappa}(s_{N_i^{\kappa+1}}, a'_{N_i^{\kappa+1}})\Delta(s,a) \\ \Longrightarrow\quad \|\hat{w}_i - \tilde{w}_i\| &\leqslant \|\widetilde{M}_i^{-1}\|\|\mathbb{E}_{s,a\sim\nu}\tilde{\phi}_{i,\kappa}(s_{N_i^{\kappa+1}}, a'_{N_i^{\kappa+1}})\Delta(s,a)\| \\ &\leqslant 2LD\left(c\rho^{\kappa+1} + \left\|\frac{\nu}{\nu^o}\right\|_\infty \frac{\epsilon_P\gamma\bar{r}}{1-\gamma}\right), \end{aligned}$$

where for the final inequality we used (5). □

Lemma 11 (Statistical Error)**.** *Fix an $i \in [n]$ and $k \in [K]$. For sample size $M_s \geqslant \log\left(\frac{2(m+1)}{\delta}\right)$, we have that with probability at least $1-2\delta$*

$$\|w_i^{(k)} - \tilde{w}_i^{(k)}\| \leqslant O\left(\log\left(\frac{(m+1)}{\delta}\right)\right)\frac{D^2L^4}{\sqrt{M_s}}$$

Proof. Again, for notational simplicity we drop the k-superscript. We first bound the differences $\|M_i - \widetilde{M}_i\|$, $\|H_i - \widetilde{H}_i\|$. Since

$$M_i = \frac{1}{|D_k|}\sum_{s,a,s',a'\in D_k}\tilde{\phi}_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})\left(\tilde{\phi}_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}}) - \gamma\tilde{\phi}_{i,\kappa}(s'_{N_i^{\kappa+1}}, a'_{N_i^{\kappa+1}})\right)^\top,$$

$$\widetilde{M}_i = \mathbb{E}_{s,a\sim\nu}\tilde{\phi}_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}}) \left(\tilde{\phi}_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}}) - \gamma\mathbb{E}_{s',a'\sim P(\cdot|s,a),\pi(\cdot|s')}\tilde{\phi}_{i,\kappa}(s'_{N_i^{\kappa+1}}, a'_{N_i^{\kappa+1}})\right)^\top,$$

From the Matrix Bernstein inequality (see Lemma 15 in Appendix 7.8) we have that when $M_s \geqslant \log\left(\frac{2(m+1)}{\delta}\right)$ with probability at least $1-\delta$

$$\|M_i - \widetilde{M}_i\| \leqslant 8L^2\sqrt{M_s^{-1}\log\left(\frac{2(m+1)}{\delta}\right)}$$

$$\|H_i - \widetilde{H}_i\| \leqslant 8L^2\sqrt{M_s^{-1}\log\left(\frac{2(m+1)}{\delta}\right)}$$

Thus with probability $1-2\delta$

$$\begin{aligned}\|w_i - \tilde{w}_i\| &= \|M_i^{-1}H_i r_i - \widetilde{M}_i^{-1}\widetilde{H}_i r_i\| \\ &\leqslant \|M_i^{-1} - \widetilde{M}_i^{-1}\|\|\widetilde{H}_i r_i\| + \|M_i^{-1}\|\|H_i - \widetilde{H}_i\|\|r_i\| \\ &\leqslant \|M_i^{-1}\widetilde{M}_i^{-1}\|\|M_i - \widetilde{M}_i\|\|\widetilde{H}_i\| + \|M_i^{-1}\|\|H_i - \widetilde{H}_i\| \\ &\leqslant O\left(\log\left(\frac{(m+1)}{\delta}\right)\right)\frac{D^2L^4 + L^4}{\sqrt{M_s}} \simeq O\left(\log\left(\frac{(m+1)}{\delta}\right)\right)\frac{D^2L^4}{\sqrt{M_s}},\end{aligned}$$

which completes the proof. □

Combining the above statement we can get the following Lemma for policy evaluation error, which is a restatement of our result in Lemma 6.

Lemma 12 (Policy Evaluation Error, restatement of Lemma 6)**.** *Suppose condition (1) in Lemma 4 holds. Suppose the sample size $M_s \geqslant \log\left(\frac{2(m+1)}{\delta/(Kn)}\right)$. Then, with probability at least $1-2\delta$, for every $i \in [n]$ and $k \in [K]$, the ground truth Q function $Q_i^{\pi(k)}(s,a)$ and the truncated Q function learnt in Algorithm 1 $\hat{Q}_i(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})$ satisfies, for any distribution $\bar{\nu}$ on $\mathcal{S}\times\mathcal{A}$,*

$$\begin{aligned}&\mathbb{E}_{\bar{\nu}}\left[|Q_i^{\pi(k)}(s,a) - \hat{Q}_i(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})|\right] \\ &\leqslant O\left(c\rho L^2 D\rho^{\kappa+1} + \log\left(\frac{(m+1)}{\delta/(Kn)}\right)\frac{D^2L^5}{\sqrt{M_s}} + L\frac{\epsilon_P\gamma\bar{r}}{1-\gamma}\left(\left\|\frac{\bar{\nu}}{\nu^o}\right\|_\infty + \left\|\frac{\nu_{\pi^{(k)}}}{\nu^o}\right\|_\infty\right)\right),\end{aligned}$$

where denoting

$$\tilde{\varphi}_{i,\kappa} := \tilde{\phi}_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}}), \tilde{\varphi}'_{i,\kappa} := \tilde{\phi}_{i,\kappa}(s'_{N_i^{\kappa+1}}, a'_{N_i^{\kappa+1}}),$$

$$\begin{aligned}&D := \max_{i\in[n],k\in[K]}\left\|(M_i^{(k)})^{-1}\right\|, \quad L := \max_{i\in[n]}\|\tilde{\varphi}_{i,\kappa}\|, \ \textit{where} \\ &M_i^{(k)} := \frac{1}{|D_k|}\sum_{s,a,s',a'\in D_k}\tilde{\varphi}_{i,\kappa}(\tilde{\varphi}_{i,\kappa} - \gamma\tilde{\varphi}'_{i,\kappa})^\top.\end{aligned}$$

Proof. Suppose the condition (1) in Lemma 4 holds. Consider any $i \in [n]$ and $k \in [K]$. From Lemma 10 and 11 we have that with probability at least $1-2\delta$,

$$\begin{aligned}&\mathbb{E}_{\bar{\nu}}\left[|Q_i^{\pi(k)}(s,a) - \tilde{\phi}_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})^\top w_i^{(k)}|\right] \\ &\leqslant \mathbb{E}_{\bar{\nu}}\left[|Q_i^{\pi(k)}(s,a) - \hat{Q}_i(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})|\right] + \mathbb{E}_{\bar{\nu}}\left[|\hat{Q}_i(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}}) - \tilde{\phi}_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})^\top w_i^{(k)}|\right] \\ &\leqslant \left(2c\rho^{\kappa+1} + \left\|\frac{\bar{\nu}}{\nu^o}\right\|_\infty\frac{\epsilon_P\gamma\bar{r}}{1-\gamma}\right) + \mathbb{E}_{\bar{\nu}}\left[|\tilde{\phi}_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})^\top(\hat{w}_i^{(k)} - w_i^{(k)})|\right] \\ &\leqslant \left(2c\rho^{\kappa+1} + \left\|\frac{\bar{\nu}}{\nu^o}\right\|_\infty\frac{\epsilon_P\gamma\bar{r}}{1-\gamma}\right) + L\left(\|\hat{w}_i^{(k)} - \tilde{w}_i^{(k)}\| + \|\tilde{w}_i^{(k)} - w_i^{(k)}\|\right)\end{aligned}$$

$$
\begin{aligned}
&\leqslant \left(2c\rho^{\kappa+1} + \left\|\frac{\bar{\nu}}{\nu^o}\right\|_\infty \frac{\epsilon_P \gamma \bar{r}}{1-\gamma}\right) + L\left(2LD(c\rho^{\kappa+1} + \left\|\frac{\nu}{\nu^o}\right\|_\infty \frac{\epsilon_P\gamma\bar{r}}{1-\gamma}) + O\left(\log\left(\frac{(m+1)}{\delta}\right)\right)\frac{D^2L^4}{\sqrt{M_s}}\right) \\
&= O\left(c\rho L^2 D\rho^{\kappa+1} + \log\left(\frac{(m+1)}{\delta}\right)\frac{D^2L^5}{\sqrt{M_s}} + L\frac{\epsilon_P\gamma\bar{r}}{1-\gamma}\left(\left\|\frac{\bar{\nu}}{\nu^o}\right\|_\infty + \left\|\frac{\nu}{\nu^o}\right\|_\infty\right)\right).
\end{aligned}
$$

The desired result then follows by rescaling $\delta := \delta/(Kn)$ and taking an union bound over all $i \in [n]$ and $k \in [K]$. □

7.7 Policy gradient analysis

We show now that our algorithm can find an approximate stationary point of the averaged discounted cumulative reward function $J(\pi^{(\theta)})$. For notational convenience, for a given set of policy parameters $\theta = (\theta_1, \dots, \theta_n)$, we define

$$
J(\theta) := J(\pi^\theta) = \mathbb{E}_{s\sim\mu_0}\mathbb{E}_{a(t)\sim\pi^{(\theta)}(\cdot|s(t))}\left[\sum_{t=0}^\infty \gamma^t r(s(t), a(t)) \mid s(0) = s\right],
$$

where we recall that $r(s,a) := \frac{1}{n}\sum_{i=1}^n r_i(s_i, a_i)$. From Lemma 0, we have that

$$
\begin{aligned}
\nabla_\theta J(\theta) &= \mathbb{E}_{s\sim d^\theta, a\sim\pi^\theta(\cdot|s)}\left[Q^\theta(s,a)\nabla_{\theta_i}\log\pi_i^\theta(a_i \mid s_{N_i^{\kappa_\pi}})\right] \\
&= \mathbb{E}_{s\sim d^\theta, a\sim\pi^\theta(\cdot|s)}\left[\frac{1}{n}\sum_{j=1}^n Q_j^\theta(s,a)\nabla_\theta \log\pi_i^{\theta_i}(a_i \mid s_{N_i^{\kappa_\pi}})\right]
\end{aligned}
$$

We first provide the following result, which shows that assuming Lipschitz continuity of the gradient of the objective function J as well as the gradients of $\log\pi^\theta$, there exists the following bound on the following weighted sum of the squared gradient norms.

Lemma 13. *Suppose that $\nabla J(\theta)$ is L'-Lipschitz continuous. Suppose that for each $i \in [n]$, $\left\|\nabla_{\theta_i}\log\pi_i^{\theta_i}(\cdot \mid \cdot)\right\| \leqslant L_{i,\pi}$. Denote $L_\pi := \sqrt{\sum_{i=1}^n L_{i,\pi}^2}$. Suppose for each round $k \in [K]$, $\theta^{(k+1)} = \theta^{(k)} - \eta\hat{g}^{(k)}$. Then,*

$$
\frac{1}{K}\sum_{k=1}^K\left\|\nabla J(\theta^{(k)})\right\|^2 \leqslant \frac{\bar{r}/(1-\gamma)}{\eta K} + \frac{1}{K}\sum_{k=1}^K\frac{L_\pi\bar{r}}{1-\gamma}\left\|\nabla J(\theta^{(k)}) - \hat{g}^{(k)}\right\| + \frac{1}{K}\sum_{k=1}^K L'\eta\left(\left\|\hat{g}^{(k)} - \nabla J(\theta^{(k)})\right\|^2 + \left(\frac{L_\pi\bar{r}}{1-\gamma}\right)^2\right).
$$

Proof. By the Lipschitz continuity of $\nabla J(\theta)$, we have

$$
\begin{aligned}
J(\theta^{(k+1)}) &\geqslant J(\theta^{(k)}) + \eta\left\langle\nabla J(\theta^{(k)}), \hat{g}^{(k)}\right\rangle - \frac{L'}{2}\left\|\eta\hat{g}^{(k)}\right\|^2 \\
&= J(\theta^{(k)}) + \eta\left\|\nabla J(\theta^{(k)})\right\|^2 + \eta\left\langle\nabla J(\theta^{(k)}), \hat{g}^{(k)} - \nabla J(\theta^{(k)})\right\rangle - \frac{L'\eta^2}{2}\left\|\hat{g}^{(k)}\right\|^2
\end{aligned}
$$

By rearranging and using a telescoping sum, we obtain

$$
\begin{aligned}
&\eta\sum_{k=1}^K\left\|\nabla J(\theta^{(k)})\right\|^2 \leqslant \sum_{k=1}^K(J(\theta^{(k+1)}) - J(\theta^{(k)})) + \eta\left\langle\nabla J(\theta^{(k)}), \nabla J(\theta^{(k)}) - \hat{g}^{(k)}\right\rangle + \frac{L'\eta^2}{2}\left\|\hat{g}^{(k)}\right\|^2 \\
\implies &\frac{1}{K}\sum_{k=1}^K\left\|\nabla J(\theta^{(k)})\right\|^2 \leqslant \frac{J(\theta^{(K+1)}) - J(\theta^{(1)})}{\eta K} + \frac{1}{K}\sum_{k=1}^K\left\|\nabla J(\theta^{(k)})\right\|\left\|\nabla J(\theta^{(k)}) - \hat{g}^{(k)}\right\| + \frac{1}{K}\sum_{k=1}^K\frac{L'\eta}{2}\left\|\hat{g}^{(k)}\right\|^2.
\end{aligned}
$$

Recall that $J(\theta^{(K+1)}) - J(\theta^{(1)}) \leqslant \bar{r}/(1-\gamma)$. Hence, by the given assumption on the bound on the derivative term $\nabla_{\theta_i}\log\pi_i^{\theta_i}$, it follows that $\|\nabla J(\cdot)\| \leqslant \frac{L_\pi\bar{r}}{1-\gamma}$. The desired bound then follows by plugging this in as well as using the triangle inequality to decompose $\left\|\hat{g}^{(k)}\right\|^2$. □

As we can tell from the above result, the crux to bounding the average stationarity gap after K rounds of optimization is the difference between the true gradient $\nabla J(\theta^{(k)})$ and the learned gradient $\hat{g}^{(k)}$ used in the update. In this next result, we bound this error, assuming that the truncated local $\hat{Q}_i$-functions are learned up to some error.

Lemma 14. *For any optimization round $k \in [K]$, let $\hat{\nu}^{(k)}$ denote the empirical distribution of the samples used during round k, i.e. $\{s(j), a(j)\}_{j \in [M_s]}$ where $(s(j), a(j)) \sim \nu_{\pi^{(k)}}$. Suppose that for each $\ell \in [n]$, the learnt $\hat{Q}_\ell$-value function satisfies the following error bound:*

$$\mathbb{E}_{\hat{\nu}^{(k)}} \left[\left| \hat{Q}_\ell^{(k)}(s_{N_\ell^{\kappa+1}}, a_{N_\ell^{\kappa+1}}) - Q_\ell(s,a) \right| \right] \leqslant \epsilon_Q \tag{6}$$

Suppose that for each $i \in [n]$, $\left\| \nabla_{\theta_i} \log \pi_i^{\theta_i}(\cdot \mid \cdot) \right\| \leqslant L_{i,\pi}$. Denote $L_\pi := \sqrt{\sum_{i=1}^n L_{i,\pi}^2}$. Then, for any optimization round $k \in [K]$, with probability at least $1-\delta$,

$$\left\| \hat{g}^{(k)} - \nabla_\theta J(\theta^{(k)}) \right\| \leqslant 2cL_\pi \rho^\kappa + \frac{2\bar{r}L_\pi}{1-\gamma} \sqrt{\frac{1}{M_s} \log\left(\frac{d_\theta + 1}{\delta/K} \right)} + \epsilon_Q L_\pi, \tag{7}$$

where the i-th component of the approximate gradient

$$\hat{g}_i^{(k)} := \frac{1}{M_s} \sum_{j=1}^{M_s} \frac{1}{n} \sum_{\ell \in N_i^{\kappa+\kappa_\pi}} \hat{Q}_\ell(s_{N_\ell^{\kappa+1}}(j), a_{N_\ell^{\kappa+1}}(j)) \nabla_{\theta_i} \log \pi_i^{(\theta_i^{(k)})}(a_i(j) \mid s_{N_i^{\kappa_\pi}}(j))$$

is defined in Line 9 of Algorithm 1.

Proof. For notational convenience, in the proof, we fix the optimization round $k \in [K]$, and hence, denote $\hat{g}_i := \hat{g}_i^{(k)}$, $\theta := \theta^{(k)}$ and $\hat{Q}_\ell := \hat{Q}_\ell^{(k)}$ unless otherwise specified. Moreover, we also denote $Q^\theta := Q^{\pi_\theta}$ for simplicity. From Lemma 0, for any agent $i \in [n]$, we have that

$$\begin{aligned} \nabla_{\theta_i} J(\theta) &= \mathbb{E}_{s \sim d^\theta, a \sim \pi^\theta(\cdot|s)} \left[Q^\theta(s,a) \nabla_{\theta_i} \log \pi_i^{\theta_i}(a_i \mid s_{N_i^{\kappa_\pi}}) \right] \\ &= \mathbb{E}_{s \sim d^\theta, a \sim \pi^\theta(\cdot|s)} \left[\frac{1}{n} \sum_{\ell=1}^n Q_\ell^\theta(s,a) \nabla_{\theta_i} \log \pi_i^{\theta_i}(a_i \mid s_{N_i^{\kappa_\pi}}) \right] \end{aligned}$$

To bound the difference between $\hat{g}_i$ and $\nabla_{\theta_i} J(\theta)$, we define the following intermediate terms.

We define the terms

$$\begin{aligned} g_i &:= \frac{1}{M_s} \sum_{j=1}^{M_s} \left(\frac{1}{n} \sum_{\ell \in N_i^{\kappa+\kappa_\pi}} Q_\ell^\theta(s(j), a(j)) \nabla_{\theta_i} \log \pi_i^{\theta_i}(a_i(j) \mid s_{N_i^{\kappa_\pi}}(j)) \right) \\ h_i &:= \mathbb{E}_{s \sim d^\theta, a \sim \pi^\theta(\cdot|s)} \left[\frac{1}{n} \sum_{\ell \in N_i^{\kappa+\kappa_\pi}} Q_\ell^\theta(s,a) \nabla_{\theta_i} \log \pi_i^{\theta_i}(a_i \mid s_{N_i^{\kappa_\pi}}) \right]. \end{aligned}$$

Then, we decompose the error as

$$\nabla_{\theta_i} J(\theta) - \hat{g}_i = \underbrace{(\nabla_{\theta_i} J(\theta) - h_i)}_{E_{J,h}} + \underbrace{(h_i - g_i)}_{E_{h,g}} + \underbrace{(g_i - \hat{g}_i)}_{E_{g,\hat{g}}}. \tag{8}$$

We proceed now to bound the three error terms in (8).

Error term $E_{J,h}$. We can bound the term $E_{J,h}$ as follows. For any $\ell \in [n]$ and positive integer κ, we define

$$\tilde{Q}_\ell^\theta(s_{N_\ell^\kappa}, a_{N_\ell^\kappa}) := \sum_{(s_{N_{-\ell}^\kappa})', (a_{N_{-\ell}^\kappa})'} w((s_{N_{-\ell}^\kappa})', (a_{N_{-\ell}^\kappa})') Q_\ell(s_{N_\ell^\kappa}, a_{N_\ell^\kappa}, (s_{N_{-\ell}^\kappa})', (a_{N_{-\ell}^\kappa})'),$$

where we let $w((s_{N_\ell^\kappa})', (a_{N_\ell^\kappa})')$ denote the uniform weight over the space $\mathcal{S}_{N_{-\ell}^\kappa} \times \mathcal{A}_{N_{-\ell}^\kappa}$. From Lemma 8, we know that

$$\left| Q_\ell^\theta(s,a) - \tilde{Q}_\ell^\theta(s_{N_\ell^\kappa}, a_{N_\ell^\kappa}) \right| \leqslant 2c\rho^\kappa. \tag{9}$$

We then have

$$\begin{aligned}
\nabla_{\theta_i} J(\theta) - h_i &= \mathbb{E}_{s\sim d^\theta, a\sim\pi^\theta(\cdot|s)} \left[\left(\frac{1}{n}\sum_{\ell=1}^n Q_\ell(s,a) - \frac{1}{n}\sum_{\ell\in N_i^{\kappa+\kappa_\pi}} Q_\ell^\theta(s,a) \right) \nabla_{\theta_i} \log \pi_i^{\theta_i}(a_i \mid s_{N_i^{\kappa_\pi}}) \right] \\
&= \mathbb{E}_{s\sim d^\theta, a\sim\pi^\theta(\cdot|s)} \left[\left(\frac{1}{n}\sum_{\ell\in N_{-i}^{\kappa+\kappa_\pi}} Q_\ell^\theta(s,a) \right) \nabla_{\theta_i} \log \pi_i^{\theta_i}(a_i \mid s_{N_i^{\kappa_\pi}}) \right] \\
&= \mathbb{E}_{s\sim d^\theta, a\sim\pi^\theta(\cdot|s)} \left[\left(\frac{1}{n}\sum_{\ell\in N_{-i}^{\kappa+\kappa_\pi}} \tilde{Q}_\ell^\theta(s_{N_\ell^\kappa}, a_{N_\ell^\kappa}) + \left(Q_\ell^\theta(s,a) - \tilde{Q}_\ell^\theta(s_{N_\ell^\kappa}, a_{N_\ell^\kappa}) \right) \right) \nabla_{\theta_i} \log \pi_i^{\theta_i}(a_i \mid s_{N_i^{\kappa_\pi}}) \right] \\
&= \mathbb{E}_{s\sim d^\theta, a\sim\pi^\theta(\cdot|s)} \left[\left(\frac{1}{n}\sum_{\ell\in N_{-i}^{\kappa+\kappa_\pi}} \left(Q_\ell^\theta(s,a) - \tilde{Q}_\ell^\theta(s_{N_\ell^\kappa}, a_{N_\ell^\kappa}) \right) \right) \nabla_{\theta_i} \log \pi_i^{\theta_i}(a_i \mid s_{N_i^{\kappa_\pi}}) \right] \\
&\quad + \mathbb{E}_{s\sim d^\theta, a\sim\pi^\theta(\cdot|s)} \left[\left(\frac{1}{n}\sum_{\ell\in N_{-i}^{\kappa+\kappa_\pi}} \tilde{Q}_\ell^\theta(s_{N_\ell^\kappa}, a_{N_\ell^\kappa}) \right) \nabla_{\theta_i} \log \pi_i^{\theta_i}(a_i \mid s_{N_i^{\kappa_\pi}}) \right] \\
&:= E_{J,h,1} + E_{J,h,2}
\end{aligned}$$

To bound $E_{J,h,1}$, utilizing the bound in (9) as well as the bound $\nabla_{\theta_i} \log \pi_i^{\theta_i}(a_i \mid s_{N_i}^\kappa) \leqslant L_{i,\pi}$ in the statement of the lemma, we have that

$$\|E_{J,h,1}\| \leqslant 2cL_{i,\pi}\rho^\kappa.$$

Meanwhile, observe that by definition, for any $\ell \in N_{-i}^{\kappa+\kappa_\pi}$, $\tilde{Q}_\ell^\theta(s_{N_\ell^\kappa}, a_{N_\ell^\kappa})$ does not depend on $s_{N_i^{\kappa_\pi}}$. Hence,

$$\begin{aligned}
E_{J,h,2} &= \mathbb{E}_{s\sim d^\theta, a\sim\pi^\theta(\cdot|s)} \left[\left(\frac{1}{n}\sum_{\ell\in N_{-i}^{\kappa+\kappa_\pi}} \tilde{Q}_\ell^\theta(s_{N_\ell^\kappa}, a_{N_\ell^\kappa}) \right) \nabla_{\theta_i} \log \pi_i^{\theta_i}(a_i \mid s_{N_i^{\kappa_\pi}}) \right] \\
&= \mathbb{E}_{s\sim d^\theta, a_{-i}\sim\pi^\theta(\cdot|s)} \left[\left(\frac{1}{n}\sum_{\ell\in N_{-i}^{\kappa+\kappa_\pi}} \tilde{Q}_\ell^\theta(s_{N_\ell^\kappa}, a_{N_\ell^\kappa}) \right) \mathbb{E}_{a_i\sim\pi_i^{\theta_i}(\cdot|s_{N_i^{\kappa_\pi}})} \left[\nabla_{\theta_i} \log \pi_i^{\theta_i}(a_i \mid s_{N_i^{\kappa_\pi}}) \right] \right] \\
&= \mathbb{E}_{s\sim d^\theta, a_{-i}\sim\pi^\theta(\cdot|s)} \left[\left(\frac{1}{n}\sum_{\ell\in N_{-i}^{\kappa+\kappa_\pi}} \tilde{Q}_\ell^\theta(s_{N_\ell^\kappa}, a_{N_\ell^\kappa}) \right) \nabla_{\theta_i} \left(\int_{a_i} \pi_i^{\theta_i}(a_i \mid s_{N_i^{\kappa_\pi}}) da_i \right) \right] \\
&= \mathbb{E}_{s\sim d^\theta, a_{-i}\sim\pi^\theta(\cdot|s)} \left[\left(\frac{1}{n}\sum_{\ell\in N_{-i}^{\kappa+\kappa_\pi}} \tilde{Q}_\ell^\theta(s_{N_\ell^\kappa}, a_{N_\ell^\kappa}) \right) \nabla_{\theta_i}(1) \right] = 0.
\end{aligned}$$

Thus $E_{J,h,2} = 0$. This implies then that

$$\|E_{J,h}\| = \|\nabla_{\theta_i} J(\theta) - h_i\| \leqslant 2cL_{i,\pi}\rho^\kappa. \tag{10}$$

Error term $E_{h,g}$. To bound $E_{h,g}$, we may use standard concentration inequalities. Observe that

$$E_{h,g} := h_i - g_i$$

$$= h_i - \frac{1}{M_s}\sum_{j=1}^{M_s}\left(\frac{1}{n}\sum_{\ell\in N_i^{\kappa+\kappa_\pi}} Q_\ell^\theta(s(j),a(j))\nabla_{\theta_i}\log\pi_i^{\theta_i}(a_i(j)\mid s_{N_i^{\kappa_\pi}}(j))\right)$$

$$= \frac{1}{M_s}\sum_{j=1}^{M_s}\underbrace{\left(h_i - \left(\frac{1}{n}\sum_{\ell\in N_i^{\kappa+\kappa_\pi}} Q_\ell^\theta(s(j),a(j))\nabla_{\theta_i}\log\pi_i^{\theta_i}(a_i(j)\mid s_{N_i^{\kappa_\pi}}(j))\right)\right)}_{E_{h,g}(j)}.$$

Since

$$h_i = \mathbb{E}_{s\sim d^\theta, a\sim\pi^\theta(\cdot|s)}\left[\frac{1}{n}\sum_{\ell\in N_i^{\kappa+\kappa_\pi}} Q_\ell^\theta(s,a)\nabla_{\theta_i}\log\pi_i^{\theta_i}(a_i\mid s_{N_i^{\kappa_\pi}})\right],$$

it follows that $\mathbb{E}\left[E_{h,g(j)}\right] = 0$. Moreover, using the fact that for any $\ell\in[n]$, θ and (s,a) pair, $0\leqslant Q_\ell^\theta(s,a)\leqslant\frac{\bar{r}}{1-\gamma}$, and the bound $\nabla_{\theta_i}\log\pi_i^{\theta_i}(a_i\mid s_{N_i^\kappa})\leqslant L_{i,\pi}$ in the assumption, we have $\|E_{h,g}(j)\|\leqslant\frac{2\bar{r}L_\pi}{1-\gamma}$. Using the i.i.d. assumption between the samples $j\in[M_s]$, we may apply Bernstein's concentration inequality for vectors (see Lemma 15) to find that for any $\delta>0$, with probability at least $1-\delta$,

$$\|E_{h,g}\|\leqslant\frac{2\bar{r}L_{i,\pi}}{1-\gamma}\sqrt{\frac{1}{M_s}\log\left(\frac{d_\theta+1}{\delta}\right)}, \tag{11}$$

where d_θ is the dimension of θ_i.

Error term $E_{g,\hat{g}}$. Observe that

$$g_i - \hat{g}_i = \frac{1}{M_s}\sum_{j=1}^{M_s}\frac{1}{n}\sum_{\ell\in N_i^{\kappa+\kappa_\pi}}\left(Q_\ell(s(j),a(j)) - \hat{Q}_\ell(s_{N_\ell^{\kappa+1}}(j),a_{N_\ell^{\kappa+1}}(j))\right)\nabla_{\theta_i}\log\pi_i^{(\theta_i^{(k)})}(a_i(j)\mid s_{N_i^{\kappa_\pi}}(j))$$

$$= \frac{1}{n}\sum_{\ell\in N_i^{\kappa+\kappa_\pi}}\frac{1}{M_s}\sum_{j=1}^{M_s}\left(Q_\ell(s(j),a(j)) - \hat{Q}_\ell(s_{N_\ell^{\kappa+1}}(j),a_{N_\ell^{\kappa+1}}(j))\right)\nabla_{\theta_i}\log\pi_i^{(\theta_i^{(k)})}(a_i(j)\mid s_{N_i^{\kappa_\pi}}(j))$$

$$\leqslant \frac{1}{n}\sum_{\ell\in N_i^{\kappa+\kappa_\pi}}\frac{1}{M_s}\sum_{j=1}^{M_s}\left|\left(Q_\ell(s(j),a(j)) - \hat{Q}_\ell(s_{N_\ell^{\kappa+1}}(j),a_{N_\ell^{\kappa+1}}(j))\right)\right|\left\|\nabla_{\theta_i}\log\pi_i^{(\theta_i^{(k)})}(a_i(j)\mid s_{N_i^{\kappa_\pi}}(j))\right\|$$

$$\overset{\text{(ix)}}{\leqslant}\frac{1}{n}\sum_{\ell\in N_i^{\kappa+\kappa_\pi}}\epsilon_Q\cdot L_{i,\pi}\leqslant\epsilon_Q\cdot L_{i,\pi}.$$

Above, (ix) follows from the bound in (6), as well as the bound $\left\|\nabla_{\theta_i}\log\pi_i^{\theta_i}(\cdot\mid\cdot)\right\|\leqslant L_{i,\pi}$ in the assumption.

Combining the bounds for $E_{J,h}, E_{h,g}$ and $E_{g,\hat{g}}$, we find that with probability at least $1-\delta$,

$$\|\nabla_\theta J(\theta)-\hat{g}\|\leqslant\sqrt{\sum_{i=1}^n\|\nabla_{\theta_i}J(\theta)-\hat{g}_i\|^2}\leqslant 2cL_\pi\rho^\kappa+\frac{2\bar{r}L_\pi}{1-\gamma}\sqrt{\frac{1}{M_s}\log\left(\frac{d_\theta+1}{\delta}\right)}+\epsilon_Q L_\pi$$

The final result then follows by applying a union bound over $k\in[K]$. □

We are now ready to state our main convergence result.

Theorem 2 (Restatement of Theorem 1)**.** *Suppose the sample size* $M_s\geqslant\log\left(\frac{2d_\kappa}{(\delta/Kn)}\right)$. *Suppose with probability at least* $1-\delta$, *for all* $i\in[n]$, *the following holds for some features* $\hat{\phi}_{i,\kappa}$ *and* $\hat{\mu}_{i,\kappa}$*:*

$$\mathbb{E}_{\nu^o}\left[\int_{s^+_{N_i^\kappa}}\left|P(s^+_{N_i^\kappa}|s_{N_i^{\kappa+1}},a_{N_i^{\kappa+1}})-\hat{\phi}_{i,\kappa}(s_{N_i^{\kappa+1}},a_{N_i^{\kappa+1}})^\top\hat{\mu}_{i,\kappa}(s^+_{N_i^\kappa})\right|ds^+_{N_i^\kappa}\right]\leqslant\epsilon_P$$

for some $\epsilon_P > 0$*. Then, if* $\eta = O(1/\sqrt{K})$*, we have that with probability at least* $1-4\delta$*,*

$$\frac{1}{K}\sum_{k=1}^{K}\left\|\nabla J(\theta^{(k)})\right\|^2 \leqslant O\left(\frac{\bar r/(1-\gamma)}{\sqrt{K}} + \frac{L_\pi \bar r \epsilon_J}{1-\gamma} + \frac{L'}{\sqrt{K}}\left(\epsilon_J^2 + \left(\frac{L_\pi \bar r}{1-\gamma}\right)^2\right)\right),$$

where

$$\epsilon_J := 2cL_\pi\rho^\kappa + \frac{2\bar r L_\pi}{1-\gamma}\sqrt{\frac{1}{M_s}\log\left(\frac{d_\theta+1}{\delta/K}\right)} + \epsilon_Q L_\pi,$$

and

$$\epsilon_Q := \max_{k\in\{0,1\ldots,K-1\}} O\left(c\rho L^2 D\rho^{\kappa+1} + \log\left(\frac{d_\kappa}{\delta}\right)\frac{D^2L^5}{\sqrt{M_s}} + L\frac{\epsilon_P\gamma\bar r}{1-\gamma}\left(\left\|\frac{\hat\nu^{(k)}}{\nu^o}\right\|_\infty + \left\|\frac{\nu_{\pi^{(k)}}}{\nu^o}\right\|_\infty\right)\right).$$

Proof. Fix a $\delta > 0$. Suppose the condition in (1) holds with probability at least $1-\delta$ for all $i \in [n]$ for a distribution ν^o over $\mathcal{S}\times\mathcal{A}$. In other words, with probability at least $1-\delta$, for all $i\in[n]$, the following holds:

$$\mathbb{E}_{\nu^o}\left[\int_{s^+_{N_i^\kappa}}\left|P(s^+_{N_i^\kappa}|s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}}) - \hat\phi_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})^\top\hat\mu_{i,\kappa}(s^+_{N_i^\kappa})\right| ds^+_{N_i^\kappa}\right] \leqslant \epsilon_P$$

for some $\epsilon_P > 0$. Then, by Lemma 6, it follows that with probability at least $1-3\delta$, for every $i\in[n]$ and optimization round $k\in[K]$, we have

$$\mathbb{E}_{\hat\nu^{(k)}}\left[\left|\hat Q_\ell^{(k)}(s_{N_\ell^{\kappa+1}}, a_{N_\ell^{\kappa+1}}) - Q_\ell(s,a)\right|\right] \leqslant \epsilon_Q^{(k)},$$

where

$$\epsilon_Q^{(k)} := O\left(c\rho L^2 D\rho^{\kappa+1} + \log\left(\frac{d_\kappa}{\delta/(Kn)}\right)\frac{D^2L^5}{\sqrt{M_s}} + L\frac{\epsilon_P\gamma\bar r}{1-\gamma}\left(\left\|\frac{\hat\nu^{(k)}}{\nu^o}\right\|_\infty + \left\|\frac{\nu_{\pi^{(k)}}}{\nu^o}\right\|_\infty\right)\right).$$

Note that by Lemma 13, with probability at least $1-\delta$, for every optimization round $k\in[K]$, we have

$$\left\|\hat g^{(k)} - \nabla_\theta J(\theta^{(k)})\right\| \leqslant 2cL_\pi\rho^\kappa + \frac{2\bar r L_\pi}{1-\gamma}\sqrt{\frac{1}{M_s}\log\left(\frac{d_\theta+1}{\delta/K}\right)} + \epsilon_Q^{(k)}L_\pi.$$

Thus, by picking $\eta = O(1/\sqrt{K})$, using union bound, with probability at least $1-4\delta$, we have

$$\frac{1}{K}\sum_{k=1}^{K}\left\|\nabla J(\theta^{(k)})\right\|^2 \leqslant O\left(\frac{\bar r/(1-\gamma)}{\sqrt{K}} + \frac{L_\pi \bar r \epsilon_J}{1-\gamma} + L'\eta\left(\epsilon_J^2 + \left(\frac{L_\pi \bar r}{1-\gamma}\right)^2\right)\right),$$

where

$$\epsilon_J := 2cL_\pi\rho^\kappa + \frac{2\bar r L_\pi}{1-\gamma}\sqrt{\frac{1}{M_s}\log\left(\frac{d_\theta+1}{\delta/K}\right)} + \max_{k\in[K]}\epsilon_Q^{(k)}L_\pi.$$

□

7.8 Concentration inequalities

Lemma 15 (Matrix Bernstein)**.** *Suppose* $\{M_k\}_{k=1}^n$ *are i.i.d random matrices where* $M_k\in\mathbb{R}^{d_1\times d_2}$ *and that*

$$\|M_k - \mathbb{E}M_k\| \leqslant C,$$

then for a given $\delta\in(0,1)$ *and* $n \geqslant \log\left(\frac{d_1+d_2}{\delta}\right)$*, we have*

$$\Pr\left(\frac{1}{n}\left\|\sum_{k=1}^{n}(M_k - \mathbb{E}M_k)\right\| \geqslant 2C\sqrt{n^{-1}\log\left(\frac{d_1+d_2}{\delta}\right)}\right) \leqslant \delta,$$

Proof. Let $\epsilon := 2C\sqrt{n^{-1}\log\left(\frac{d_1+d_2}{\delta}\right)}$, then since $n \geqslant \log\left(\frac{d_1+d_2}{\delta}\right)$, we have $\epsilon \leqslant 2C$.

Now we can apply the matrix Bernstein inequality (Theorem 6.1.1 in [Tropp et al., 2015]) and get that

$$\begin{aligned}\Pr\left(\frac{1}{n}\left\|\sum_{k=1}^{n}(M_k - \mathbb{E}M_k)\right\| \geqslant \epsilon\right) &\leqslant (d_1+d_2)\exp\left(\frac{-n^2\epsilon^2/2}{nC^2 + Cn\epsilon/3}\right)\\ &\leqslant (d_1+d_2)\exp\left(\frac{-n^2\epsilon^2/2}{nC^2+nC^2}\right) = (d_1+d_2)\exp\left(\frac{n\epsilon^2}{4C^2}\right)\end{aligned}$$

Substituting $\epsilon = 2C\sqrt{n^{-1}\log\left(\frac{d_1+d_2}{\delta}\right)}$ into the right hand side of the equation we get

$$\Pr\left(\frac{1}{n}\left\|\sum_{k=1}^{n}(M_k - \mathbb{E}M_k)\right\| \geqslant \epsilon\right) \leqslant \delta,$$

which completes the proof. □

7.9 Simulation details

All code for this project is available as a zip folder with the supplementary material.

7.9.1 Thermal control of multi-zone building

Problem setup details. In the simulations, we consider a discrete-time linear thermal dynamics model adapted from [Zhang et al., 2016, Li et al., 2021], where for any $i \in [n]$,

$$x_i(t+1) - x_i(t) = \frac{\Delta}{v_i \zeta_i}(\theta^o(t) - x_i(t)) + \sum_{j \in N_i} \frac{\Delta}{v_i \zeta_{ij}}(x_j(t) - x_i(t)) + \frac{\Delta}{v_i}\alpha_i a_i(t) + \sqrt{\frac{\Delta}{v_i}}\beta_i w_i(t),$$

where $x_i(t)$ denotes the temperature of zone i at time t, $a_i(t)$ denotes the control input of zone i that is related with the air flow rate of the HVAC system, $\theta^o(t)$ denotes the outdoor temperature, π_i represents a constant heat from external sources to zone i, $w_i(t)$ represents random disturbances, Δ is the time resolution, v_i is the thermal capacitance of zone i, ζ_i represents the thermal resistance of the windows and walls between zone i and the outside environment, ζ_{ij} represents the thermal resistance of the walls between zone i and j, and α_i and β_i denote scaling factors on the input and noise respectively. The local reward is defined as

$$r_i(t) = -\rho_i((x_i(t) - \theta_i^*))^2 + a_i(t)^2,$$

where θ_i^* is the target temperature and ρ_i is a trade-off parameter.

The parameters in the dynamics and rewards are set as follows. For simplicity, we center the temperatures at 0, and hence set the target θ_i^* to be 0. We set $\rho_i = 3$. We set the following parameters for the dynamics: $\Delta = 20, v_i = 200, \zeta_{ij} = 1, \zeta_i = \frac{1}{2}, \alpha_i = \frac{1}{7}, \beta_i = \sqrt{\frac{v_i}{\Delta}}, \theta^0 = 0$.

We also assume $w_i(t)$ to be drawn iid from $N(0,1)$. We set the discount factor in the problem to be 0.75, and (when collecting data) set the horizon length of each episode to be 20.

Connectivity. In this problem, there are $n = 50$ agents, and the agents have circular connectivity and has two neighbors each, such that agent 1 is connected to agents N and agent 2, agent 2 is connected to agents 1 and 3, so on and so forth.

Experimental details. We assume knowledge of the dynamics and rewards. For policy truncation parameter $\kappa_\pi = 0, 1, 2, 3$, we use $\kappa = 0, 1, 2, 2$ respectively[4] as the evaluation κ parameter. We now explain the simulation setup for our implementation of Algorithm 1 with random features, as well as the benchmark algorithm using a two-hidden layer NN.

1. (Spectral embedding generation step). For Algorithm 1 with random features, for each agent i, we use random feature dimension of $m = 30, 50, 800, 800$ for each of the four experiments ($\kappa_\pi = 0, 1, 2, 3$) to represent the function $T^\pi(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}}) := \frac{Q_i^\pi(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}}) - r_i(s_i, a_i)}{\gamma}$. For the NN implementation, we used a two-hidden layer NN with 128 neurons to represent the function $T^\pi(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})$.

2. (Policy evaluation step) We used $M_s = 100, 200, 500, 1000$ episodes respectively for each of the four experiments ($\kappa_\pi = 0, 1, 2, 3$) to perform the policy evaluation. For the random features implementation, we used the least squares method in Algorithm 1 to compute the new weights for the local value functions. For the NN implementation, we ran batch gradient descent, and used a target network with update rate of 0.005.

3. (Policy update step). For both implementations, we normalize the policy gradient, and run gradient descent with $\eta = 0.2$.

7.9.2 Kuramoto synchronization

Problem setup details. We recall the setup described earlier in the paper. We consider here a Kuramoto system with n agents, with an underlying graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N} = \{1, \ldots, n\}$ is the set of agents and $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$ is the set of edges. The state of each agent i is its phase $\theta_i \in [-\pi, \pi]$, and the action of each agent is

[4] We found in practice that using $\kappa = 3$ for $\kappa_\pi = 3$ performed less well in this specific example.

a scalar $a_i \in \mathcal{A}_i \subset \mathbb{R}$ in a bounded subset of $\mathbb{R}$. The dynamics of each agent is influenced only by the states of its neighbors as well as its own action, satisfying the following form in discrete time [Mozafari et al., 2012]:

$$\theta_i(t+1) = \theta_i(t) + dt \underbrace{\left(\omega_i(t) + a_i(t) + \left(\sum_{j \in N_i} K_{ij} \sin(\theta_j - \theta_i)\right)\right)}_{:=\dot{\theta}_i(t)} + \epsilon_i(t).$$

Above, ω_i denotes the natural frequency of agent i, dt is the discretization time-step, K_{ij} denotes the coupling strength between agents i and j, $a_i(t)$ is the action of agent i at time t, and $\epsilon_i(t) \sim N(0, \sigma^2)$ is a noise term faced by agent i at time t. We note that this fits into the localized transition considered in network MDPs. For the reward, we consider frequency synchronization to a fixed target ω_{target}. In this case, the local reward of each agent can be described as $r_i(\theta_{N_i}, a_i) = -\left|\dot{\theta}_i - \omega_{\text{target}}\right|$.

The parameters in the dynamics and rewards are set as follows. We set the target ω_{target} to be 0.75. We set the action space as $[-3, 3]$. For agents i and j that are connected, we sample K_{ij} uniformly at random from $[0.2, 1.2]$. For the natural frequency ω_i's, we sample them iid uniformly at random from $[0, 1.5]$. For the noise, we sample $\epsilon_i(t) \sim N(0, 0.01^2)$. The time resolution is $dt = 0.01$.

We also assume $w_i(t)$ to be drawn iid from $N(0, 1)$. We set the discount factor in the problem to be 0.99, and set the horizon length to be 800 steps.

Connectivity. In this problem, there are $n = 20$ agents, and the agents have circular connectivity and has two neighbors each, such that agent 1 is connected to agents N and agent 2, agent 2 is connected to agents 1 and 3, so on and so forth.

Experimental details (model-free). In this case, we do not assume access to the dynamics function. We now explain the simulation setup for our implementation of Algorithm 1 with spectral features, as well as the benchmark algorithm using a two-hidden layer NN.

1. (Policy evaluation step) For both the spectral feature and NN implementation, the features are the last layer of a two-hidden layer neural network with hidden dimension 256. At each iteration, for each agent i, we draw a batch (of 128 transitions) from the replay buffer and we run 1 step of gradient descent on the least square bellman error, and used a target network with update rate of 0.005.

2. (Policy update step). For each agent i, the policy is parameterized to be a 3-hidden layer NN which outputs the mean and standard deviation of the agent's action, and the input is s_{N_i}, i.e. the states of the neighborhood of agent i. We update the policy parameters $\{\theta_i\}_{i=1}^n$ by taking one gradient descent step on the following objective:

$$J_\pi(\theta) = \mathbb{E}_{s \sim \mathcal{D}}\left[D_{\text{KL}}\left(\prod_{i=1}^n \pi_{\theta_i}(\cdot_i \mid s_{N_i}) \middle\| \frac{\exp(\sum_{i=1}^n \hat{Q}_i(s_{N_i^{\kappa+1}}, \{\cdot_i\}_{i=1}^n))}{Z(s)}\right)\right],$$

   where $\mathcal{D}$ is a set of data from the replay buffer, $Z(s)$ is a normalization constant. Above, we assume the temperature parameter τ to be 1. This objective is identical to the implementation in Soft-Actor-Critic (SAC) [Haarnoja et al., 2018] but for factored policies, as well as using the learned $\hat{Q}_i$-value functions to approximate the value function.

3. (Feature step). For the spectral features, for each agent i, we add an additional feature step, which seeks to regularize the features such that they approximate the top left eigenfunctions of the probability transition $P(s'_{N_i^\kappa} \mid s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})$, by taking a gradient descent step on the following objective to update agent i's feature $\phi(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})$:

$$\min_{\phi = \{\phi_1, \ldots, \phi_L\}} \mathbb{E}_{d(s,a)}[\|\phi(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})\|^2] - 2\mathbb{E}_{d(s,a), s' \sim P(\cdot|s,a)}[\omega(s'_{N_i^\kappa})^\top \phi(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})], \tag{12}$$

where in practice we pick $d(s,a)$ to be the a set of samples from the current replay buffer. We note that (12) can be seen as a randomized way to compute the singular value decomposition (SVD) of the transition operator $P(s'_{N_i^\kappa} \mid s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})$, and we picked it due to its better numerical performance compared to existing spectral decomposition methods in the literature [Ren et al., 2022b], in our simulation example. We also note that unlike for the model-based case (with random features), we do not have guarantees on the end-to-end performance of the model-free version of the algorithm. However, we note that the feature step encourages the features to minimize the objective in (1). We leave more detailed analysis of this to future work. We give more details on the derivation of (12) in Appendix 7.9.3.

Experimental details and results (model-based). In this case, we do assume knowledge of the dynamics function and that the noise is Gaussian, allowing us to use random features as the spectral features.

We focus on discussing the feature generation step, since this is the only difference with the previous model-free case. For the random features, for each agent i, we select the random features according to the procedure in Lemma 5 (with feature dimension being 1024), and in the simulations we set $\alpha = 0$. For the NN implementation, we use a two-hidden layer NN with hidden dimension 256. For the NN implementation, for a fair comparison, we also give it knowledge of the dynamics function $f_{i,\kappa}(s_{N_i^{\kappa+1}}, a_{N_i^{\kappa+1}})$, such that it can use this information when computing the local value functions. We note that for the policy evaluation step, for both random features and NN, we perform gradient descent on the Bellman least square error.

The results of the learning performance are shown below. We see that while the spectral-based method has more variance initially, it soon displays comparable performance to the NN-based implementation.

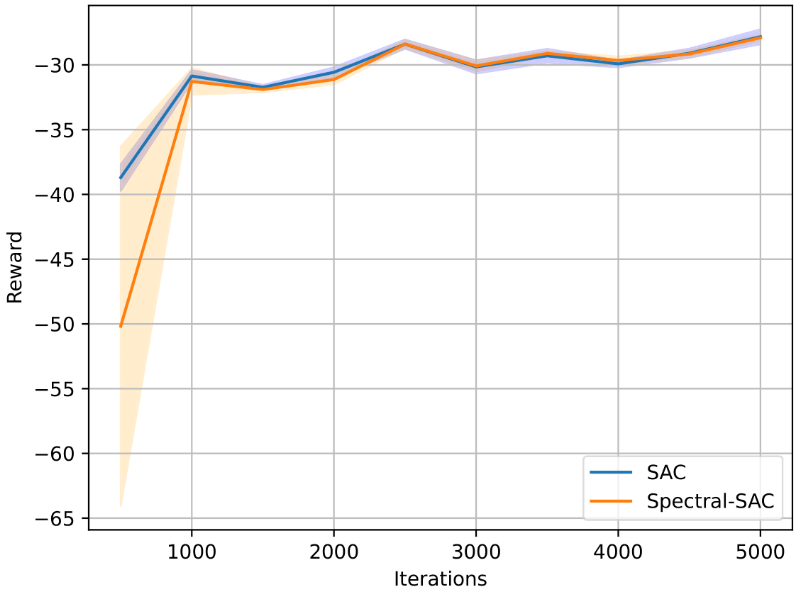


Figure 4: Change in reward during training for Kuramoto oscillator control, $N = 20$, $\kappa_\pi = 1, \kappa = 2$. In this experiment, the dynamics model is known. The performance for each algorithm is averaged over 5 seeds.

7.9.3 Randomized Spectral Decomposition

We give here a more detailed derivation of the objective in (12). For simplicity, we focus on the single-agent case, when we are trying to decompose $P(s' \mid s,a)$ as $P(s' \mid s,a) \approx \phi(s,a)^\top \mu(s')$, and in particular trying to find the $\phi(s,a)$ in this decomposition. As suggested in [Ren et al., 2022b], this is akin to finding the top left eigenfunctions of $P(s' \mid s,a)$. Motivated by randomized SVD for computing the top left singular vectors of finite-dimensional matrices, in the functional space setting, we can perform an analogous randomized SVD to learn the top left eigenfunctions of $P(s' \mid s,a)$ according to the following procedure.

1. Fix a positive integer L.
2. For each $i \in [L]$, sample a random function $\omega_i(s') \in \mathbb{R}$, e.g. $\omega_i(s') = \cos(\alpha_i^\top s' + \beta_i)$, where $\alpha_i \sim N(0, I_S)$ and $\beta_i \sim \mathrm{Unif}([0, 2\pi])$.
3. For each $i \in [L]$, learn a $\phi_i(s,a)$ that approximates $P\omega_i(s,a) := \int_{s'} P(s' \mid s,a)\omega_i(s')ds'$ as follows:
   (a) Pick a sampling distribution $d(s,a)$, e.g. uniform distribution.
   (b) For each $i \in [L]$, solve

$$\begin{aligned} & \min_{\phi_i} \int_{s,a} d(s,a)\left(\phi_i(s,a) - P\omega_i(s,a)\right)^2 \\ \Longleftrightarrow\ & \min_{\phi_i} \int_{s,a} d(s,a)\left(\phi_i(s,a) - \int_{s'} P(s' \mid s,a)\omega_i(s')\right)^2 \\ \Longleftrightarrow\ & \min_{\phi_i} \int_{s,a} d(s,a)\phi_i(s,a)^2 - 2\int_{s,a} d(s,a)\int_{s'} ds' P(s' \mid s,a)\omega_i(s')\phi_i(s,a) \\ \Longleftrightarrow\ & \min_{\phi_i} \mathbb{E}_{d(s,a)}[\phi_i(s,a)^2] - 2\mathbb{E}_{d(s,a),s' \sim P(\cdot|s,a)}[\omega_i(s')\phi_i(s,a)] \end{aligned}$$

We note that the final objective is equivalent to solving the L ϕ_i's jointly which is single-agent analogue of the objective in (12):

$$\min_{\phi=\{\phi_1,\ldots,\phi_L\}} \mathbb{E}_{d(s,a)}[\|\phi(s,a)\|^2] - 2\mathbb{E}_{d(s,a),s' \sim P(\cdot|s,a)}[\omega(s')^\top \phi(s,a)].$$