Variance-Reduced Q-Learning over Static and Time-Varying Networks

Sreejeet Maity[†,*], Feng Zhu[†,*], Aritra Mitra[*], and Robert W. Heath Jr.

***Abstract*— We investigate a decentralized reinforcement learning problem involving multiple agents that interact with the same Markov Decision Process (MDP). The agents can exchange information over a network to collectively learn the optimal state-action value function. For this setting, we introduce a novel epoch-based distributed Q-learning algorithm called VRDQ, where within each epoch, agents locally estimate the Bellman optimality operator and diffuse information using a consensus-based protocol. For both static and time-varying networks, we establish high-probability finite-time convergence rates for VRDQ that enjoy linear speedups from collaboration. Crucially, we prove that such speedups in sample-complexity require only $\tilde{O}(1)$ communication, substantially improving upon the communication costs in prior work.**

I. Introduction

Given the promise of cooperative multi-agent reinforcement learning (MARL) in improving the sample efficiency of online decision-making, we consider a decentralized RL setting involving N agents that can exchange information over a (potentially time-varying) network. Each agent interacts with a *common environment* modeled as a Markov Decision Process (MDP), with the goal of learning an optimal policy that maximizes a long-term cumulative return. In the single-agent setting, such an optimal policy can be learned using the celebrated Q-learning algorithm [1]. Given the collective information available across the network in our setting, we focus on answering two basic questions: (i) (**Sample Efficiency**) By exchanging information, can each agent learn the optimal policy using fewer samples (relative to the single-agent case)? (ii) (**Communication Efficiency**) If so, what is the *communication overhead* required to realize such collaborative speedups? Perhaps surprisingly, as we discuss below, these questions remain unresolved even for simple tabular RL problems, thereby motivating our current study.

Related Work. Asymptotic convergence guarantees for a decentralized variant of the classical Q-learning algorithm were first provided in [2]; similar results for actor-critic algorithms were later derived in [3], [4]. However, to characterize explicit statistical gains from collaboration, one requires a finer non-asymptotic analysis absent in these papers. In a series of follow-up papers, finite-time rates were derived for decentralized temporal difference learning in [5], Q-learning in [6], [7], and general stochastic approximation in [8]. However, there are two key limitations to all the aforementioned works. First, the final performance bounds do not explicitly demonstrate any benefit of collaboration between agents. Moreover, each method incurs a communication cost that scales linearly with the time horizon, i.e., the total number of samples. This creates a natural tension: *with existing decentralized RL methods, achieving a high accuracy by gathering more samples comes at the expense of commensurately high communication costs, hindering their practical deployment in resource-constrained environments.*

In this work, we resolve the above tension by developing a novel distributed Q-learning algorithm that (i) enjoys *near-optimal* statistical benefits of collaboration, and (ii) incurs a communication cost that scales only *poly-logarithmically* in the total number of samples, marking a significant improvement over existing methods that require linear-in-time communication costs. Our results apply to both static and time-varying networks, and, as such, are significantly more general in scope relative to some recent papers that assume a central coordinator [9]–[11]. Importantly, our proposed algorithm has a structure that is fundamentally different from all the papers we have reviewed thus far.

• **Algorithmic Contribution.** We introduce a new decentralized RL algorithm called Variance-Reduced Diffused Q-learning (VRDQ) that combines two crucial ingredients: *local operator estimation* and *diffusion*. In contrast to standard approaches that update the Q-function (or other relevant parameters) at *every* time-step using noisy update directions that suffer from high variance, our approach relies on making fewer *infrequent* updates using low-variance directions obtained by locally estimating the Bellman optimality operator. The low frequency of updates, in turn, directly translates into the low communication overhead of our algorithm. The second key ingredient of our approach is to show how the local operator estimation phase can be run in parallel with an average-consensus-based diffusion phase meant to exchange information. The decoupled nature of these phases allows us to easily disentangle statistical errors from network-induced errors, leading to a simple overall analysis.

• **Theoretical Contribution.** Our first main result, namely Theorem 1, pertains to static networks, and establishes a high-probability finite-sample convergence rate of $\tilde{\mathcal{O}}(1/\sqrt{NT})$ for our proposed algorithm VRDQ, where T is the number of samples per agent, and N is the number of agents. This result reveals a clear benefit of collaboration over the single-agent rate of $\tilde{\mathcal{O}}(1/\sqrt{T})$ recently established in [12]–[14]. In Theorem 2, we prove that VRDQ continues to enjoy the same collaborative gains for a fairly general class of time-varying networks. Crucially, for both static and time-varying networks, we show that such collaborative gains can be achieved with just $\mathcal{O}(\log^2(NT))$ communication.

[†]Equal Contribution. [*]The authors are with the Department of Electrical and Computer Engineering, North Carolina State University. Email: {smaity2,fzhu5,amitra2}@ncsu.edu. Robert W. Heath Jr. is with the Department of Electrical and Computer Engineering, University of California San Diego, San Diego, USA. Email: rwheathjr@ucsd.edu.

II. Notation and Problem Formulation

Graph model. We start by introducing our network model. Let $\mathcal{V} = \{1, 2, \ldots, N\}$ be a set of N agents interacting with the same environment (modeled as an MDP). The network is an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes (agents), and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges representing communication links between agents. Since the graph is undirected, we have $(i, j) \in \mathcal{E}$ if and only if $(j, i) \in \mathcal{E}$. We say that agent j is a *neighbor* of agent i if $(i, j) \in \mathcal{E}$, and define the *neighbor set* of agent i as the set of all its neighbors (including itself): $\mathcal{N}_i = \{j \mid (i, j) \in \mathcal{E}\}$. We associate a *mixing matrix* $W \in \mathbb{R}^{N \times N}$ with the network, where the entry $(W)_{ij}$ denotes the weight that agent i assigns to agent j's information. If $i \neq j$, and $(i, j) \notin \mathcal{E}$, then $(W)_{ij} = 0$. The mixing matrix W plays a central role in distributed algorithms, as it governs how agents combine information from their neighbors. As is standard, we assume that W is symmetric and *doubly stochastic*, i.e., $W = W^\top$, all entries are non-negative, and each row and column sums to one. Later, in Section VI, we will see how our results can be easily generalized to time-varying networks.

MDP Model. We now introduce the basic MDP notation used in this paper. We assume that the agents in $\mathcal{V}$ interact with the same environment, which can be modeled as an MDP $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \gamma\}$, where $\mathcal{S}$ and $\mathcal{A}$ are the finite state and action spaces whose cardinalities are denoted as S and A, and $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, with $\mathcal{R}(s, a)$ denoting the immediate deterministic reward received by playing action a at state s. Throughout this paper, we assume bounded rewards with $|\mathcal{R}(s, a)| \leq \bar{R}$, where $\bar{R} > 0$ is some constant. The object $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the transition kernel, with $\mathcal{P}(s' \mid s, a)$ denoting the probability of transitioning to the next state s' by playing action a at state s. Finally, $\gamma \in (0, 1)$ is the discount factor.

The behavior of an agent is captured by a stochastic *policy* $\pi : \mathcal{S} \to \Delta(\mathcal{A})$, which outputs a probability distribution over the action space $\mathcal{A}$ at a given state $s \in \mathcal{S}$. It is then natural to develop a metric to measure the *goodness* of a policy π when the agent interacts with the MDP $\mathcal{M}$ following this policy. This leads to the concept of the *value function* $V^\pi \in \mathbb{R}^S$, defined as the expected infinite-horizon cumulative discounted reward starting from state $s \in \mathcal{S}$:

$$V^\pi(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_t, a_t) \;\middle|\; s_0 = s, \pi\right], \tag{1}$$

where the expectation is taken over the randomness w.r.t. state transitions and the stochastic policy π; here, s_t and a_t represent the state and action, respectively, at time t. Similarly, we introduce the concept of the *state-action value function*, or the *Q-function* $Q^\pi \in \mathbb{R}^{S \times A}$, which evaluates the policy starting from state s under initial action a:

$$Q^\pi(s, a) = \mathcal{R}(s, a) + \mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^t \mathcal{R}(s_t, a_t) \;\middle|\; \pi\right]. \tag{2}$$

Q-Learning. The general objective of an agent is to find the optimal policy π^* that maximizes the value function V^π for all states $s \in \mathcal{S}$. Unlike in dynamic programming problems, in our RL setup, the underlying MDP model comprising the transition kernels and the reward functions is *unknown* to the agent. In this setting, the celebrated Q-learning algorithm [1] learns π^* by first iteratively estimating the optimal Q-function $Q^* \in \mathbb{R}^{S \times A}$, and then extracting an associated optimal policy π^* via greedy action selection: $\pi^*(s) = \operatorname{argmax}_{a \in \mathcal{A}} Q^*(s, a)$ at each state $s \in \mathcal{S}$. The core idea behind Q-learning is to exploit the fact that Q^* is the unique fixed-point of the *Bellman optimality operator* $\mathcal{T}^* : \mathbb{R}^{S \times A} \to \mathbb{R}^{S \times A}$ defined below [15]:

$$\mathcal{T}^* f(s, a) := \mathcal{R}(s, a) + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}(s' \mid s, a) \max_{a' \in \mathcal{A}} f(s', a'), \tag{3}$$

$\forall f \in \mathbb{R}^{S \times A}$. To run Q-learning, one maintains a sequence of noisy empirical estimates of $\mathcal{T}^*$ using data generated by a suitable sampling model. In this work, we will consider the popular *generative synchronous sampling model* [12], [14], [16], [17] where an agent makes observations of the following form. At each time step $t = 0, 1, \ldots$, for **every** state-action pair $(s, a) \in \mathcal{S} \times \mathcal{A}$, the agent independently samples a next state $s_t(s, a) \sim \mathcal{P}(\cdot \mid s, a)$, and observes an immediate deterministic reward $\mathcal{R}(s, a)$. The agent then constructs a *noisy empirical* estimate $\mathcal{T}_t : \mathbb{R}^{S \times A} \to \mathbb{R}^{S \times A}$ of the Bellman optimality operator $\mathcal{T}^*$, defined as follows:

$$\mathcal{T}_t f(s, a) = \mathcal{R}(s, a) + \gamma \max_{a' \in \mathcal{A}} f(s_t(s, a), a'), \tag{4}$$

$\forall f \in \mathbb{R}^{S \times A}$. Using $\mathcal{T}_t$, the Q-learning algorithm updates the Q-estimate as follows $\forall (s, a) \in \mathcal{S} \times \mathcal{A}$:

$$Q_{t+1}(s, a) = (1 - \alpha_t) Q_t(s, a) + \alpha_t \mathcal{T}_t Q_t(s, a), \tag{5}$$

where $Q_t \in \mathbb{R}^{S \times A}$ is the estimated Q-function at time-step t, and $\{\alpha_t\}$ is a suitable step-size sequence. Classical asymptotic results show that the sequence of iterates $\{Q_t\}$ generated by (5) converges to Q^* almost surely [18]. More recent work [12]–[14] has established non-asymptotic guarantees, revealing that with T samples, the estimation error $\|Q_T - Q^*\|_\infty$ is $\tilde{\mathcal{O}}\left(1/\sqrt{T}\right)$, with high probability.

Networked Q-Learning Problem. We can now state our problem of interest. Suppose that each agent in $\mathcal{V}$ is allowed to acquire, in parallel, T *statistically independent* samples per state-action pair, through T queries to a generative model. Acting independently, each agent can trivially achieve a convergence rate of $\tilde{\mathcal{O}}\left(1/\sqrt{T}\right)$, following standard results from single-agent Q-learning. Given that there are N agents in total, we ask: *Can agents collaborate to achieve an accelerated rate of* $\tilde{\mathcal{O}}\left(1/\sqrt{NT}\right)$*?* This question is non-trivial due to the decentralized nature of the setting: agents communicate over a graph that is **not assumed to be fully connected**, i.e., $\mathcal{N}_i \neq \mathcal{V}$ in general. Therefore, to achieve a linear speedup w.r.t. N, agents must *diffuse* their information throughout the network to collectively approximate Q^* at an improved $\tilde{\mathcal{O}}\left(1/\sqrt{NT}\right)$ rate. We further require such a bound to hold with probability at least $1-\delta$, where $\delta \in (0, 1)$ is a prescribed failure probability.

In this context, we ask two basic questions: **(i)** *What* information should each agent diffuse with others? **(ii)** How *frequently* should such information be diffused? We provide concrete answers to these questions in the following section, where we present our algorithmic framework in detail. Before doing so, the following remark is in order.

Remark 1. *While one can certainly consider more involved RL settings accounting for function approximation and/or asynchronous, Markovian sampling, we restrict our attention to a tabular setting with synchronous sampling for the following reasons. First, the generative synchronous sampling model we consider here has been extensively used in the theoretical analysis of RL algorithms [12], [14], [16], [17]; in a similar vein, tabular Q-learning has been used for building key insights in both single [12]–[14] and multi-agent RL [10]. Second, the tabular setting we study here enables us to better convey the new algorithmic aspects of our approach. Third, and most importantly, a fundamental understanding of the amount of communication needed to achieve statistical collaborative gains under network constraints is not well understood for the setting in our paper.*

III. Variance-Reduced Diffused Q-Learning

Algorithm 1 Variance-Reduced Diffused Q-Learning (VRDQ)

Require: Total samples per agent T, epoch length H, diffusion period L, failure probability δ.

Initialize $Q_{i,0}(s,a) \leftarrow 0, \forall (s,a) \in \mathcal{S} \times \mathcal{A}$ and $i \in [N]$.
for epoch $k = 0$ to $K-1$ **do**
  for $i \in [N]$ **do**
    `Estimate` $\mathcal{T}_{i,k}$ as in (7).
    `Diffuse` $d_{i,k}^{(0)}(s,a)$ by running consensus for L steps as per (8).
    `Update` local estimate $Q_{i,k}$ as per (11).
  end for
end for

In this section, we introduce our proposed algorithm VRDQ (outlined as Algorithm 1), which, given a prescribed failure probability $\delta \in (0,1)$, aims to generate local Q-function estimates at every agent that enjoy (due to collaboration) an error rate of $\tilde{O}(1/\sqrt{NT})$ with probability at least $1-\delta$. Here, T is the number of samples at each agent; since these samples are acquired in parallel, one can also think of T as the total run-time duration of the algorithm. The key guiding observations behind our algorithm are as follows. We note that agents need to exchange information essentially only when their Q-tables are locally updated. So, it is natural to then ask how infrequently the Q-tables can be updated, while preserving the optimal $\tilde{O}(1/\sqrt{NT})$ rate. It turns out that just $\tilde{O}(1)$ updates suffice, provided each update is made using a "low-variance" estimate of the Bellman optimality operator. Accordingly, our algorithm runs in epochs, where within an epoch, an agent *simultaneously* performs (i) *local estimation* of the Bellman optimality operator, and (ii) *diffusion* of an update direction generated in the previous epoch.

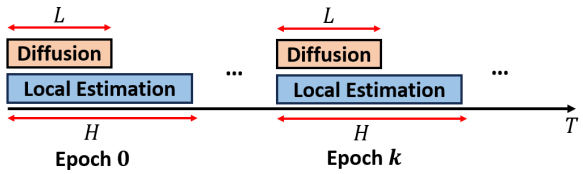

Fig. 1. Illustration of VRDQ which runs in epochs of length H. Throughout the duration of each epoch, every agent locally estimates the Bellman optimality operator using samples acquired from the generative sampling model. In parallel, agents run an average consensus protocol for the first L steps of the epoch to diffuse information.

Importantly, an update to the Q-table is made *only once* at the end of each epoch. As we shall see, this approach incurs only a logarithmic communication overhead and admits a straightforward analysis. We now describe the key ideas.

Module 1 (Local Operator Estimation). In conventional single-agent and distributed RL algorithms, agents typically construct a noisy estimate $\mathcal{T}_t$ (as defined in (4)) of the Bellman operator immediately upon observing a new sample, and subsequently update their Q-tables using the update rule in (5). Inspecting the noisy operator $\mathcal{T}_t$ in (4), we notice that it is constructed using only a single sampled next state, as opposed to the true Bellman optimality operator $\mathcal{T}^*$ in (3), which takes an expectation over all possible next states. Thus, in standard approaches, agents update their Q-values **all the time** using *high-variance* noisy operator estimates. Given this observation, we ask: *Does it suffice for each agent to update its Q-table intermittently with a refined (low-variance) operator to achieve the same performance?*

Our main innovation is to show that this is indeed possible via a variance reduction technique. We divide the T samples of each agent into K epochs of H samples each, such that $T = KH$ (assuming T is divisible by K for simplicity); here, K is a design parameter that will be specified later. At epoch $k \in \{0, \cdots, K-1\}$, agent $i \in [N]$ generates an estimate $\mathcal{T}_{i,k}$ of the Bellman optimality operator $\mathcal{T}^*$, by estimating the transition kernel $\mathcal{P}$ and the reward function $\mathcal{R}$. To do so, the agent sequentially queries the generative sampling model, which provides H independent observations for every state-action pair within each epoch. Since we assume the rewards to be non-noisy, the reward function can be directly estimated as $\mathcal{R}_{i,k}(s,a) = \mathcal{R}(s,a)$ for all (s,a) pairs, where $\mathcal{R}_{i,k}$ is the estimate of the reward function of agent i at epoch k. The kernel $\mathcal{P}$ is estimated as follows:

$$\mathcal{P}_{i,k}(s' \mid s,a) = (1/H) \sum_{u=0}^{H-1} \mathbf{1}_{i,k,u}^{s,a}(s'), \tag{6}$$

where $\mathcal{P}_{i,k}$ is the transition kernel estimate of agent i at epoch k, and $\mathbf{1}_{i,k,u}^{s,a}(s')$ is an indicator function that equals 1 if, at the u-th query to state-action pair (s,a) in epoch k, agent i observes s' as the next state. Using these estimates, agent i constructs a refined Bellman optimality operator estimate $\mathcal{T}_{i,k} : \mathbb{R}^{S \times A} \to \mathbb{R}^{S \times A}$, defined as

$$\mathcal{T}_{i,k} f(s,a) := \mathcal{R}_{i,k}(s,a) + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{i,k}(s' \mid s,a) \max_{a' \in \mathcal{A}} f(s',a'), \tag{7}$$

$\forall f \in \mathbb{R}^{S \times A}$. Intuitively, a larger epoch length H indicates a better estimate for $\mathcal{T}^*$. The choice of this hyperparameter will be detailed in the next section.

Module 2 (Diffusion). Having generated $\mathcal{T}_{i,k}$, it is then necessary to diffuse this information across the network to leverage collaborative gains. Our key observation in this regard is to note that *the time it takes to obtain an accurate estimate of the Bellman operator dominates the time taken to spread information*. As such, our **core idea** is to have the estimation and diffusion phases evolve *in parallel*, in a decoupled manner; see Fig. 1 for an illustration. Formally, in each epoch $k = 0, 1, \ldots,$ every agent i aims to diffuse the object $\mathcal{T}_{i,k-1}Q_{i,k}$, where $Q_{i,k} \in \mathbb{R}^{S\times A}$ is the Q-estimate of agent i at the beginning of epoch k, with $Q_{i,0} = 0$. We use the convention that $\mathcal{T}_{i,-1}f = 0, \forall f \in \mathbb{R}^{S\times A}$. To diffuse information, agents run a consensus protocol for only the first L steps of the H-length epoch, where L is a parameter to be specified later. The diffusion steps evolve as

$$d_{i,k}^{(\ell+1)}(s,a) = \sum_{j\in\mathcal{N}_i} (W)_{ij} d_{j,k}^{(\ell)}(s,a), \quad \forall (s,a) \in \mathcal{S}\times\mathcal{A}, \tag{8}$$

for $\ell = 0, \ldots, L-1$, where the weights $\{(W)_{ij}\}$ are entries of the mixing matrix W, and $d_{i,k}^{(\ell)} \in \mathbb{R}^{S\times A}$ denotes the diffusion model of agent i at diffusion step ℓ within epoch k. This object is initialized as

$$d_{i,k}^{(0)} = \mathcal{T}_{i,k-1}Q_{i,k}. \tag{9}$$

By convention, note that $d_{i,0}^{(0)} = 0$. Define $d_k^{(\ell)}(s,a) := \big(d_{1,k}^{(\ell)}(s,a), \ldots, d_{N,k}^{(\ell)}(s,a)\big)^\top$, for all $\ell \in \{0, \cdots, L-1\}$. Based on (8), after L rounds of diffusion in the k-th epoch, the following holds for each $(s,a) \in \mathcal{S}\times\mathcal{A}$:

$$d_k^{(L)}(s,a) = W^L d_k^{(0)}(s,a). \tag{10}$$

Finally, each agent $i \in [N]$ updates its Q-estimate as follows:

$$Q_{i,k+1} = (1-\alpha)Q_{i,k} + \alpha d_{i,k}^{(L)}, \tag{11}$$

where $\alpha \in (0,1)$ is the step-size. Intuitively, the diffused model $d_{i,k}^{(L)}$ aggregates information from all agents in the network, thereby reducing the variance of each agent's Q estimate and accelerating convergence through collaborative learning. This intuition will be made precise in the next section, where we will formally analyze VRDQ.

IV. Main Result

To state our main result, we require the following key property of doubly-stochastic matrices.

Fact 1. *Let $W \in \mathbb{R}^{N\times N}$ be doubly stochastic. Assume W is* primitive*, i.e., W is irreducible and aperiodic (it suffices that the induced graph is connected and $(W)_{ii} > 0$ for all $i \in [N]$). Then there exist constants $C_1 > 0$ and $\rho \in (0,1)$ such that, for all $i \in [N]$ and all $t \geq 0$,*

$$\left\| [W^t]_i - \tfrac{1}{N}\mathbf{1}^\top \right\| \leq C_1 \rho^t, \tag{12}$$

where $[W^t]_i$ denotes the i-th row of W^t and $\mathbf{1} \in \mathbb{R}^N$ is the all-ones vector.

Next, define the agent-wise epoch-k error as $e_{i,k} := \|Q_{i,k} - Q^*\|_\infty$. Our main result for VRDQ is then as follows.

Theorem 1. *Given any failure probability $\delta \in (0,1)$, suppose the step-size α, number of epochs K, and the diffusion period L in Algorithm 1 be chosen as follows:*

$$\begin{aligned} &\alpha = \frac{\log(NT)}{(1-\gamma)K}; \quad K = \lceil c_1 \log(NT)/(1-\gamma)\rceil; \\ &L = \lceil \log\Big(c_2 N^{3/2}\sqrt{T}\sqrt{1-\gamma}\Big)/\log(1/\rho)\rceil, \end{aligned} \tag{13}$$

where c_1, c_2 are universal constants, and $\rho \in (0,1)$ is as defined in (12). Then, after K epochs, the following bound holds with probability at least $1-\delta$:

$$e_{i,K} \leq \frac{e_{i,0}}{NT} + \mathcal{O}\left(\frac{\bar{R}\sqrt{\log(NT)\log\left(\frac{2SAT}{\delta}\right)}}{(1-\gamma)^{2.5}\sqrt{NT}} \right). \tag{14}$$

The proof of Theorem 1 is deferred to the next section. We now highlight a few takeaways from the above result.

• **Near-Optimal Rates with Collaborative Speedup.** To interpret the guarantee in (14), fix any agent $i \in [N]$. For a single learner, prior work [12], [13] has shown that Q-learning converges at the rate $\tilde{\mathcal{O}}\Big(\frac{1}{(1-\gamma)^{2.5}\sqrt{T}}\Big)$ when trained on T samples. With N agents jointly interacting with the same environment $\mathcal{M}$, the *best possible rate* is $\tilde{\mathcal{O}}\Big(\frac{1}{\sqrt{NT}}\Big)$, up to polynomial factors in $(1-\gamma)$. This matches the rate in Theorem 1, revealing that VRDQ enjoys *near-optimal guarantees* that benefit from collaboration.

• **Poly-Log Communication Overhead.** The total communication overhead (per agent) of VRDQ is KL, where K is the number of epochs, and L is the number of diffusion steps per epoch. From (13), we note that K and L are logarithmic in both the number of samples T, and the number of agents N. This marks a significant reduction in communication cost relative to prior work on distributed/federated RL, which suffer from $\mathcal{O}(T)$ dependence on the number of samples and/or $\mathcal{O}(N)$ dependence on the number of agents [9]–[11]. In summary, VRDQ *achieves near-optimal statistical efficiency while drastically reducing communication costs*, a crucial property for large-scale multi-agent RL systems.

• **Effect of Network Topology.** From (14), we note that the network topology does not directly influence the final convergence rate. However, it does influence the initial burn-in time needed for our guarantees to kick in. To see why, observe from (13) that the network structure, as captured by the parameter ρ, sets a lower bound on the diffusion period L. Moreover, we require $T \geq KL$, since we need $L \leq H = T/K$, where H is the epoch length. Thus, the network-dependent quantity ρ imposes a lower bound on the total run-time duration T.

V. Simulation Results for VRDQ

In this section, we evaluate the performance of VRDQ on a synthetic grid-world environment with 10 states, 5 actions, and discount factor $\gamma = 0.9$. All rewards belong to the interval $[0,1]$. The step size is set to $\alpha = 0.1$, time-steps to $T = 10^5$, and the failure probability to $\delta = 0.01$. The left panel of Fig. 2 illustrates the benefits of collaboration from

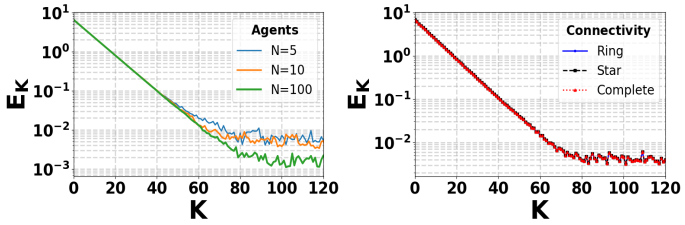

Fig. 2. Plots of the ℓ_∞ error $E_K = \sum_{i\in[N]} \|Q_{i,K} - Q^*\|_\infty / N$ for VRDQ as a function of the number of epochs K, with varying number of agents **(Left)**, and under different network topologies with $N = 100$ **(Right)**.

VRDQ, showing a lower error floor with increasing N. In the right panel, we note that when the epoch length is chosen to be large enough, the network topology does not affect the convergence rate of VRDQ, complying with theory.

VI. Q-Learning over Time-Varying Networks

In this section, we will briefly explain how our developments can be easily extended to account for a fairly general class of undirected time-varying networks. To do so, consider an undirected graph sequence $\{\mathcal{G}(0), \mathcal{G}(1), \ldots\}$, where at a given time-step t, $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t))$, with $\mathcal{E}(t)$ representing the (time-varying) edge set at time t. We say that $(i, j) \in \mathcal{E}(t)$ if and only if agents i and j can exchange information with each other at time t. Accordingly, the neighbor set of agent i at time t is defined as $\mathcal{N}_i(t) := \{j | (i, j) \in \mathcal{E}(t)\}$. Let $W(t)$ be the mixing weight matrix at time t that is consistent with $\mathcal{G}(t)$ in the sense that if $i \neq j$, and $(j, i) \notin \mathcal{E}(t)$, then $(W)_{ij}(t) = 0$, where $(W)_{ij}(t)$ is the ij-th component of the matrix $W(t)$. To proceed, for any $b = 1, 2, \ldots$, and $t \geq b-1$, let us define $W_b(t) := W(t)W(t-1)\cdots W(t-b+1)$. Following [19], we then impose the following standard assumptions on the sequence $\{W(t)\}$ of mixing matrices.

Assumption 1. *The sequence $\{W(t)\}$ satisfies the following:*

(i) $W(t)\mathbf{1} = \mathbf{1}$, *and* $\mathbf{1}^\top W(t) = \mathbf{1}^\top, \forall t \geq 0$.

(ii) *There exists a positive integer B such that*

$$\omega := \sup_{t \geq B-1} \omega(t) < 1, \text{ where } \omega(t) := \|W_B(t) - \frac{\mathbf{1}\mathbf{1}^\top}{N}\|_2,$$

and $\omega(t)$ is defined as above for all $t \geq B - 1$.

We note that the above assumptions have appeared extensively in the study of consensus algorithms; see [19] and the references therein. To account for time-varying networks, the only minor modification to VRDQ is in the diffusion step (8), which now takes the following form:

$$d_{i,k}^{(\ell+1)}(s,a) = \sum_{j \in \mathcal{N}_i(kH+\ell)} (W)_{ij}(kH+\ell) d_{j,k}^{(\ell)}(s,a), \quad (15)$$

where $\ell = 0, 1, \ldots, \bar{L}B - 1$, and $\bar{L}$ is a parameter to be specified shortly. Two points are worth noting. First, the ℓ-th diffusion step within the k-th epoch corresponds to time-step $kH + \ell$; hence, $W(kH + \ell)$ governs the diffusion process at this step. Second, the agents run diffusion for only the first $\bar{L}B$ time steps within each epoch. We then have the following result for time-varying graphs.

Theorem 2. *Suppose Assumption 1 holds, and VRDQ is run with the same choice of step-size α and epoch count K as specified in (13). Moreover, suppose that*

$$\bar{L} = \lceil \log\Big(C\, N\sqrt{T}\sqrt{1-\gamma}\Big) / \log(1/\omega) \rceil,$$

where ω is as defined in Assumption 1, and C is a suitable universal constant. Then, VRDQ provides the exact same guarantee as in (14).

The above result tells us that VRDQ continues to enjoy the same collaborative benefits as before, even for a fairly general sequence of time-varying networks. Moreover, the communication cost remains poly-logarithmic in T. To see why, we note that the communication overhead per agent is now $K\bar{L}B$, where K and $\bar{L}$ remain logarithmic in both N and T, and B has no dependence on T. That said, we note that B might very well depend on N.

VII. Analysis

In this section, we develop finite–time convergence guarantees for our proposed algorithm VRDQ. As a first step, we pass from the agent-wise recursion in (11) to the network average $\bar{Q}_k := \frac{1}{N}\sum_{i=1}^N Q_{i,k}$, as follows:

$$\bar{Q}_{k+1} = (1-\alpha)\cdot \underbrace{\frac{1}{N}\sum_{i=1}^N Q_{i,k}}_{:=\bar{Q}_k} + \alpha \cdot \underbrace{\frac{1}{N}\sum_{i=1}^N d_{i,k}^{(L)}}_{:=\bar{d}_k}. \quad (16)$$

Next, we perform a simple decomposition of (16) to obtain the following recursive relation:

$$\begin{aligned}\bar{Q}_{k+1} - Q^* &= (1-\alpha)(\bar{Q}_k - Q^*) + \alpha(\bar{d}_k - Q^*) \\ &= (1-\alpha)(\bar{Q}_k - Q^*) + \alpha(\mathcal{T}^*\bar{Q}_k - \mathcal{T}^*Q^*) + \alpha(\bar{d}_k - \mathcal{T}^*\bar{Q}_k),\end{aligned} \quad (17)$$

where we used the fixed-point property of the Bellman optimality operator, namely $\mathcal{T}^*Q^* = Q^*$ [15]. To proceed, we will use the following contractive property of the Bellman optimality operator [15]:

$$\|\mathcal{T}^* f_1 - \mathcal{T}^* f_2\|_\infty \leq \gamma \|f_1 - f_2\|_\infty, \forall f_1, f_2 \in \mathbb{R}^{S\times A}. \quad (18)$$

Taking the ℓ_∞-norm on both sides of (17), and using the contraction property in (18), we obtain the following bound on $\bar{e}_{k+1} := \|\bar{Q}_{k+1} - Q^*\|_\infty$ as follows:

$$\begin{aligned}\bar{e}_{k+1} &\leq (1-\alpha(1-\gamma))\bar{e}_k + \alpha\|\bar{d}_k - \mathcal{T}^*\bar{Q}_k\|_\infty \\ &\leq (1-\alpha(1-\gamma))\bar{e}_k + \alpha \underbrace{\Big\|\bar{d}_k - \frac{1}{N}\sum_{i=1}^N \mathcal{T}_{i,k-1}Q_{i,k}\Big\|_\infty}_{(*)} \\ &\quad + \alpha \underbrace{\Big\|\frac{1}{N}\sum_{i=1}^N \mathcal{T}_{i,k-1}Q_{i,k} - \mathcal{T}^*\bar{Q}_k\Big\|_\infty}_{(**)}.\end{aligned} \quad (19)$$

In analyzing the convergence of VRDQ, the key hurdle is to control the deviation $\|\bar{d}_k - \mathcal{T}^*\bar{Q}_k\|_\infty$. This term reflects two sources of error: the *diffusion error* $(*)$, analyzed in Lemma 2 and arising from the diffusion scheme in Algorithm 1, and the *operator estimation error* $(**)$, addressed in Lemma 3. For controlling both sources of error, we need the following lemma that provides a bound on the iterates.

Lemma 1. ***(Boundedness of Iterates)*** *The following is true for the iterates generated by Algorithm 1, $\forall k \geq 0$:*

$$|Q_{i,k}(s,a)| \leq \frac{\bar{R}}{1-\gamma}, \forall (s,a) \in \mathcal{S} \times \mathcal{A}, \forall i \in [N]. \tag{20}$$

Proof. We prove this result by induction. Since $\bar{R} > 0$ and $Q_{i,0}(s,a) = 0$ for all $(s,a) \in \mathcal{S} \times \mathcal{A}$ in Algorithm 1, the bound in (20) holds trivially at $k = 0$ for all agents. Suppose that the bound in (20) holds for *all agents* $i \in [N]$, and for *all epochs* up to some epoch $k \geq 0$. We need to show that the same bound applies to $Q_{i,k+1}(s,a), \forall i \in [N], \forall (s,a) \in \mathcal{S} \times \mathcal{A}$. To that end, fix an agent i, a state–action pair (s,a), and recall that $Q_{i,k+1}(s,a)$ is generated as per (11), where each agent i computes $d_{i,k}^{(L)}(s,a)$ according to (8) in epoch k. We proceed to bound $d_{i,k}^{(L)}(s,a)$ as follows:

$$\begin{aligned} \left|d_{i,k}^{(L)}(s,a)\right| &= \left| \sum_{j=1}^{N} (W^L)_{ij}\, d_{j,k}^{(0)}(s,a) \right| \\ &\overset{(a)}{\leq} \max_{j \in [N]} \left| d_{j,k}^{(0)}(s,a) \right| \\ &\overset{(b)}{=} \max_{j \in [N]} \left| \mathcal{T}_{j,k-1} Q_{j,k}(s,a) \right| \\ &\overset{(c)}{\leq} \bar{R} + \gamma \max_{j \in [N]} \|Q_{j,k}\|_\infty \leq \frac{\bar{R}}{1-\gamma}. \end{aligned} \tag{21}$$

Here, $(W^L)_{ij}$ denotes the (i,j)-th entry of W^L. Now, (a) holds since powers of doubly-stochastic matrices remain doubly stochastic; (b) uses the definition of $d_{j,k}^{(0)}$ in (9), and (c) follows from the definition of $\mathcal{T}_{j,k-1}$ in (7), together with the induction hypothesis. Next, from (11), we obtain

$$\begin{aligned} \left|Q_{i,k+1}(s,a)\right| &\leq (1-\alpha) \left|Q_{i,k}(s,a)\right| + \alpha \left|d_{i,k}^{(L)}(s,a)\right| \\ &\overset{(\bullet)}{\leq} (1-\alpha) \frac{\bar{R}}{1-\gamma} + \alpha \frac{\bar{R}}{1-\gamma} = \frac{\bar{R}}{1-\gamma}, \end{aligned} \tag{22}$$

where $(\bullet)$ uses the induction hypothesis and the bound in (21). The proof follows by noting that the above argument applies identically to all agents and state-action pairs. □

Our next result helps control the diffusion error.

Lemma 2. ***(Bounding Diffusion Error)*** *For all $k \in [K]$, the following bounds hold deterministically:*

$$(a) \quad \left\| \bar{d}_k - \tfrac{1}{N} \textstyle\sum_{i=1}^{N} \mathcal{T}_{i,k-1} Q_{i,k} \right\|_\infty \leq \frac{C_1 N \bar{R} \rho^L}{1-\gamma} := \Delta_1,$$

$$(b) \quad \left\| d_{i,k}^{(L)} - \bar{d}_k \right\|_\infty \leq \frac{2 C_1 N \bar{R} \rho^L}{1-\gamma},$$

where $\bar{d}_k$ is as defined in (16).

Proof. To establish item (a), start by fixing an agent $i \in [N]$, an epoch $k \in [K]$, and a state-action pair $(s,a) \in \mathcal{S} \times \mathcal{A}$. Using (9) and (10), we then have

$$\begin{aligned} |d_{i,k}^{(L)}(s,a) - \tfrac{1}{N} \sum_{i=1}^{N} \mathcal{T}_{i,k-1} Q_{i,k}(s,a)| &= \left([W^L]_i - \frac{\mathbf{1}^\top}{N} \right) d_k^{(0)}(s,a) \\ &\overset{(a)}{\leq} \left\| [W^L]_i - \frac{1}{N} \mathbf{1}^\top \right\|_\infty \|d_k^{(0)}(s,a)\|_1 \\ &\overset{(b)}{\leq} \frac{C_1 N \bar{R} \rho^L}{1-\gamma} := \Delta_1, \end{aligned} \tag{23}$$

where for (a), we used Holder's inequality, and for (b), we used (12) and $\max_{j \in [N]} \left| \mathcal{T}_{j,k-1} Q_{j,k}(s,a) \right| \leq \bar{R}/(1-\gamma)$ (from the analysis in Lemma 1). Since the analysis in (23) applies to all state-action pairs, we conclude that

$$\left\| d_{i,k}^{(L)} - \tfrac{1}{N} \sum_{i=1}^{N} \mathcal{T}_{i,k-1} Q_{i,k} \right\|_\infty \leq \frac{C_1 N \bar{R} \rho^L}{1-\gamma} := \Delta_1. \tag{24}$$

Noting that the above inequality applies identically to all agents establishes item (a) in the statement of the lemma. For item (b), we simply apply the triangle inequality, and combine item (a) with (24). □

The following result controls the estimation error.

Lemma 3. ***(Bounding Estimation Error)*** *With probability at least $1-\delta$, the following bound holds for all $k \in [K]$:*

$$\underbrace{\left\| \frac{1}{N} \sum_{i=1}^{N} \mathcal{T}_{i,k-1} Q_{i,k} - \mathcal{T}^* \bar{Q}_k \right\|_\infty}_{(*)} \leq \Delta_2, \text{ where} \tag{25}$$

$$\Delta_2 := \mathcal{O}\left(\frac{\bar{R} \sqrt{\log(2SAT/\delta)}}{(1-\gamma)\sqrt{NH}} \right) + \mathcal{O}\left(\frac{N \bar{R} \rho^L}{1-\gamma} \right). \tag{26}$$

Proof. To facilitate our analysis, we decompose the estimation error term defined in (25), denoted by $(*)$, as $(*) \leq \mathcal{A}_1 + \mathcal{A}_2$, where

$$\begin{aligned} \mathcal{A}_1 &:= \left\| \frac{1}{N} \sum_{i=1}^{N} \left[\mathcal{T}_{i,k-1} Q_{i,k} - \mathcal{T}^* Q_{i,k} \right] \right\|_\infty, \\ \mathcal{A}_2 &:= \left\| \frac{1}{N} \sum_{i=1}^{N} \left[\mathcal{T}^* Q_{i,k} - \mathcal{T}^* \bar{Q}_k \right] \right\|_\infty. \end{aligned} \tag{27}$$

The first term, $\mathcal{A}_1$, captures the statistical error between the empirical Bellman operator in (7) and the true Bellman optimality operator in (3). The second term, $\mathcal{A}_2$, arises from the "consensus gap" between $Q_{i,k}$ and $\bar{Q}_k$.

Bounding $\mathcal{A}_1$. To bound $\mathcal{A}_1$, we will leverage concentration inequalities for sums of sub-Gaussian random variables.[1] To see how this can be done, fix a particular state–action pair $(s,a) \in \mathcal{S} \times \mathcal{A}$. First, we decompose (7) by invoking the empirical probability distribution in (6):

$$\begin{aligned} \mathcal{T}_{i,k-1} Q_{i,k}(s,a) &= \mathcal{R}(s,a) + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{i,k-1}(s' \mid s,a) \max_{a' \in \mathcal{A}} Q_{i,k}(s',a') \\ &= \frac{1}{H} \sum_{u=0}^{H-1} \underbrace{\left(\mathcal{R}(s,a) + \gamma \sum_{s' \in \mathcal{S}} \mathbf{1}_{i,k-1,u}^{s,a}(s') \max_{a' \in \mathcal{A}} Q_{i,k}(s',a') \right)}_{X_u}. \end{aligned} \tag{28}$$

Now observe that

$$\begin{aligned} &\mathbb{E}\left[X_u \mid \{Q_{i,k}\}_{i \in [N]} \right] \\ &= \mathcal{R}(s,a) + \gamma \sum_{s' \in \mathcal{S}} \mathbb{E}\left[\mathbf{1}_{i,k-1,u}^{s,a}(s') \mid \{Q_{i,k}\}_{i \in [N]} \right] \max_{a' \in \mathcal{A}} Q_{i,k}(s',a') \\ &\overset{(\bullet)}{=} \mathcal{R}(s,a) + \gamma \mathbb{E}_{s' \sim \mathcal{P}(\cdot \mid s,a)} \max_{a' \in \mathcal{A}} Q_{i,k}(s',a') \\ &= \mathcal{T}^* Q_{i,k}(s,a). \end{aligned} \tag{29}$$

[1] A random variable $X \in \mathbb{R}$ is said to be sub-Gaussian with variance proxy σ^2 (or σ-sub-Gaussian) if its moment-generating function satisfies $\mathbb{E}[\exp(sX)] \leq \exp(s^2\sigma^2/2), \forall s \in \mathbb{R}$ [20].

As per (11), the iterates $\{Q_{i,k}\}_{i\in[N]}$ depend only on the randomness realized up to the end of epoch $k-2$, and are therefore independent of the randomness in epoch $k-1$ under the i.i.d. synchronous sampling model. As a result, in $(\bullet)$, we used $\mathbb{E}\Big[\mathbf{1}^{s,a}_{i,k-1,u}(s') \,\Big|\, \{Q_{i,k}\}_{i\in[N]}\Big] = \mathbb{E}\Big[\mathbf{1}^{s,a}_{i,k-1,u}(s')\Big] = \mathcal{P}(s' \mid s,a)$. Consequently, conditioned on $\{Q_{i,k}\}_{i\in[N]}$, each random variable $\{X_u - \mathcal{T}^*Q_{i,k}(s,a)\}$ is zero-mean for all $u \in [H]$. Moreover, under the i.i.d. synchronous sampling model, the samples $\{X_u\}_{u=0}^{H-1}$ generated within epoch $k-1$ are independent. Hence, the collection $\{X_u - \mathcal{T}^*Q_{i,k}(s,a)\}_{u=1}^{H}$ forms an i.i.d. sequence of zero-mean random variables, conditional on $\{Q_{i,k}\}_{i\in[N]}$. Furthermore, by invoking the boundedness of the Bellman optimality operator in (3), agent-wise Bellman update in (7), and local Q-updates from Lemma 2, we obtain

$$|X_u - \mathcal{T}^*Q_{i,k}(s,a)| \le |X_u| + |\mathcal{T}^*Q_{i,k}| \le \frac{2\bar{R}}{1-\gamma} := \Gamma. \tag{30}$$

By the conditional zero-mean property established in (29), together with the boundedness property in (30), we note that the sequence $\{X_u - \mathcal{T}^*Q_{i,k}(s,a)\}$ forms an i.i.d. collection of Γ-sub-Gaussian random variables [21, Example 5.6]. It also follows from (28) that the sequence $\{\mathcal{T}_{i,k-1}Q_{i,k}(s,a) - \mathcal{T}^*Q_{i,k}(s,a)\}$ is itself sub-Gaussian with variance proxy Γ^2/H [20, Lemma 1.8]. Conditional on $\{Q_{i,k}\}_{i\in[N]}$, the sole source of randomness in $\mathcal{T}_{i,k-1}Q_{i,k}(s,a)$ arises from the state transitions within the epoch, which are independent across agents under our sampling model, Thus, $\{\mathcal{T}_{i,k-1}Q_{i,k}(s,a) - \mathcal{T}^*Q_{i,k}(s,a)\}_{i=1}^{N}$ are i.i.d. $\Gamma/\sqrt{H}$-sub-Gaussian random variables [20, Lemma 1.8]. Hence, averaging across agents further reduces the variance proxy by a factor of N. As a result, conditioned on $\{Q_{i,k}\}_{i\in[N]}$, the following event holds with probability at least $1-\bar{\delta}$:

$$\mathcal{E} := \left\{ \left| \frac{1}{N}\sum_{i=1}^{N} \Big[\mathcal{T}_{i,k-1}Q_{i,k}(s,a) - \mathcal{T}^*Q_{i,k}(s,a)\Big] \right| \le \frac{\mathtt{C}_0\Gamma}{\sqrt{H}}\sqrt{\frac{\log(2/\bar{\delta})}{N}} \right\},$$

where $\mathtt{C}_0$ is some universal constant. Let $\mathbf{1}_{\mathcal{E}}$ be the indicator of the event $\mathcal{E}$. We then have

$$\mathbb{P}(\mathcal{E}) = \mathbb{E}[\mathbf{1}_{\mathcal{E}}] = \mathbb{E}\big[\mathbb{E}[\mathbf{1}_{\mathcal{E}} \mid \{Q_{i,k}\}_{i\in[N]}]\big] \;\ge\; 1-\bar{\delta},$$

where the last inequality follows from $\mathbb{P}(\mathcal{E} \mid \{Q_{i,k}\}_{i\in[N]}) \ge 1-\bar{\delta}$. By taking a union bound over all state–action pairs $(s,a) \in \mathcal{S}\times\mathcal{A}$ and epochs $k \in [K]$, with $K \le T$, the following bound is guaranteed to hold simultaneously for all (s,a) and k with probability at least $1-\delta$:

$$\left| \frac{1}{N}\sum_{i=1}^{N} \Big[\mathcal{T}_{i,k-1}Q_{i,k}(s,a) - \mathcal{T}^*Q_{i,k}(s,a)\Big] \right| \le \frac{\mathtt{C}_0\Gamma}{\sqrt{H}}\sqrt{\frac{\log(2SAT/\delta)}{N}}. \tag{31}$$

The definition of the ∞-norm implies the exact same bound as above on $\mathcal{A}_1$.

Bounding $\mathcal{A}_2$. To bound $\mathcal{A}_2$, we start by controlling the error $\epsilon_{i,k} := Q_{i,k} - \bar{Q}_k$ by subtracting (16) from (11):

$$Q_{i,k+1} - \bar{Q}_{k+1} = (1-\alpha)\big(Q_{i,k} - \bar{Q}_k\big) + \alpha\big(d^{(L)}_{i,k} - \bar{d}_k\big). \tag{32}$$

Unrolling (32) over k epochs yields the following:

$$\epsilon_{i,k} = (1-\alpha)^k\epsilon_{i,0} + \alpha\sum_{\tau=0}^{k-1}(1-\alpha)^{k-1-\tau}\big(d^{(L)}_{i,\tau} - \bar{d}_\tau\big). \tag{33}$$

As a consequence of the initialization of $Q_{i,0}$ in Algorithm 1, we have $\epsilon_{i,0} = 0$ for all $i \in [N]$. It follows that

$$\begin{aligned}\|\epsilon_{i,k}\|_\infty &= \|\alpha\sum_{\tau=0}^{k-1}(1-\alpha)^{k-1-\tau}\left(d^{(L)}_{i,\tau} - \bar{d}_\tau\right)\|_\infty \\ &\le \alpha\sum_{\tau=0}^{k-1}(1-\alpha)^{k-1-\tau}\|d^{(L)}_{i,\tau} - \bar{d}_\tau\|_\infty \\ &\overset{(\blacklozenge)}{\le} \alpha\sum_{\ell=0}^{\infty}(1-\alpha)^{\ell}\cdot\frac{\mathtt{C}_1 N\bar{R}\rho^L}{1-\gamma} = \frac{\mathtt{C}_1 N\bar{R}\rho^L}{1-\gamma},\end{aligned} \tag{34}$$

where for $(\blacklozenge)$, we applied (b) in Lemma 2. The claim of Lemma 3 follows by combining the individual bounds on $\mathcal{A}_1$ and $\mathcal{A}_2$ that we derived above. □

We are now ready to prove our main result, Theorem 1.

Proof of Theorem 1. Lemmas 2 and 3 inform us that there exists a *good event* $\bar{\mathcal{E}}$ of measure at least $1-\delta$, on which, $\|\bar{d}_k - \mathcal{T}^*\bar{Q}_k\|_\infty \le \Delta := \Delta_1 + \Delta_2, \forall k = 0,1,\ldots,K-1$, where Δ_1 is as in item (a) of Lemma 2, and Δ_2 is as in (26). On this event, unrolling (19) yields:

$$\bar{e}_K \le \underbrace{(1-\alpha(1-\gamma))^{K-1}\bar{e}_1}_{(*)} + \underbrace{\sum_{k=1}^{K-1}\alpha(1-\alpha(1-\gamma))^{K-1-k}\Delta}_{(**)}. \tag{35}$$

The term $(**)$ can be further bounded as

$$(**) \le \alpha\Delta\sum_{p=0}^{\infty}(1-\alpha(1-\gamma))^p = \frac{\Delta}{1-\gamma}.$$

Plugging the above bound in (35), on the event $\bar{\mathcal{E}}$, we have

$$\bar{e}_K \le (1-\alpha(1-\gamma))^{K-1}\bar{e}_0 + \frac{\Delta}{(1-\gamma)}. \tag{36}$$

In the above step, we used the fact that since $d^{(0)}_{i,0}(s,a)$ is initialized as zero, $Q_{i,1}$'s will also be zero-vectors $\forall i \in [N]$, implying $\bar{e}_1 = \bar{e}_0$. To arrive at the final bound in Theorem 1, set α according to (13). Under this choice, we have

$$(*) = (1-\alpha(1-\gamma))^{K-1}\bar{e}_0 \le \exp\left(-\alpha(1-\gamma)(K-1)\right)\bar{e}_0 = \bar{e}_0/(NT),$$

where we used $(1-x) \le \exp(-x)$, for $x \in (0,1)$. Using the expressions for Δ_1 and Δ_2, we then obtain

$$\bar{e}_K \le \frac{\bar{e}_0}{NT} + \tilde{\mathcal{O}}\left(\frac{\bar{R}}{(1-\gamma)^{\frac{5}{2}}\sqrt{NH}}\right) + \mathcal{O}\left(\frac{N\bar{R}\rho^L}{(1-\gamma)^2}\right). \tag{37}$$

Substituting $H = T/K$ and the choices of K and L from (13) into (37) yields the following bound, which holds with probability at least $1-\delta$:

$$\bar{e}_K \le \frac{\bar{e}_0}{NT} + \mathcal{O}\left(\frac{\bar{R}\sqrt{\log(NT)\log\left(\frac{2SAT}{\delta}\right)}}{(1-\gamma)^{5/2}\sqrt{NT}}\right). \tag{38}$$

Finally, we bound the agent-wise sub-optimality error as $e_{i,K} \le \bar{e}_K + \|\epsilon_{i,K}\|_\infty$. Using the bound on $\|\epsilon_{i,k}\|_\infty$ in (34), and the choice of L in (13), it is easy to verify that the bound on $\bar{e}_K$ in (38) also applies to $e_{i,k}$ (up to universal constants). This completes our proof of Theorem 1. □

We now provide the proof for Theorem 2.

Proof of Theorem 2. Since the proof of Theorem 2 shares the same structure as that of Theorem 1, we only elaborate on the key distinction that arises in controlling the diffusion error. To that end, fix an epoch k, a state-action pair (s,a), and notice from (15) that

$$d_k^{(\bar{L}B)}(s,a) = W_B(k;\bar{L})\cdots W_B(k;1)d_k^{(0)}(s,a), \tag{39}$$

where we define $W_B(k;\ell) := W_B(kH+\ell B-1), \ell = 1,\ldots,\bar{L}$. We claim that the following is true $\forall \ell \in [\bar{L}]$:

$$z_k^{(\ell B)}(s,a) = \tilde{W}_B(k;\ell)\cdots\tilde{W}_B(k;1)d_k^{(0)}(s,a), \tag{40}$$

where $z_k^{(\ell B)}(s,a) := d_k^{(\ell B)}(s,a) - \frac{\mathbf{1}\mathbf{1}^\top}{N}d_k^{(0)}(s,a)$, and $\tilde{W}_B(k;\ell) := W_B(kH+\ell B-1) - \mathbf{1}\mathbf{1}^\top/N$. Assuming this claim to be true for now, let us complete the rest of the analysis. Taking the 2-norm on both sides of (40) with ℓ set to $\bar{L}$, and using item (ii) in Assumption 1, we obtain

$$\|z_k^{(\bar{L}B)}(s,a)\|_\infty \le \|z_k^{(\bar{L}B)}(s,a)\|_2 \le \omega^{\bar{L}}\|d_k^{(0)}(s,a)\|_2. \tag{41}$$

Owing to the double-stochasticity of the sequence $\{W(k)\}$, the exact same bound on the iterates as derived in (21) continues to apply; thus, combined with (41), it is easy to then verify that $\|z_k^{(\bar{L}B)}(s,a)\|_\infty \le G$, where $G = \mathcal{O}\left(\omega^{\bar{L}}\sqrt{N}\bar{R}/(1-\gamma)\right)$. Recalling that $d_{j,k}^{(0)}(s,a) = \mathcal{T}_{j,k-1}Q_{j,k}(s,a)$, and noting that the above argument applies identically to every state-action pair, we conclude that

$$\|d_{i,k}^{(\bar{L}B)} - \frac{1}{N}\sum_{j=1}^N \mathcal{T}_{j,k-1}Q_{j,k}\|_\infty \le G.$$

From the above display, one can bound each of the objects in items (a) and (b) of Lemma 2 by $\mathcal{O}(G)$. The rest of the analysis is identical to that of Theorem 1. We now justify (40) by inducting on ℓ. The base case with $\ell = 1$ follows directly from the relation $d_k^{(B)}(s,a) = W_B(k;1)d_k^{(0)}(s,a)$ by adding and subtracting $\mathbf{1}\mathbf{1}^\top/N$ to $W_B(k;1)$. Now suppose the claim in (40) holds for all $\ell = 1,2,\ldots,p-1$, where $p = 2,\ldots,\bar{L}$. To extend the claim to $\ell = p$, starting from $d_k^{(pB)}(s,a) = W_B(k;p)d_k^{((p-1)B)}(s,a)$, observe that

$$\begin{aligned} z_k^{(pB)}(s,a) &= W_B(k;p)d_k^{((p-1)B)}(s,a) - \frac{\mathbf{1}\mathbf{1}^\top}{N}d_k^{(0)}(s,a) \\ &\overset{(a)}{=} W_B(k;p)z_k^{((p-1)B)}(s,a) \\ &\overset{(b)}{=} \tilde{W}_B(k;p)\cdots\tilde{W}_B(k;1)d_k^{(0)}(s,a) + \Psi, \end{aligned} \tag{42}$$

where $\Psi = (\mathbf{1}\mathbf{1}^\top/N)\tilde{W}_B(k;p-1)\cdots\tilde{W}_B(k;1)d_k^{(0)}(s,a)$. In the above steps, (a) follows by noting that $W_B(k;p)$ is also doubly-stochastic in light of double-stochasticity of $\{W(k)\}$, and (b) uses the induction hypothesis. To complete the proof, we need to argue that $\Psi = 0$. For this, noting that $p \ge 2$, observe: $\frac{\mathbf{1}\mathbf{1}^\top}{N}\tilde{W}_B(k;p-1) = \frac{\mathbf{1}\mathbf{1}^\top}{N}\left(W_B(k;p-1) - \frac{\mathbf{1}\mathbf{1}^\top}{N}\right) = \left(\frac{\mathbf{1}\mathbf{1}^\top}{N} - \frac{\mathbf{1}\mathbf{1}^\top}{N}\right) = 0$, where we used $\mathbf{1}^\top W_B(k;p-1) = \mathbf{1}^\top$. This establishes $\Psi = 0$, thereby completing the proof. □

VIII. Conclusion

We introduced a novel approach for distributed Q-learning over static and time-varying networks, and showed that collaborative speedups in sample-complexity can be achieved with just a logarithmic communication overhead. In future work, we plan to extend our approach to account for function approximation and Markov sampling.

References

[1] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, pp. 279–292, 1992.

[2] S. Kar, J. M. F. Moura, and H. V. Poor, "$\mathcal{QD}$-learning: A collaborative distributed strategy for multi-agent reinforcement learning through consensus+innovations," *IEEE Transactions on Signal Processing*, vol. 61, no. 7, pp. 1848–1862, 2013.

[3] K. Zhang, Z. Yang, H. Liu, T. Zhang, and T. Basar, "Fully decentralized multi-agent reinforcement learning with networked agents," in *International conference on machine learning*. PMLR, 2018, pp. 5872–5881.

[4] Y. Zhang and M. M. Zavlanos, "Distributed off-policy actor-critic reinforcement learning with policy consensus," in *2019 IEEE 58th Conference on decision and control (CDC)*. IEEE, 2019, pp. 4674–4679.

[5] T. Doan, S. Maguluri, and J. Romberg, "Finite-time analysis of distributed TD (0) with linear function approximation on multi-agent reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2019, pp. 1626–1635.

[6] P. Heredia, H. Ghadialy, and S. Mou, "Finite-sample analysis of distributed q-learning for multi-agent networks," in *2020 American Control Conference (ACC)*. IEEE, 2020, pp. 3511–3516.

[7] H.-D. Lim and D. Lee, "A finite-time analysis of distributed Q-Learning," *Reinforcement Learning Journal*, 2025.

[8] S. Zeng, T. T. Doan, and J. Romberg, "Finite-time convergence rates of decentralized stochastic approximation with applications in multi-agent and multi-task learning," *IEEE Transactions on Automatic Control*, vol. 68, no. 5, pp. 2758–2773, 2022.

[9] S. Khodadadian, P. Sharma, G. Joshi, and S. T. Maguluri, "Federated reinforcement learning: Linear speedup under Markovian sampling," in *ICML*. PMLR, 2022, pp. 10 997–11 057.

[10] J. Woo, G. Joshi, and Y. Chi, "The blessing of heterogeneity in federated Q-learning: Linear speedup and beyond," in *International Conference on Machine Learning*. PMLR, 2023, pp. 37 157–37 216.

[11] H. Wang, A. Mitra, H. Hassani, G. J. Pappas, and J. Anderson, "Federated temporal difference learning with linear function approximation under environmental heterogeneity," *arXiv:2302.02212*, 2023.

[12] M. J. Wainwright, "Stochastic approximation with cone-contractive operators: Sharp ℓ_∞-bounds for Q-learning," *arXiv preprint arXiv:1905.06265*, 2019.

[13] A. W. Guannan Qu, "Finite-time analysis of asynchronous stochastic approximation and Q-learning," in *Proceedings of Machine Learning Research*, vol. 125. Kluwer Academic Publisher, 2020, pp. 1–21.

[14] G. Li, C. Cai, Y. Chen, Y. Wei, and Y. Chi, "Is Q-learning minimax optimal? a tight sample complexity analysis," *Operations Research*, vol. 72, no. 1, pp. 222–236, 2024.

[15] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[16] M. Kearns and S. Singh, "Finite-sample convergence rates for Q-learning and indirect algorithms," *Advances in neural information processing systems*, vol. 11, 1998.

[17] A. Sidford, M. Wang, X. Wu, L. Yang, and Y. Ye, "Near-optimal time and sample complexities for solving Markov decision processes with a generative model," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[18] J. N. Tsitsiklis, "Asynchronous stochastic approximation and Q-learning," *Machine learning*, vol. 16, pp. 185–202, 1994.

[19] A. Nedic, A. Olshevsky, and W. Shi, "Achieving geometric convergence for distributed optimization over time-varying graphs," *SIAM Journal on Optimization*, vol. 27, no. 4, pp. 2597–2633, 2017.

[20] P. Rigollet and J.-C. Hütter, "High-dimensional statistics," *ArXiv preprint ArXiv:2310.19244*, 2023.

[21] T. Lattimore and C. Szepesvári, *Bandit Algorithms*. Cambridge University Press, 2020.