Convergence Rates for Localized Actor-Critic in Networked Markov Potential Games

Zhaoyi Zhou[1] **Zaiwei Chen**[2] **Yiheng Lin**[2] **Adam Wierman**[2]

[1]Institute for Interdisciplinary Information Sciences, Tsinghua University
[2]The Computing + Mathematical Sciences (CMS) Department, California Institute of Technology

Abstract

We introduce a class of networked Markov potential games where agents are associated with nodes in a network. Each agent has its own local potential function, and the reward of each agent depends only on the states and actions of agents within a neighborhood. In this context, we propose a localized actor-critic algorithm. The algorithm is scalable since each agent uses only local information and does not need access to the global state. Further, the algorithm overcomes the curse of dimensionality through the use of function approximation. Our main results provide finite-sample guarantees up to a localization error and a function approximation error. Specifically, we achieve an $\tilde{\mathcal{O}}(\tilde{\epsilon}^{-4})$ sample complexity measured by the averaged Nash regret. This is the first finite-sample bound for multi-agent competitive games that does not depend on the number of agents.

1 INTRODUCTION

Large-scale systems where agents interact competitively with each other have received significant attention recently, motivated by applications in power systems [Shi et al., 2022], EV charging [Lee et al., 2022], and board games [Silver et al., 2017], etc. Controlling such systems can be challenging due to the scale of the system, uncertainty about the model, communication constraints, and the interaction between agents. Inspired by the recent success of reinforcement learning (RL), there is an increasing interest in applying RL methods to environments with multi-agent interactions. However, in multi-agent RL (MARL), the analysis of the system behavior becomes challenging due to the time-varying nature of the environment faced by each agent, which results from the (time-varying) competitive decisions of other agents. As a result, the theoretical analysis of MARL, especially in the competitive setting, is still limited especially when it comes to large-scale systems.

Results on MARL in competitive settings to this point have tended to focus on games with a small number of players, e.g., 2-player zero-sum stochastic games [Littman, 1994], or games with special structure, e.g., Markov potential games (MPGs) [Fox et al., 2022]. MPGs in particular provide a setting in which the challenges of large-scale systems can be studied. The intuition behind an MPG parallels that of classical (one-shot) potential games. Specifically, the existence of a potential function guarantees that agents can converge to a global equilibrium even when using greedy localized updates. MPGs have wide-ranging applications including variants of congestion games [Leonardos et al., 2022, Fox et al., 2022], medium access control [Macua et al., 2018], and the stochastic lake game [Dechert and O'Donnell, 2006]. However, existing theoretical results for MPGs rely on the assumption that a centralized global state exists and can be observed by each individual agent. Such an assumption rules out applications in many large-scale systems including transportation networks [Zhang and Pavone, 2016] and social networks [Chakrabarti et al., 2008], where the global state space can be exponentially large in the number of agents and/or each agent can only observe its own local state.

A promising approach for the design of scalable and local MARL algorithms in competitive settings is to exploit the networked structure of practical applications to design algorithms with sample complexity that only depends on *local* properties of the network instead of the *global* state. This approach has recently been successful in the case of cooperative MARL. For example, Qu et al. [2020], Lin et al. [2021], Zhang et al. [2022c] provides a scalable localized algorithm with a sample complexity that does not depend on the number of agents. However, to this point, local algorithms that exploit network structure do not exist in the competitive MARL setting. Thus, we ask: *Can we design a scalable and local algorithm with finite-time bounds for networked MARL with competitive agents?*

Proceedings of the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023), PMLR 216:2563–2573.

1.1 MAIN CONTRIBUTIONS

We address the question above by introducing a class of networked Markov potential games (NMPGs) as the networked counterpart of classical MPGs. Importantly, NMPGs represent a broader class of games than MPGs, and draw focus to algorithm design that uses only local information.

We design a localized actor-critic algorithm that is a combination of independent policy gradient and localized TD(λ) with linear function approximation. Notably, our algorithm is *model-free*, uses only *local information*, and successfully incorporates *function approximation*. This avoids both the need for communication of the global state and the so-called "curse of dimensionality" in MARL.

Our main results provide a finite-sample bound on the averaged Nash regret for our proposed algorithm, which implies an $\tilde{\mathcal{O}}(\tilde{\epsilon}^{-4})$ sample complexity (where $\tilde{\epsilon}$ is the accuracy) up to an approximation error of using local information and a function approximation error. To our knowledge, we are the first to develop a localized algorithm in competitive MARL settings with provable performance guarantees that do not depend on the number of agents.

Our results are enabled by a novel analysis of the critic in our localized actor-critic framework. In particular, we propose a localized cost evaluation problem, a new MARL setting to investigate the performance of a local algorithm under a fixed policy. As a critical part of the proof, we propose a novel concept called a "sub-chain" that connects local algorithms to their global counterparts, enabling performance bounds via bounds on the gap between the two.

1.2 RELATED WORK

Markov Potential Games. Our work adds to the literature on MPGs in MARL. Analytic results for non-cooperative MARL are challenging to obtain because agents learn in a non-stationary environment as other agents update their policies. As a result, existing analysis has focused on special cases like 2-player stochastic games [Littman, 1994], adversarial team Markov games [Kalogiannis et al., 2022], and MPGs [Fox et al., 2022]. The case of MPGs has received considerable attention recently because the potential games are broadly applicable [Leonardos et al., 2022] and the existence of potential functions enables provable guarantees [Zhang et al., 2022b, Ding et al., 2022, Fox et al., 2022, Zhang et al., 2022a]. While these papers provide algorithms with provable convergence guarantees, they assume that all agents share a common global state and can observe the global state to decide local actions. An important open question is understanding how to learn in settings where global information is not available. Our work studies the MARL setting where each agent has its own local state and can only decide local actions based on the local states.

MARL in Networked Systems. The Markov decision process (MDP) model we study is inspired by a series of works on Networked MARL [Qu et al., 2020, Lin et al., 2021, Zhang et al., 2022c], where RL agents are located on a network. In such models, the local state transition of an agent is affected by its own local state/action and its direct neighbors' local states. Networked MARL is applicable to a wide range of applications, including communication networks [Vogels et al., 2003], social networks [Chakrabarti et al., 2008], and traffic networks [Zhang and Pavone, 2016]. Compared with general MARL, the additional structure of networked MARL enables us to establish a critical exponential decay property on the local Q-functions, which leads to the design of localized actor-critic algorithms [Qu et al., 2020, Lin et al., 2021]. All prior works on networked MARL study the case when agents cooperatively maximize the sum of all local rewards. In contrast, our work studies a non-cooperative NMPG in which each agent has its own objective.

Another approach to study MARL problems is to use mean-field control (MFC) [Gu et al., 2021a, Mondal et al., 2022a,b]. The major difference between the mean-field setting and our setting is that mean-field MARL focuses on homogeneous agents, while we allow each agent to have different transition probabilities and local policies.

Finite-Sample Analysis of TD-Learning Variants. TD-learning and its variants are widely used for policy evaluation in RL, which plays a critical role in most policy-space algorithms. The asymptotic analysis of TD-learning dates back to Tsitsiklis [1994], Jaakkola et al. [1994], while finite-sample convergence bounds have received attention in the last decade. In TD-learning, function approximation is a useful technique to reduce the dimension of learning parameters at the cost of incurring an approximation error that depends on the function class. Recently, many breakthroughs are made on finite-sample error bounds for TD-learning with function approximation [Bhandari et al., 2018, Srikant and Ying, 2019, Dalal et al., 2018, Yu and Bertsekas, 2009]. Meanwhile, in multi-agent settings, localized TD-learning is crucial for limiting communication and the need for global information [Lin et al., 2021]. Our work provides a novel finite-sample error bound for localized TD-learning with function approximation.

2 PROBLEM DESCRIPTION

Network Structure. We study MARL in the context of networked multi-agent Markov games. Specifically, we consider a setting with n agents that are associated with an undirected graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N} = \{1, 2, \ldots, n\}$ is the set of nodes and $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$ is the set of edges. We denote by $\text{dist}(i, j)$ the graph distance between agents i and j. The local state space and local action space of agent i are denoted by $\mathcal{S}_i$ and $\mathcal{A}_i$, respectively, which are both finite sets. The

global state is denoted as $s = (s_1, \ldots, s_n) \in \mathcal{S} := \prod_{i=1}^n \mathcal{S}_i$ and the global action is defined similarly. For any subset $I \subseteq \mathcal{N}$, we use s_I to denote the joint state of the agents in I and use $\mathcal{S}_I := \prod_{i\in I} \mathcal{S}_i$ to denote the joint state space of agents in I. Similarly, we define a_I and $\mathcal{A}_I$ as the joint action and joint action space of the agents in I. Denote $\mu \in \Delta(\mathcal{S})$ as the initial state distribution, where $\Delta(\mathcal{S})$ denotes the $|\mathcal{S}|$-dimensional probability simplex.

Transition Probabilities. At time $t \geq 0$, given current state $s(t)$ and action $a(t)$, for each agent $i \in \mathcal{N}$, its successor state $s_i(t+1)$ is independently generated according to the following transition probability, which is only dependent on its neighbors' states and its own action:

$$\mathcal{P}(s(t+1) \,|\, s(t), a(t)) = \prod_{i=1}^{n} \mathcal{P}_i(s_i(t+1) \,|\, s_{\mathcal{N}_i}(t), a_i(t)),$$

where $\mathcal{N}_i = \{i\} \cup \{j \in \mathcal{N} \mid (i,j) \in \mathcal{E}\}$ denotes the neighborhood of i, including i itself. In addition, given an arbitrary integer $\kappa \geq 0$, we use N_i^κ to denote the κ-hop neighborhood of i, i.e., $N_i^\kappa = \{i\} \cup \{j \in \mathcal{N} \mid \text{dist}(i,j) \leq \kappa\}$, and use $-N_i^\kappa = \mathcal{N}/N_i^\kappa$ to denote the set of agents that are not in N_i^κ. We use $U_i^\kappa = N_i^\kappa/\{i\}$ to denote the agents in the κ-hop neighborhood of i, excluding i itself.

Remark 2.1. We require that each agent's transition probability depends only on the states of its neighbors and its own action, which is common in networked MARL literature [Qu et al., 2020, Zhang et al., 2022c]. Intuitively, it implies that the impact from far-away agents on the network is "negligible", which eventually leads to the exponential decay property (cf. Lemma 4.1).

Reward Function. Each agent $i \in \mathcal{N}$ is associated with a deterministic reward function $r_i : \mathcal{S} \times \mathcal{A} \mapsto [0,1]$. The interval $[0,1]$ is chosen without loss of generality over the set of bounded reward functions. In general, agent i's reward depends on the global state and the global action. Due to the network structure, we assume that there exists a non-negative integer κ_r such that the reward function of each agent depends only on the states and the actions of other agents within its κ_r-hop neighborhood, i.e., $r_i(s,a) = r_i(s_{\mathcal{N}_i^{\kappa_r}}, a_{\mathcal{N}_i^{\kappa_r}})$ for all i. This makes intuitive sense as we expect the dependence between two agents to weaken as their graph distance grows.

Policy. In this work, we consider stationary policies [Zhang et al., 2021]. Specifically, each agent $i \in \mathcal{N}$ is associated with a localized policy $\xi_i : \mathcal{S}_i \mapsto \Delta(\mathcal{A}_i)$. Given a subset $I \subseteq \mathcal{N}$, we define $\xi_I : \mathcal{S}_I \mapsto \Delta(\mathcal{A}_I)$ as the joint policy of agents in I. Note that $\xi_I(a_I \mid s_I) = \prod_{i\in I} \xi_i(a_i \mid s_i)$. We use Ξ_i to denote agent i's local policy space, and Ξ_I to denote the joint policy space of agents in I. When $I = \mathcal{N}$, we omit the subscript and just write ξ for $\xi_\mathcal{N}$ (and Ξ for $\Xi_\mathcal{N}$). Throughout, we also use $\xi = (\xi_1, \xi_2, \cdots, \xi_n)$ to highlight the local policy components. In this work, we will frequently work with softmax policies, which are defined as

$$\xi_i^{\theta_i}(a_i|s_i) = \frac{\exp(\theta_{i,s_i,a_i})}{\sum_{a_i' \in \mathcal{A}_i} \exp(\theta_{i,s_i,a_i'})}, \ \forall\, i, s_i, a_i, \quad (1)$$

where $\xi_i^{\theta_i}$ stands for agent i's local policy parametrized by the weight vector $\theta_i \in \mathbb{R}^{|\mathcal{S}_i||\mathcal{A}_i|}$. We denote $\theta = (\theta_1, \theta_2, \cdots, \theta_n)$ as the parameter of a global policy ξ^θ.

Value Function. Given a global policy ξ and an agent i, we define agent i's Q-function $Q_i^\xi \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ as

$$Q_i^\xi(s,a) = \sum_{t=0}^{\infty} \gamma^t \mathbb{E}_\xi \left[r_i(s(t), a(t)) \,|\, s(0) = s, a(0) = a \right]$$

for all (s,a), where $\gamma \in (0,1)$ is the discount factor, and $\mathbb{E}_\xi[\,\cdot\,]$ is taken w.r.t. the randomness in the (stochastic) policy ξ and the transition probabilities. With Q_i^ξ defined above, the averaged Q-function $\overline{Q}_i^\xi \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}_i|}$ and the value function $V_i^\xi \in \mathbb{R}^{|\mathcal{S}|}$ of agent i are defined as $\overline{Q}_i^\xi(s,a_i) = \mathbb{E}_{a_{-i}\sim\xi_{-i}(\cdot|s_{-i})}[Q_i^\xi(s,a_i,a_{-i})]$ for all (s,a_i) and $V_i^\xi(s) = \mathbb{E}_{a_i\sim\xi_i(\cdot|s_i)}[\overline{Q}_i^\xi(s,a_i)]$ for all s, where we use s_{-i}, a_{-i}, and ξ_{-i} to denote the joint state, the joint action, and the joint policy of the agents in $\mathcal{N}/\{i\}$, respectively. With the initial state distribution μ, we define $J_i(\xi) = \mathbb{E}_{s\sim\mu}[V_i^\xi(s)]$. Finally, we define the advantage function of agent i as $A_i^\xi(s,a) = Q_i^\xi(s,a) - V_i^\xi(s)$ for all (s,a), and the averaged advantage function of agent i as $\overline{A}_i^\xi(s,a_i) = \overline{Q}_i^\xi(s,a_i) - V_i^\xi(s)$ for all (s,a_i). When the policy uses softmax parameterization with parameter θ, we may abuse the policy parameter θ to represent the policy ξ for simplicity. For example, we may write $J_i(\theta)$ for $J_i(\xi^\theta)$.

Discounted State Visitation Distribution. Given a policy ξ and an initial state s', we define the *discounted state visitation distribution* as $d_{s'}^\xi(s) = (1-\gamma)\sum_{t=0}^\infty \gamma^t \Pr^\xi[s(t) = s \mid s(0) = s']$ for all $s \in \mathcal{S}$, where $\Pr^\xi[s(t) = s \mid s(0) = s']$ denotes the probability that $s(t) = s$ given that the initial state is s' and the global policy is ξ. We use $d^\xi(s) := \mathbb{E}_{s'\sim\mu}[d_{s'}^\xi(s)]$ to represent the discounted state visitation distribution when the initial state distribution is μ.

3 NETWORKED MPGS

Our focus is a class of networked multi-agent Markov games that we named NMPGs, which is defined in the following.

Definition 3.1. *A multi-agent Markov game is called a κ_G-NMPG (where κ_G is a non-negative integer) if there exists a set of local potential functions $\{\Phi_i\}_{i\in\mathcal{N}}$, where $\Phi_i : \Xi \to \mathbb{R}$ for all $i \in \mathcal{N}$, such that the following equality holds for any $i \in \mathcal{N}$, $j \in \mathcal{N}_i^{\kappa_G}$, $\xi_j, \xi_j' \in \Xi_j$, and $\xi_{-j} \in \Xi_{-j}$:*

$$J_j(\xi_j', \xi_{-j}) - J_j(\xi_j, \xi_{-j}) = \Phi_i(\xi_j', \xi_{-j}) - \Phi_i(\xi_j, \xi_{-j}). \quad (2)$$

Definition 3.1 states that when agent j changes its local policy, the change in its objective function $J_j(\cdot, \xi_{-j})$ can be measured by the change of local potential functions from any agent in its κ_G-hop neighborhood. The non-negative integer κ_G is determined by the networked MPG setting and reflects the extent to which the networked MPG is relaxed from an MPG. Recall that in the definition of a standard MPG [Leonardos et al., 2022], there exists a (global) potential function Φ such that Eq. (2) holds with Φ_i being replaced by Φ for all i. Therefore, an MPG is always an NMPG (by choosing $\Phi_i = \Phi$ for all i), and hence NMPG represents a strictly broader class of games. More discussions are given in Appendix F.1, and a concrete example of an NMPG is presented in Section 3.1.

Due to the boundedness of the reward function and Eq. (2), the local potential functions are uniformly bounded from above and below, i.e., there exist $\Phi_{\min}, \Phi_{\max} > 0$ such that $\Phi_i(\xi) \in [\Phi_{\min}, \Phi_{\max}]$ for all $i \in \mathcal{N}$ and $\xi \in \Xi$. See Appendix F.6 for more details.

Unlike in single-agent RL or cooperative MARL, the optimal policy is not well-defined in the competitive setting, and thus our goal is to design algorithms that learn Nash equilibria of NMPGs. We next introduce the concepts of Nash equilibrium, Nash gap, and averaged Nash regret.

Definition 3.2. *A global policy ξ is a Nash equilibrium if $J_i(\xi_i, \xi_{-i}) \geq J_i(\xi_i', \xi_{-i})$ for all $\xi_i' \in \Xi_i$ and $i \in \mathcal{N}$.*

To measure the performance of a policy by its "distance" to a Nash equilibrium, we use the Nash gap.

Definition 3.3. *Given a global policy ξ, agent i's Nash gap and the global Nash gap are defined as*

$$\textit{NE-Gap}_i(\xi) := \max_{\xi_i'} J_i(\xi_i', \xi_{-i}) - J_i(\xi_i, \xi_{-i}),$$
$$\textit{NE-Gap}(\xi) := \max_{i \in \mathcal{N}} \textit{NE-Gap}_i(\xi).$$

With NE-Gap$(\cdot)$ defined above, given $\hat{\epsilon} > 0$, we say that a policy ξ is an $\hat{\epsilon}$-approximate Nash equilibrium if NE-Gap$(\xi) \leq \hat{\epsilon}$. When using a softmax policy with parameter θ, we may abuse the notation to denote NE-Gap$_i(\theta)$ for NE-Gap$_i(\xi^\theta)$ and also NE-Gap(θ) for NE-Gap(ξ^θ).

While Definition 3.3 enables us to measure the performance of a single policy, in MARL, most algorithms iterate over a sequence of policies. To measure the performance of a sequence of policies, we use the averaged Nash regret, which is defined in the following.

Definition 3.4. *Given a sequence of M policies $\{\xi(0), \xi(1), \ldots, \xi(M-1)\}$, the averaged Nash regret of agent i and the global averaged Nash regret are defined as*

$$\textit{Avg-Nash-Regret}_i(M) = \frac{1}{M} \sum_{m=0}^{M-1} \textit{NE-Gap}_i(\xi(m)),$$
$$\textit{Avg-Nash-Regret}(M) = \max_{i \in \mathcal{N}} \textit{Avg-Nash-Regret}_i(M).$$

Note that a similar concept called "Nash Regret" was previously introduced in Ding et al. [2022], and is defined as

$$\text{Nash-Regret}(M) = \frac{1}{M} \sum_{m=0}^{M-1} \max_{i \in \mathcal{N}} \text{NE-Gap}_i(\xi(m)). \quad (3)$$

By using Jensen's inequality and the fact that the maximum of a set of positive real numbers is less than the summation, we easily have Avg-Nash-Regret$(M) = \Theta($Nash-Regret$(M))$. See Appendix F.4 for the proof. As a result, Avg-Nash-Regret(M) and Nash-Regret(M) have the same rate of convergence (up to a multiplicative constant that depends on the number of agents).

3.1 AN EXAMPLE OF NMPGS

To illustrate the model, we present an extension of classical congestion games [Roughgarden and Tardos, 2004] and distributed welfare games [Marden and Wierman, 2013]. In this example, n agents are located on a traffic network $\mathcal{T} = (\mathcal{V}, \zeta)$, where $\mathcal{V}$ denotes the set of nodes and ζ denotes the set of directed edges with self-loops[1]. The objective of each agent i is to commute from its start node h_i to its destination d_i. In this example, the local state $s_i(t)$ of agent i at time t is its current location (a node $v \in \mathcal{V}$). By choosing a directed edge $(v, u) \in \zeta$ as its local action $a_i(t)$ at time t, agent i will transit to state $s_i(t+1) = u$ at time $t+1$. Without the loss of generality, we assume an agent will stay at the same node after it arrives at its destination.

The reward of agent i is defined as $r_i(t) = 0$ if $s_i(t) = d_i$, $r_i(t) = -\bar{\epsilon}$ if $s_i(t+1) = s_i(t)$, and $r_i(t) = -\bar{\epsilon} - N(a_i(t), t)$ otherwise, where $\bar{\epsilon} > 0$ is a constant and $N(e, t)$ denote the number of agents that chooses edge e at time t. The reward is designed so that the agent incurs a time cost of $\bar{\epsilon}$ for every step spent on its trip and a congestion cost of $N(a_i(t), t)$ depending on the traffic on the edge it travels through. The congestion cost is avoided if the agent chooses to wait at its current location (i.e., $s_i(t+1) = s_i(t)$). Each agent's goal is to maximize its expected discounted cumulative reward $\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_i(t)\right]$.

To see that this congestion game fits in our NMPG framework, consider the following communication network $\mathcal{G}$: agents i and j are neighbors if and only if there exists a global policy ξ such that $\sum_{t=0}^{\infty} \Pr(s_i(t) = s_j(t), s_i(t) \neq d_i, s_j(t) \neq d_j) > 0$. Under this communication network, the transition kernel is completely local because the next state of any agent i is decided completely locally and the local reward of agent i is a function that depends on the 1-hop local states and actions $(s_{\mathcal{N}_i^1}, a_{\mathcal{N}_i^1})$. We provide more discussion of this example and numerical simulations using it in Appendix A.

[1] Note that the traffic network and the communication network $\mathcal{G}$ may be different.

4 ALGORITHM DESIGN

We now present a novel algorithm for solving NMPGs. Our approach uses a combination of independent policy gradient (IPG) with localized TD-learning to form a localized actor-critic framework.

4.1 ACTOR: INDEPENDENT POLICY GRADIENT

Suppose that the agents have complete knowledge about the underlying model (e.g., reward function and transition dynamics). Then a popular approach for solving MPGs is to use IPG, which is presented in Algorithm 1 [Leonardos et al., 2022, Zhang et al., 2022b, Ding et al., 2022, Fox et al., 2022, Zhang et al., 2022a].

Algorithm 1 Independent Policy Gradient

1: **Input:** Initialization $\theta_i(0) = 0, \forall\, i \in \mathcal{N}$.
2: **for** $m = 0, 1, 2, \cdots, M-1$ **do**
3: $\quad \theta_i(m+1) = \theta_i(m) + \beta \nabla_{\theta_i} J_i(\theta(m))$ for all $i \in \mathcal{N}$
4: **end for**

In each round of Algorithm 1, each agent simultaneously updates its policy by implementing gradient ascent (in the policy space) w.r.t. their own objective function (cf. Algorithm 1 Line 3). Notably, to carry out Algorthm 1, each agent only needs to know its own policy. While Algorithm 1 is promising, it is not a model-free algorithm as computing the gradient requires knowledge of the underlying MDP model. This motivates the design of a critic to help estimate the gradient.

4.2 CRITIC: LOCALIZED TD(λ) WITH LINEAR FUNCTION APPROXIMATION

To motivate the design of the critic, we first present an explicit expression of the policy gradient of agent i [Sutton et al., 1999]:

$$\nabla_{\theta_i} J_i(\theta) = \sum_{t=0}^{\infty} \gamma^t \mathbb{E}_{\xi^\theta} \big[\nabla_{\theta_i} \log \xi_i^{\theta_i}(a_i(t)|s_i(t)) \times \overline{Q}_i^{\theta}(s(t), a_i(t)) \big]. \tag{4}$$

Similar versions of policy gradient theorems under different multi-agent settings were previously developed in Zhang et al. [2022a], Mao et al. [2022]. For completeness, we present a proof of Eq. (4) in Appendix F.2.

In view of Eq. (4), to estimate $\nabla_{\theta_i} J_i(\theta)$, the key is to construct an estimate of the averaged Q-function $\overline{Q}_i^{\theta}$. However, directly estimating the averaged Q-function of agent i requires information about the global state, incurring long-distance communication. To localize the algorithm, we introduce a hyper-parameter $\kappa_c \in \mathbb{N}$, and for each agent, we learn an approximation of the averaged Q-function (which we refer to as the κ_c-truncated averaged Q-function) using only information in its κ_c-hop neighborhood.

Truncated Averaged Q-functions. Given the non-negative integer κ_c, agent $i \in \mathcal{N}$, and a global policy parameter θ, we define $\mathcal{Q}_i^{\theta,\kappa_c}$ as the class of κ_c-truncated averaged Q-functions w.r.t. $\overline{Q}_i^{\theta}$. Specifically,

$$\mathcal{Q}_i^{\theta,\kappa_c} = \Big\{ \overline{Q}_i^{\theta,\kappa_c} \in \mathbb{R}^{|\mathcal{S}_{N_i^{\kappa_c}}||\mathcal{A}_i|} \;\Big|\; \exists\, u_i \in \Delta(\mathcal{S}_{-N_i^{\kappa_c}}) \text{ s.t. } \overline{Q}_i^{\theta,\kappa_c}(s_{N_i^{\kappa_c}}, a_i) = \mathbb{E}_{s_{-N_i^{\kappa_c}} \sim u_i} \Big[\overline{Q}_i^{\theta}(s_{N_i^{\kappa_c}}, s_{-N_i^{\kappa_c}}, a_i) \Big], \; \forall\, (s_{N_i^{\kappa_c}}, a_i) \in \mathcal{S}_{N_i^{\kappa_c}} \times \mathcal{A}_i \Big\}.$$

Note that when $\kappa_c \geq \max_{i,j} \text{dist}(i,j)$, there is essentially no truncation, i.e., any element in $\mathcal{Q}_i^{\theta,\kappa_c}$ is equal to $\overline{Q}_i^{\theta}$. When $\kappa_c < \max_{i,j} \text{dist}(i,j)$, we have the following *exponential-decay property*. See Appendix F.3 for the proof.

Lemma 4.1. *For any $\kappa_c \in \mathbb{N}$, agent i, and global policy parameter θ, it holds that*

$$\sup_{\overline{Q}_i^{\theta,\kappa_c} \in \mathcal{Q}_i^{\theta,\kappa_c}} \max_{s,a_i} \Big| \overline{Q}_i^{\theta,\kappa_c}(s_{N_i^{\kappa_c}}, a_i) - \overline{Q}_i^{\theta}(s, a_i) \Big| \leq \frac{2 \min\big(\gamma^{\kappa_c - \kappa_r + 1}, 1\big)}{1-\gamma}. \tag{5}$$

In view of Lemma 4.1, the κ_c-truncated averaged Q-function approximates the averaged Q-function (at a geometric rate) as κ_c increases. Therefore, it is enough for the critic to estimate an arbitrary κ_c-truncated averaged Q-function within the class $\mathcal{Q}_i^{\theta,\kappa_c}$. It is worth noting that the use of truncated Q-functions and the exponential-decay property have been widely exploited in the cooperative MARL literature for communication and dimension reduction in recent years [Qu et al., 2020, Gu et al., 2021b, Lin et al., 2021]. In this work, we show how to use such an approach in a non-cooperative setting for the first time.

Linear Function Approximation. While using the κ_c-truncated Q-functions enables us to overcome the computational bottleneck as the number of agents increases, there is still the challenge due to the curse of dimensionality. To further reduce the parameter dimension, we use linear function approximation. To be specific, for each $i \in \mathcal{N}$, let $\phi_i : \mathcal{S}_{\mathcal{N}_i^{\kappa_c}} \times \mathcal{A}_i \to \mathbb{R}^{d_i}$ be a feature mapping of agent i. Then, with weight vector $w_i \in \mathbb{R}^{d_i}$, we consider approximating the κ-truncated Q-functions using $\hat{Q}_i(s_{\mathcal{N}_i^{\kappa_c}}, a_i, w_i) = \langle \phi_i(s_{\mathcal{N}_i^{\kappa_c}}, a_i), w_i \rangle$ for all $(s_{\mathcal{N}_i^{\kappa}}, a_i)$. Let $\tilde{\phi}_i(s, a_i) = \phi_i(s_{N_i^{\kappa_c}}, a_i)$ for any $i \in \mathcal{N}$, $s \in \mathcal{S}$, and $a_i \in \mathcal{A}_i$. That is, given an agent i, for each pair (s, a_i) of global state and local action, we look at the states of agents in agent i's κ_c-hop neighborhood (i.e., $s_{N_i^{\kappa_c}}$) and agent i's action (i.e., a_i) and assign the vector $\phi(s_{N_i^{\kappa_c}}, a_i)$ to $\tilde{\phi}_i(s, a_i)$. Then agent i's feature matrix Ω_i is defined

to be an $|\mathcal{S}||\mathcal{A}_i| \times d_i$ matrix with its (s, a_i)-th row being $\tilde{\phi}_i^\top(s, a_i)$, where $(s, a_i) \in \mathcal{S} \times \mathcal{A}_i$.

We propose a novel policy evaluation algorithm called localized TD(λ) with linear function approximation, which is presented in Algorithm 2. The algorithm can be viewed as an extension of the classical TD(λ) with linear function approximation [Tsitsiklis and Van Roy, 1997] to the case where we estimate the κ_c-truncated averaged Q-functions using local information.

Algorithm 2 Localized TD(λ) with Linear Function Approximation

1: **Input**: Target policy ξ^θ, positive integers K and $\kappa_c \geq \kappa_r$, initializations $w_i(0) = 0$ for all i, step size $\alpha > 0$, $\lambda \in [0, 1)$, and $\epsilon > 0$.
2: Construct ϵ-exploration policy $\hat{\xi}_i(a_i|s_i) = (1-\epsilon)\xi_i^{\theta_i}(a_i|s_i) + \epsilon/|\mathcal{A}_i|$, for all i, a_i, and s_i.
3: The agents use the joint policy $\hat{\xi} = (\hat{\xi}_1, \hat{\xi}_2, \cdots, \hat{\xi}_n)$ to collect a sequence of samples $\tau = \{(s(t), a(t), r(t))\}_{0 \leq t \leq K-1} \cup \{s(K)\}$
4: **for** $i = 1, 2, \cdots, n$ **do**
5: $\quad \tau|_{(i,\kappa_c)} := \{(s_{\mathcal{N}_i^{\kappa_c}}(t), a_i(t), r_i(t))\}_{0 \leq t \leq K-1} \cup \{s_{\mathcal{N}_i^{\kappa_c}}(K)\}$
6: $\quad$ **for** $t = 0, 1, \cdots, K-1$ **do**
7: $\quad\quad \delta_i(t) = \phi_i(s_{\mathcal{N}_i^{\kappa_c}}(t), a_i(t))^\top w_i(t) - r_i(t) - \gamma\phi_i(s_{\mathcal{N}_i^{\kappa_c}}(t+1), a_i(t+1))^\top w_i(t)$
8: $\quad\quad w_i(t+1) = w_i(t) - \alpha\delta_i(t)\zeta_i^{\kappa_c}(t)$
9: $\quad\quad \zeta_i^{\kappa_c}(t+1) = (\gamma\lambda)\zeta_i^{\kappa_c}(t) + \phi_i(s_{\mathcal{N}_i^{\kappa_c}}(t+1), a_i(t+1))$
10: $\quad$ **end for**
11: **end for**
12: **Return** $\{w_i(K)\}_{i \in \mathcal{N}}$.

Note from Algorithm 2 Line 2 that we use ϵ-exploration policies to ensure exploration in localized TD(λ). Denote the set of all ϵ-exploration policies by Ξ^ϵ. Importantly, agent i requires only the states and the actions of the agents in its κ_c-hop neighborhood to carry out the algorithm, where κ_c can be viewed as a tunable parameter that trades off the communication effort and the accuracy. In particular, the larger κ_c is, the closer the κ_c-truncated averaged Q-function is to the true averaged Q-function, albeit at a cost of requiring more communication among agents.

4.3 LOCALIZED ACTOR-CRITIC

Combining IPG with localized TD(λ), we arrive at a localized actor-critic algorithm for solving NMPGs, which is presented in Algorithm 3.

The algorithm consists of three major steps. First, in Algorithm 3 Line 3, each agent calls localized TD(λ) with linear function approximation for policy evaluation and outputs a weight vector w_i^m for all $i \in \mathcal{N}$. Then, in Algorithm 3 Lines 4 – 8, each agent uses the averaged Q-function estimate to iteratively construct an estimate of the independent policy gradient. Specifically, since the independent policy gradient is an expected discounted sum of the averaged Q-functions (cf. Eq. (4)), we essentially construct an estimator $\Delta_i^T(m)$ (cf. Algorithm 3 Line 8) of it by taking average of total T samples $\{\eta_i^t(m)\}_{0 \leq t \leq T-1}$ (cf. Algorithm 3 Line 6). Finally, in Algorithm 3 Line 9, using the estimated gradient, each agent implements an approximate version of the IPG algorithm presented in Algorithm 1.

Compared with Algorithm 1, Algorithm 3 has the following strengths: (1) the algorithm is model-free, (2) due to the use of truncated Q-functions, each agent only requires information from its κ_c-hop neighborhood to carry out the algorithm, which eliminates long-distance communication along the network, and (3) the algorithm, to some extent, overcomes the curse of dimensionality thanks to the use of linear function approximation.

5 ALGORITHM ANALYSIS

We next present the main results of the paper. We formally state our assumptions in Section 5.1 and then present convergence bounds for Algorithms 1, 2, and 3 in Section 5.2. A proof sketch of our main theorems is given in Section 5.3.

5.1 ASSUMPTIONS

We make the following assumptions.

Assumption 5.1. *There exists a decreasing function $\nu : \mathbb{N} \to \mathbb{R}^+$ such that:*

$$\begin{aligned} &\left|\Phi_i(\theta_{N_i^\kappa}, \theta'_{-N_i^\kappa}) - \Phi_i(\theta_{N_i^\kappa}, \theta_{-N_i^\kappa})\right| \\ &\leq \nu(\kappa) \max_{j \in -N_i^\kappa} \left\|\theta'_j - \theta_j\right\|, \ \forall\, \kappa \in \mathbb{N}, \end{aligned} \tag{6}$$

where $\Phi_i(\theta)$ is the short-hand notation for $\Phi_i(\xi^\theta)$.

Assumption 5.1 captures the idea that, for each agent, its potential function is less impacted by the agents far away, and can be viewed as a generalization of the decay property of the Q-functions in the existing literature to the networked MPG setting [Qu et al., 2020, Lin et al., 2021, Zhang et al., 2022c]. In the extreme case where κ exceeds the diameter $\max_{i,j} \text{dist}(i, j)$ of the network, we have $\nu(\kappa) = 0$. Note that this assumption is automatically satisfied for our illustrative example in Section 3.1, where changing the policy of an agent will only affect its direct neighbors. In Appendix F.5, we show that this assumption is also satisfied when each local potential function admits a stage-wise representation [Zhang et al., 2022a].

Algorithm 3 Localized Actor-Critic

Input: Non-negative integers M, T, K, H, $\kappa_c \geq \kappa_r$, and a positive real number $\epsilon > 0$, initializations $\theta_i(0) = 0$ for all i, and $\Delta_i^0(m) = 0$ for all i and m.
for $m = 0, 1, 2, \cdots, M-1$ **do**
  All agents simultaneously execute localized TD(λ) with linear function approximation (with K iterations) to estimate their κ_c-truncated averaged Q-function $T^i_{\kappa_c}\overline{Q}_i^{\theta(m)}$, $i \in \mathcal{N}$, and output weight vectors $\{w_i^m\}_{i\in\mathcal{N}}$. ▷ Critic Update
  for $t = 0, 1, \cdots, T-1$ **do**
    The agents use the joint policy $\xi^{\theta(m)} = (\xi_1^{\theta_1(m)}, \xi_2^{\theta_2(m)}, \cdots, \xi_n^{\theta_n(m)})$ to collect a sequence of samples $\{(s^t(k), a^t(k))\}_{0\leq k\leq H-1}$
    $\eta_i^t(m) = \sum_{k=0}^{H-1} \gamma^k \nabla_{\theta_i} \log \xi_i^{\theta_i(m)}(a_i^t(k)|s_i^t(k))\phi_i(s^t_{\mathcal{N}_i^{\kappa_c}}(k), a_i^t(k))^\top w_i^m$
    $\Delta_i^{t+1}(m) = \frac{t}{t+1}\Delta_i^t(m) + \frac{1}{t+1}\eta_i^t(m)$
  end for
  $\theta_i(m+1) = \theta_i(m) + \beta\Delta_i^T(m)$ ▷ Actor Update
end for

Assumption 5.2. *It holds that* $\inf_\theta \min_{s\in\mathcal{S}} d^\theta(s) > 0$, *where we recall that* d^θ *is the discounted state visitation distribution under a softmax policy* ξ^θ

Assumption 5.2 states that every state can be visited with positive probability under any policy, which easily holds when the initial state distribution $\mu(\cdot)$ is supported on the entire state space. This assumption is standard and has been used in, e.g., Zhang et al. [2022a], Agarwal et al. [2021], Mei et al. [2020]. Under Assumption 5.2, we define $D = 1/\inf_\theta \min_{s\in\mathcal{S}} d^\theta(s)$, which is finite.

Assumption 5.3. *There exists a joint policy* ξ *such that the Markov chain* $\{s(t)\}$ *induced by* ξ *is uniformly ergodic.*

Under Assumption 5.3, [Zhang et al., 2022c, Lemma 4] implies a uniform exploration property for the Markov chain $\{(s(t), a(t))\}$ induced by any policy with entries bounded away from zero, which includes ϵ-exploration policy. Therefore, for any $\hat{\xi} \in \Xi^\epsilon$, the Markov chain $\{(s(t), a(t))\}$ induced by $\hat{\xi}$ has a unique stationary distribution, denoted by $\overline{\pi}^{\hat{\xi}} \in \Delta(\mathcal{S}\times\mathcal{A})$, which satisfies $\pi_{\min} := \inf_{\hat{\xi}\in\Xi^\epsilon} \min_{i\in\mathcal{N}} \min_{s_{\mathcal{N}_i^{\kappa_c}}, a_i} \overline{\pi}^{\hat{\xi}}(s_{\mathcal{N}_i^{\kappa_c}}, a_i) > 0$.

While Assumption 5.2, to some extent, already ensures uniform exploration of our policy class, we further impose Assumption 5.3 to deal with the Markovian sampling in Algorithm 3. This type of assumption is standard in the existing literature even for the single-agent setting [Srikant and Ying, 2019, Tsitsiklis and Van Roy, 1997].

Assumption 5.4. *For all* $i \in \mathcal{N}$, *the feature mapping is normalized so that* $\max_{i,s,a_i} \|\tilde{\phi}_i(s, a_i)\| \leq 1$. *In addition, the feature matrix* Ω_i *(the row vectors of which are* $\{\tilde{\phi}_i^\top(s, a_i)\}_{(s,a_i)\in\mathcal{S}\times\mathcal{A}_i}$*) has linearly independent columns.*

Assumption 5.4 is indeed without loss of generality because neither disregarding dependent features nor performing feature normalization changes the approximation power of the function class [Bertsekas and Tsitsiklis, 1996].

To state our last assumption, let $D^{\hat{\xi}} \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|\times|\mathcal{S}||\mathcal{A}|}$ be the diagonal matrix with diagonal entries $\{\overline{\pi}^{\hat{\xi}}(s, a)\}_{(s,a)\in\mathcal{S}\times\mathcal{A}}$. Since $D^{\hat{\xi}}$ has strictly positive diagonal entries under Assumption 5.3 and the feature matrix Ω_i has linearly independent columns for all i, we have $\underline{\lambda} := \min_{i\in\mathcal{N}} \inf_{\hat{\xi}\in\Xi^\epsilon} \lambda_{\min}(\Omega_i D^{\hat{\xi}} \Omega_i) > 0$, where $\lambda_{\min}(\cdot)$ returns the smallest eigenvalue of a positive definite matrix. For any $i \in \mathcal{N}$ and $\theta \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$, let $c_i(\theta) := \min_s \sum_{a_i^*\in\arg\max_{a_i} \overline{Q}_i^\theta(s,a_i)} \xi_i^{\theta_i}(a_i^*|s_i)$.

Assumption 5.5. $c := \inf_{m\geq 0} \min_{1\leq i\leq N} c_i(\theta(m)) > 0$, *where* $\{\theta(m)\}_{m\geq 0}$ *are policy parameters encountered from the algorithm trajectory (cf. Algorithm 3).*

The inequality stated in Assumption 5.5 is called a non-uniform Łojasiewicz inequality [Zhang et al., 2022a, Mei et al., 2020], which is used to connect the NE-Gap with the gradient of the objective function through gradient domination. This assumption automatically holds in the existing literature when the policy gradient is exact [Zhang et al., 2022a]. However, for Algorithm 3, due to the more challenging model-free setup and the presence of noise in sampling, c is not necessarily strictly positive, which motivates Assumption 5.5 as a means for analytical tractability. Further relaxing this assumption is our immediate future direction. One approach for removing Assumption 5.5 is to regularize the problem (e.g., using log-barrier regularization like in Zhang et al. [2022b]), which prevents the policy generated by IPG from being deterministic, albeit at a cost of introducing an asymptotic bias due to regularization.

5.2 RESULTS

We are now ready to present our main results. We first present the averaged Nash-regret bound of the IPG algorithm (cf. Algorithm 1) as a warm-up, then we present the finite-sample bound of Algorithm 3, which involves a critic

error. Finally, we present a concise bound of the critic estimation error when using our localized TD(λ) with linear function approximation. Given an arbitrary integer κ, let $n(\kappa) := \max_{i\in\mathcal{N}} |N_i^\kappa|$ be the size of the largest κ-hop neighborhood.

Theorem 5.6. *Consider $\{\theta_i(m)\}_{0\leq m\leq M-1}$ generated by Algorithm 1. Suppose that Assumptions 5.1, 5.2, and 5.5 are satisfied, and the step size $\beta = \frac{(1-\gamma)^3}{6n(\kappa_G)}$. Then,*

$$\begin{aligned}
&\textit{Avg-Nash-Regret}(M)\\
&\leq \mathcal{O}\left(\frac{D}{c}\sqrt{\frac{\max_{j\in\mathcal{N}}|\mathcal{A}_j| n(\kappa_G)(\Phi_{\max}-\Phi_{\min})}{(1-\gamma)^3 M}}\right)\\
&\quad + \mathcal{O}\left(\frac{D\sqrt{\max_{j\in\mathcal{N}}|\mathcal{A}_j|\nu(\kappa_G)}}{c(1-\gamma)}\right). \qquad (7)
\end{aligned}$$

The first term on the right-hand side of Eq. (7) goes to zero at a rate of $\mathcal{O}(M^{-1/2})$, which matches with the existing convergence rate of IPG for solving MPGs [Zhang et al., 2022a]. Note that, unlike in existing results, the total number of agents n does not appear in the bound. Instead, we have $n(\kappa_G)$, which captures the impact of network structure. The second term on the right-hand side of Eq. (7) arises because of the relaxation from MPG to NMPG (see Definition 3.1), which decreases with κ_G, and vanishes when $\kappa_G \geq \max_{i,j} \text{dist}(i,j)$.

We next move on to study Algorithm 3.

Theorem 5.7. *Consider $\{\theta_i(m)\}_{0\leq m\leq M-1}$ generated by Algorithm 3. Suppose that Assumptions 5.1 – 5.5 are satisfied, and $\beta = \frac{(1-\gamma)^3}{24n(\kappa_G)}$. Then,*

$$\begin{aligned}
&\mathbb{E}\left[\textit{Avg-Nash-Regret}(M)\right]\\
&\leq \frac{\sqrt{\max_{j\in\mathcal{N}}|\mathcal{A}_j|}D}{c}\Bigg\{\mathcal{O}\left(\frac{\sqrt{n(\kappa_G)(\Phi_{\max}-\Phi_{\min})}}{(1-\gamma)^{1.5}M^{1/4}}\right)\\
&\quad +\mathcal{O}\left(\frac{\sqrt{\nu(\kappa_G)}}{1-\gamma}\right)+\mathcal{O}\left(\frac{\sqrt{n(\kappa_G)}[1+(1-\gamma)\epsilon_{\textit{critic}}]}{(1-\gamma)^2M^{1/4}}\right)\\
&\quad +\mathcal{O}\left(\frac{\sqrt{n(\kappa_G)}\epsilon_{\textit{critic}}^{1/2}}{(1-\gamma)^{1.5}}\right)+\mathcal{O}\left(\frac{\sqrt{n(\kappa_G)}\gamma^{H/2}}{(1-\gamma)^2}\right)\Bigg\}, \qquad (8)
\end{aligned}$$

where $\epsilon_{\textit{critic}}$ stands for the critic estimation error in policy evaluation:

$$\epsilon_{\textit{critic}} = \sup_{\theta,i}\mathbb{E}^{1/2}\left[\sup_{s,a_i}\left|\overline{Q}_i^\theta(s,a_i) - \phi_i(s_{\mathcal{N}_i^{\kappa_c}},a_i)^\top w_i^\theta\right|^2\right].$$

The first two terms on the right-hand side of Eq. (8) are analogous to the two terms on the right-hand side of the IPG error bounds presented in Theorem 5.6. The last 4 terms are approximation errors for the independent policy gradient, which (in the order as they appear in the bound) consist of a localization error, an error incurred by using a finite sum (Algorithm 3 Line 6) to approximate an infinite sum (cf. Eq. (4)), a critic error, and an error incurred by using a finite average (Algorithm 3 Lines 4 – 8) to approximate an expectation (cf. Eq. (4)).

To establish an overall sample complexity bound of Algorithm 3, we need to specify how the critic error decays as a function of the number of iterations in localized TD(λ) with linear function approximation, which is presented in the following.

Theorem 5.8. *Consider $\{w_i(K)\}_{i\in\mathcal{N}}$ generated by Algorithm 2. Suppose that Assumption 5.3 is satisfied. Then, with appropriately chosen step size α (see Appendix D for the explicit requirements) and large enough K, we have*

$$\begin{aligned}
\epsilon_{\textit{critic}} &\leq \mathcal{O}(1-(1-\gamma)\underline{\lambda}\alpha)^{\frac{K}{2}} + \mathcal{O}\left[\frac{\alpha\log(1/\alpha)}{(1-\gamma)\underline{\lambda}}\right]^{1/2}\\
&\quad + \mathcal{O}\left(\frac{\epsilon_{\textit{app}}}{\pi_{\min}(1-\gamma)}\right) + \mathcal{O}\left(\frac{\gamma^{\kappa_c-\kappa_r}}{1-\gamma}\right)\\
&\quad + \mathcal{O}\left(\frac{n\epsilon}{(1-\gamma)^2}\right), \qquad (9)
\end{aligned}$$

where $\epsilon_{\textit{app}}$ stands for the function approximation error. See Appendix D for the explicit definition.

The first two terms on the right-hand side of Eq. (9) represent the convergence bias (which has geometric convergence rate) and the variance (which decreases with the step size α), and their behaviors agree with existing results on stochastic approximation [Srikant and Ying, 2019, Chen et al., 2022]. The third term arises from using linear function approximation and vanishes in the tabular setting where we use a complete basis. The fourth term represents the error between the averaged Q-function and the κ_c-truncated averaged Q-function, which is introduced to overcome the scalability issue when the number of agents increases. Note that the fourth term decays exponentially with the choice of κ_c, and vanishes when κ_c is greater than the diameter (i.e., $\max_{i,j}\text{dist}(i,j)$) of the network. The last term arises because of using ϵ-exploration behavior policies to ensure sufficient exploration.

Combining Theorem 5.7 and Theorem 5.8 leads to the following sample complexity bound.

Corollary 5.9. *To achieve $\mathbb{E}[\textit{Avg-Nash-Regret}(M)] \leq \tilde{\epsilon} + \mathcal{E}_{EX} + \mathcal{E}_{FA} + \mathcal{E}_{LO}$, the sample complexity is $\tilde{\mathcal{O}}(\tilde{\epsilon}^{-4})$, where $\mathcal{E}_{EX}$ stands for the induced error from exploration (cf. the last term on the right-hand side of Eq. (9)), $\mathcal{E}_{FA}$ stands for the function approximation error (cf. the third term on the right-hand side of Eq. (9)), and $\mathcal{E}_{LO}$ stands for the induced error from localization (cf. the summation of the second last term on the right-hand side of Eq. (9) and the third term on the right-hand side of Eq. (8)).*

In Corollary 5.9 The presence of $\mathcal{E}_{\text{EX}} + \mathcal{E}_{\text{FA}} + \mathcal{E}_{\text{LO}}$ are due to the fundamental limit of the problem, such as the approximation power of function class, using truncated averaged Q-functions to approximate global averaged Q-functions, and using "soft" policies to ensure exploration.

In single-agent RL, popular algorithms such as Q-learning and natural actor-critic are known to achieve $\tilde{\mathcal{O}}(\tilde{\epsilon}^{-2})$ sample complexity [Qu and Wierman, 2020, Lan, 2022]. While we study the more challenging setting of using localized algorithms to solve MARL problems, it is an interesting direction to investigate whether there is a fundamental gap. In addition, while Localized Actor-Critic (cf. Algorithm 3) is an independent learning algorithm, our theoretical results require all agents to follow the same learning dynamics, which suggests some implicit coordination among the agents. Although this is common in the existing literature [Leonardos et al., 2022, Ding et al., 2022, Zhang et al., 2022a], developing completely independent learning dynamics is an interesting future direction.

5.3 PROOF SKETCH

Analysis of the Actor. At a high level, we use a Lyapunov approach to analyze the policy update, where the potential function is a natural choice of the Lyapunov function. The key is to bound $\Phi_i(\theta(m+1)) - \Phi_i(\theta(m))$, $i \in \mathcal{N}$, in each iteration using the gradient of objective function $J_i(\cdot)$, which is related to NE-Gap of agent i through the non-uniform Łojasiewicz inequality [Zhang et al., 2022a, Mei et al., 2020]. To exploit the network structure and to remove the raw dependence on the total number of agents in the NMPG setting, instead of directly bounding $\Phi_i(\theta(m+1)) - \Phi_i(\theta(m))$, we perform the following decomposition:

$$\begin{aligned}
&\Phi_i(\theta(m+1)) - \Phi_i(\theta(m)) \\
&= \underbrace{\left[\Phi_i(\theta_{N_i^{\kappa_G}}(m+1), \theta_{-N_i^{\kappa_G}}(m)) - \Phi_i(\theta(m)\right]}_{(a)} \\
&\quad + \underbrace{\left[\Phi_i(\theta(m+1)) - \Phi_i(\theta_{N_i^{\kappa_G}}(m+1), \theta_{-N_i^{\kappa_G}}(m))\right]}_{(b)}.
\end{aligned}$$

The term (a) captures the policy change of the agents inside the κ_G-hop neighborhood of agent i, and the first step of bounding it is to use the smoothness property of the potential function, which is similar to that of Zhang et al. [2022a]. However, unlike existing analysis of IPG, we also need to bound the error in approximating the gradient, which can be decomposed into three error terms:

- e_1: error due to estimating the averaged Q-function, which is exactly the critic error;
- e_2: error due to the randomness in the trajectory sampling (see Algorithm 3 Lines 4 – 8), which has zero mean;
- e_3: error resulted from truncating the sample trajectory at horizon H (see Algorithm 3 Lines 6), which decays exponentially with H.

Term (b) results from the policy change of agents outside the κ_G-hop neighborhood of agent i, and is a decreasing function of κ_G (cf. Assumption 5.1).

Analysis of the Critic. The critic is designed to perform policy evaluation of a softmax policy ξ^θ using localized TD(λ) with linear function approximation. Similar to Chen et al. [2022], Srikant and Ying [2019], we formulate localized TD(λ) as a stochastic approximation algorithm and again use a Lyapunov approach to establish the finite-sample bound of the difference between $w_i(K)$ and w_i^θ, where w_i^θ is the solution to a properly defined projected Bellman equation associated with agent i.

The challenge lies in bounding the difference between the Q-function associated with the weight vector w_i^θ (denoted by $Q(w_i^\theta)$) and the true averaged Q-function $\overline{Q}_i^\theta$ of policy ξ^θ, which we decompose into a function approximation error, an error due to using ϵ-exploration policy, and an error due to truncating the averaged Q-function at its κ_c-hop neighborhood, and bound them separately. To achieve that, we develop a novel approach involving the construction of a "sub-chain", which is an auxiliary Markov chain with state space $\mathcal{S}_{\mathcal{N}_i^{\kappa_c}} \times \mathcal{A}_i$. See Appendix D for more details.

6 CONCLUSION

We study MARL in the context of MPGs and introduce a networked structure that allows agents to learn equilibria using local information. In particular, we develop a localized actor-critic framework for minimizing the averaged Nash regret of NMPGs. Importantly, the algorithm is scalable and uses function approximation. We provide finite-sample convergence bounds to theoretically support our proposed algorithm and conduct numerical simulations to demonstrate its empirical effectiveness.

An immediate future direction is to investigate whether there is a fundamental gap in the convergence rates between localized MARL algorithms and single-agent RL algorithms. It is also interesting to see if localized algorithms (with provable guarantees) can be designed to solve other classes of games beyond NMPGs.

Acknowledgements

This work is supported by NSF Grants CNS-2146814, CPS-2136197, CNS-2106403, NGSDI-2105648, with additional support from Amazon AWS. Yiheng Lin was supported by PIMCO graduate fellowship in Date Science and Amazon AI4Science fellowship. Zaiwei Chen was supported by PIMCO postdoctoral fellowship in Data Science and the Simoudis Discovery Prize.

References

Alekh Agarwal, Sham M Kakade, Jason D Lee, and Gaurav Mahajan. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. *The Journal of Machine Learning Research*, 22(1):4431–4506, 2021.

Dimitri Bertsekas and John N Tsitsiklis. *Neuro-dynamic programming*. Athena Scientific, 1996.

Jalaj Bhandari, Daniel Russo, and Raghav Singal. A Finite Time Analysis of Temporal Difference Learning With Linear Function Approximation. In *Conference On Learning Theory*, pages 1691–1692, 2018.

Deepayan Chakrabarti, Yang Wang, Chenxi Wang, Jurij Leskovec, and Christos Faloutsos. Epidemic thresholds in real networks. *ACM Transactions on Information and System Security (TISSEC)*, 10(4):1, 2008.

Zaiwei Chen, Sheng Zhang, Thinh T Doan, John-Paul Clarke, and Siva Theja Maguluri. Finite-sample analysis of nonlinear stochastic approximation with applications in reinforcement learning. *Automatica*, 146:110623, 2022.

Gal Dalal, Gugan Thoppe, Balázs Szörényi, and Shie Mannor. Finite sample analysis of two-timescale stochastic approximation with applications to reinforcement learning. In *Conference On Learning Theory*, pages 1199–1233. PMLR, 2018.

W Davis Dechert and SI O'Donnell. The stochastic lake game: A numerical solution. *Journal of Economic Dynamics and Control*, 30(9-10):1569–1587, 2006.

Dongsheng Ding, Chen-Yu Wei, Kaiqing Zhang, and Mihailo Jovanovic. Independent policy gradient for large-scale markov potential games: Sharper rates, function approximation, and game-agnostic convergence. In *International Conference on Machine Learning*, pages 5166–5220. PMLR, 2022.

Roy Fox, Stephen M Mcaleer, Will Overman, and Ioannis Panageas. Independent natural policy gradient always converges in markov potential games. In *International Conference on Artificial Intelligence and Statistics*, pages 4414–4425. PMLR, 2022.

Haotian Gu, Xin Guo, Xiaoli Wei, and Renyuan Xu. Mean-field controls with Q-learning for cooperative MARL: convergence and complexity analysis. *SIAM Journal on Mathematics of Data Science*, 3(4):1168–1196, 2021a.

Haotian Gu, Xin Guo, Xiaoli Wei, and Renyuan Xu. Mean-field multi-agent reinforcement learning: A decentralized network approach. *Preprint arXiv:2108.02731*, 2021b.

Tommi Jaakkola, Michael I Jordan, and Satinder P Singh. Convergence of stochastic iterative dynamic programming algorithms. In *Advances in neural information processing systems*, pages 703–710, 1994.

Fivos Kalogiannis, Ioannis Anagnostides, Ioannis Panageas, Emmanouil-Vasileios Vlatakis-Gkaragkounis, Vaggos Chatziafratis, and Stelios Stavroulakis. Efficiently computing nash equilibria in adversarial team markov games. *Preprint arXiv:2208.02204*, 2022.

Guanghui Lan. Policy mirror descent for reinforcement learning: Linear convergence, new sampling complexity, and generalized problem classes. *Mathematical programming*, pages 1–48, 2022.

Zachary J Lee, Tongxin Li, Steven H Low, and Sunash B Sharma. Systems and methods for adaptive ev charging, July 5 2022. US Patent 11,376,981.

Stefanos Leonardos, Will Overman, Ioannis Panageas, and Georgios Piliouras. Global convergence of multi-agent policy gradient in markov potential games. In *International Conference on Learning Representations*, 2022.

Yiheng Lin, Guannan Qu, Longbo Huang, and Adam Wierman. Multi-agent reinforcement learning in stochastic networked systems. *Advances in Neural Information Processing Systems*, 34:7825–7837, 2021.

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

Sergio Valcarcel Macua, Javier Zazo, and Santiago Zazo. Learning parametric closed-loop policies for markov potential games. In *International Conference on Learning Representations*, 2018.

Weichao Mao, Lin Yang, Kaiqing Zhang, and Tamer Basar. On improving model-free algorithms for decentralized multi-agent reinforcement learning. In *International Conference on Machine Learning*, pages 15007–15049. PMLR, 2022.

Jason R Marden and Adam Wierman. Distributed welfare games. *Operations Research*, 61(1):155–168, 2013.

Jincheng Mei, Chenjun Xiao, Csaba Szepesvari, and Dale Schuurmans. On the global convergence rates of softmax policy gradient methods. In *International Conference on Machine Learning*, pages 6820–6829. PMLR, 2020.

Washim Uddin Mondal, Vaneet Aggarwal, and Satish Ukkusuri. On the near-optimality of local policies in large cooperative multi-agent reinforcement learning. *Transactions on Machine Learning Research*, 2022a. ISSN 2835-8856.

Washim Uddin Mondal, Vaneet Aggarwal, and Satish V Ukkusuri. Can mean field control (mfc) approximate cooperative multi agent reinforcement learning (marl) with non-uniform interaction? In *Uncertainty in Artificial Intelligence*, pages 1371–1380. PMLR, 2022b.

Guannan Qu and Adam Wierman. Finite-time analysis of asynchronous stochastic approximation and Q-learning. In *Conference on Learning Theory*, pages 3185–3205. PMLR, 2020.

Guannan Qu, Adam Wierman, and Na Li. Scalable reinforcement learning of localized policies for multi-agent networked systems. In *Learning for Dynamics and Control*, pages 256–266. PMLR, 2020.

Tim Roughgarden and Éva Tardos. Bounding the inefficiency of equilibria in nonatomic congestion games. *Games and economic behavior*, 47(2):389–403, 2004.

Yuanyuan Shi, Guannan Qu, Steven Low, Anima Anandkumar, and Adam Wierman. Stability constrained reinforcement learning for real-time voltage control. In *2022 American Control Conference (ACC)*, pages 2715–2721. IEEE, 2022.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

Rayadurgam Srikant and Lei Ying. Finite-time error bounds for linear stochastic approximation and TD learning. In *Conference on Learning Theory*, pages 2803–2830. PMLR, 2019.

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999.

John N Tsitsiklis. Asynchronous stochastic approximation and Q-learning. *Machine learning*, 16:185–202, 1994.

John N Tsitsiklis and Benjamin Van Roy. An Analysis of Temporal-Difference Learning with Function Approximation. *IEEE Transactions on Automatic Control*, 42(5): 674–690, 1997.

Werner Vogels, Robbert van Renesse, and Ken Birman. The power of epidemics: Robust communication for large-scale distributed systems. *SIGCOMM Comput. Commun. Rev.*, 33(1):131–135, January 2003. ISSN 0146-4833. doi: 10.1145/774763.774784.

Huizhen Yu and Dimitri P Bertsekas. Convergence results for some temporal difference methods based on least squares. *IEEE Transactions on Automatic Control*, 54(7): 1515–1531, 2009.

Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *Handbook of reinforcement learning and control*, pages 321–384, 2021.

Rick Zhang and Marco Pavone. Control of robotic mobility-on-demand systems: a queueing-theoretical perspective. *The International Journal of Robotics Research*, 35(1-3): 186–203, 2016.

Runyu Zhang, Jincheng Mei, Bo Dai, Dale Schuurmans, and Na Li. On the global convergence rates of decentralized softmax gradient play in markov potential games. In *Advances in Neural Information Processing Systems*, 2022a.

Runyu Zhang, Zhaolin Ren, and Na Li. Gradient play in stochastic games: Stationary points and local geometry. In *25th International Symposium on Mathematical Theory of Networks and Systems (MTNS 2022)*, 2022b.

Yizhou Zhang, Guannan Qu, Pan Xu, Yiheng Lin, Zaiwei Chen, and Adam Wierman. Global Convergence of Localized Policy Iteration in Networked Multi-Agent Reinforcement Learning. *Preprint arXiv:2211.17116*, 2022c.