Newton-based Policy Search for Networked Multi-agent Reinforcement Learning

Giorgio Manganini giorgio.manganini@gssi.it
Department of Computer Science, Gran Sasso Science Institute (GSSI)
L'Aquila, Italy

Simone Fioravanti simone.fioravanti@gssi.it
Department of Computer Science, Gran Sasso Science Institute (GSSI)
L'Aquila, Italy

Giorgia Ramponi gramponi@ethz.ch
ETH AI Center
Zürich, Switzerland

Abstract

Newton's method is a standard optimization algorithm, characterized by a fast rate of convergence and used in many popular approximated methods that use second-order information. Despite its well understood theoretical properties, quadratic convergence rate and extended applications, Newton's method is seldom used for policy optimization in Multi-Agent Reinforcement Learning problems. In this work we investigate a distributed Newton consensus scheme for performing policy search in a networked cooperative environment, where the agents are endowed with private local rewards, though they aim to collaborate for maximizing the network-wise averaged long-term return. In the proposed algorithm, the agents seek for the parameters of the optimal global policy by locally computing an approximated Newton's direction for the global objective function, and sequentially update it in a distributed fashion by means of an average consensus procedure. The strategy is purely policy-based and does not involve any representation of the global value-function. We analyse the computational and theoretical properties of the algorithm and prove, under suitable assumptions, global converge to the true maximizer. Additionally, we provide convergence guarantees also under finite-sample conditions. Beside the theoretical properties, we perform numerical experiments showing the validity of the approach and highlight its improved convergence speed when compared to a simpler first-order distributed method.

Keywords: Multi-agent Learning, Reinforcement Learning, Distributed Optimization, Newton Method

1. Introduction

In this paper we take into account the *collaborative* MARL problem (Zhang et al., 2018b, 2019b), where collaborative agents perceive *local reward functions*, but aim to optimize their long-term average return common to all. In this setting, a central role is played by the communication protocol among the agent since it is vital to reach the common goal. A simplifying and common assumption is to have a central entity that receives the reward of all the agent and update their parameters to maximize the objective function. In this case, the problem reduces to a single-agent RL problem, and relative classical algorithms can be applied. However, this configuration has also many disadvantages: the framework is vulnerable to security attacks (with its single point of failure), can suffer from scalability and flexibility issues (*e.g.*, high communication and computational burden for the central node), and cannot guarantee any privacy requirement (which is a major issue for many multi-agent systems, from unmanned vehicles (Fax and Murray, 2004), robotics (Corke et al., 2005), and mobile sensor networks (Cortés et al., 2004)). To avoid these drawbacks, we remove here the presence of any central controller and investigate a fully decentralized protocol, in which the agents can rely only on local information and learn in a distributed manner. This framework, also called the Networked MARL problem, requires the agents to be connected by a possibly time-varying and sparse communication network. The agents are able to share their *public* information, as, for example, their policies or parameters, while retaining other *private* ones as the reward functions. The major difficulties with this distributed setting concern the performances and convergence guarantees of appropriate learning algorithms, which constitute exactly the scope and the contribution of our work.

Inspired by the recent developments in distributed/consensus optimization with networked agents, we propose a novel decentralized policy search gradient-based learning technique for cooperative agents with no central controller, where we consider heterogeneous and private reward functions for different agents, with the collective goal of maximizing the global average long-term return. Our algorithm exploits second-order Newton-like information of the (estimated) global objective function and updates the global policy parameters based on a consensus scheme. We avoid any auxiliary representation of the global value function and tackle the mutual dependencies among the agents by directly seeking the optimal parameter of the global policy. During the training, the agents do not need to share reward signals or value functions with other agents. On the contrary, each agent makes use of independent Newton updates, and auxiliary variables to compute the network-wise average Newton direction by exchanging only local gradients and Hessians estimations with their neighbors, making our approach fully distributed and privacy compliant.

Finally, we establish theoretical results on the algorithm's global and local convergence properties, which remarkably hold also in the case of finite-sample scenarios. Numerical results confirm the validity of the approach, significantly reducing the learning iterations. Beyond this, our first investigation of a Newton algorithm in MARL supports also the quest for more complicated but computationally cheaper quasi-Newton methods.

2. Preliminaries

In this section, we provide the necessary background on the collaborative (or cooperative) Continuous MARL problem; we define the problem setting, present the communication protocol and finally revise the (single-agent) Newton optimization method.

Cooperative Continuous MARL The classical theoretical formalism for MARL is the Continuous Markov Game (MG), which is defined by a tuple $\mathcal{M} = \langle N, \mathcal{S}, \{\mathcal{A}_i\}_{i\in\mathcal{N}}, \mathcal{P}, \{\mathcal{R}_i\}_{i\in\mathcal{N}}, \gamma\rangle$, where we have a team $\mathcal{N} = \{1, 2, \ldots, N\}$ of N agents, $\mathcal{S}$ is the state space, $\mathcal{A}_i$ is the action space for agent i, and $\mathcal{A} \triangleq \mathcal{A}_1 \times \cdots \times \mathcal{A}_N$ is the joint action space. Transition probabilities from state s to s' for any joint action a is given by $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$, with $\Delta(\Omega)$ denoting the set of probability measures over a generic set Ω. The function $\mathcal{R}_i : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the local reward received by the agent i after taking joint action a at state s, and $\gamma \in (0, 1)$ is the discount factor. Both s and a are available to all agents, whereas the reward function $\mathcal{R}_i$ is private for each agent i. Given the current state, each agent follows its policy $\pi_i : \mathcal{S} \to \Delta(\mathcal{A}_i)$ and hence selects his action. The resulting joint policy of all the agents $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ is equal to $\pi(a|s) = \prod_{i\in\mathcal{N}} \pi_i(a_i|s)$. In this paper we consider twice differentiable policies $\pi_{\boldsymbol{\psi}_i}$ parametrized by parameters $\boldsymbol{\psi}_i \in \mathbb{R}^{d_i}$. The goal, in the cooperative setting, is to find the joint policy $\pi_{\boldsymbol{\theta}}$, with $\boldsymbol{\theta} = [\boldsymbol{\psi}_1^\top, \ldots, \boldsymbol{\psi}_N^\top]^\top \in \mathbb{R}^d$ and $d = \sum_{i=1}^N d_i$, that maximizes the expected cumulative sum of the returns of all agents, represented by the global reward function $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, defined as $\mathcal{R}(s, a) = N^{-1} \sum_{i\in\mathcal{N}} \mathcal{R}_i(s, a)$, with $s \in \mathcal{S}$ and $a \in \mathcal{A}$. Formally, joint policies can be ranked by their expected discounted reward[1]

$$J(\boldsymbol{\theta}) = \mathop{\mathbb{E}}_{\tau\sim\mathcal{P},\pi_{\boldsymbol{\theta}}}[\mathcal{R}(\tau)] = \int_{\mathcal{T}} p(\tau|\pi_{\boldsymbol{\theta}})\mathcal{R}(\tau)\mathrm{d}\tau, \tag{1}$$

where $p(\tau|\pi_{\boldsymbol{\theta}}) = p(s_0) \prod_{t=1}^T \mathcal{P}(s_t|s_{t-1}, a_{t-1})\pi_{\boldsymbol{\theta}}(a_{t-1}|s_{t-1})$ is the probability density from which a trajectory $\tau = \{s_t, a_t\}_{t=0}^T \in \mathcal{T}$ of length T [2] is drawn, and $\mathcal{R}(\tau) = \sum_{t=0}^T \gamma^t \mathcal{R}(s_t, a_t)$ is the associated discounted global reward.

Networked communication protocol We consider a fully decentralized communication model, where both the reward and the action are received and executed locally by each agent. In this setting, there is no central controller able to solve the collaborative decision problem (*e.g.*, by collecting information and running the required computations), but each agent makes individual decisions based on local information. To foster collaboration, the agents are able to exchange information, under some privacy limitations, over a *communication network*. The communication network is modeled as an undirected graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where the set of agents represent the vertices, and the edges $\mathcal{E} \subset \mathcal{N} \times \mathcal{N}$ are the available communication links. We assume that the graph is connected, *i.e.*, there is path between any pair of nodes (i, j), and that the adjacency matrix of the graph $\boldsymbol{A} = [a_{ij}] \in \mathbb{R}^{N\times N}$, with $a_{ij} > 0$ only if $(i, j) \in \mathcal{E}$ and $a_{ij} = 0$ otherwise, is symmetric and stochastic, such that $\mathbf{1}_N \boldsymbol{A} = \mathbf{1}_N$ (where $\mathbf{1}_N \triangleq [1 \cdots 1]^\top \in \mathbb{R}^N$).

Newton Method The Newton method (Nocedal and Wright, 2006) is a popular optimization algorithm that, under suitable assumptions, guarantees quadratic convergence rates. This method employs the search direction obtained by

[1] For sake of clarity, in the rest of the paper we will often directly use the optimization variables $\boldsymbol{\theta}$ to refer to the corresponding parametrized policy $\pi_{\boldsymbol{\theta}}$.

[2] We consider episodic MDPs, which have not to be confused with a finite horizon problem where the optimal policy is non-stationary.

minimizing the second-order Taylor series approximation of the function $J(\boldsymbol{\theta})$ that we want to maximize around the current point $\boldsymbol{\theta}(k)$ and updates the variables according to:[3]

$$\begin{aligned}\boldsymbol{\theta}(k+1) &= \boldsymbol{\theta}(k) + \epsilon \boldsymbol{\phi}_{\mathrm{N}}(\boldsymbol{\theta}(k)),\\ \boldsymbol{\phi}_{\mathrm{N}}(\boldsymbol{\theta}) &= \boldsymbol{\nabla}^2 J(\boldsymbol{\theta})^{-1} \boldsymbol{\nabla} J(\boldsymbol{\theta}).\end{aligned} \tag{2}$$

Newton method is globally convergent, *i.e.*, it converges to a stationary point from any point in the domain and enjoys a quadratic convergence rate. These favourable properties hold when the Hessian matrix is positive definite over all the iterations and have a bounded condition number, *i.e.*, $mI \leq \boldsymbol{\nabla}^2 J(\boldsymbol{\theta}) \leq MI$ for some $m, M \in \mathbb{R}_+$. To enforce these conditions and guarantee convergence to (local) maximum even for non-concave functions, we define the matrix operator $[\cdot]_m$ as

$$[\boldsymbol{Z}]_m \triangleq \begin{cases} \boldsymbol{Z} & \text{if } \boldsymbol{Z} \geq \frac{m}{2}\boldsymbol{I} \\ \frac{m}{2}\boldsymbol{I} & \text{otherwise.} \end{cases} \tag{3}$$

3. Distributed Newton-based Optimization for Cooperative MARL

We propose here our new decentralized algorithm to solve the collaborative MARL problem defined in Section 2. First of all, given the structure of the global reward function $\mathcal{R}$, we can decouple the maximization of (1) in terms of the local reward functions $\mathcal{R}_i$ and the common set of variables $\boldsymbol{\theta}$ as

$$\max_{\boldsymbol{\theta} \in \mathbb{R}^d} J(\boldsymbol{\theta}) = \frac{1}{N} \sum_{i \in \mathcal{N}} J_i(\boldsymbol{\theta}), \quad \text{with} \quad J_i(\boldsymbol{\theta}) = \mathop{\mathbb{E}}_{\tau \sim \mathcal{P}, \pi_{\boldsymbol{\theta}}} [\mathcal{R}_i(\tau)]. \tag{4}$$

This is a distributed optimization problem with two main challenges. First, the objective function consists of a sum of N local contributions J_i, but each J_i is assumed to be known only by the agent i for the presence of the private reward $\mathcal{R}_i$. Second, (4) is decision-coupled by the global vector $\boldsymbol{\theta}$, since the objective function involves all the agents' policies because of the coupled dynamics of the (cooperative) Markov Game. In other words, the dependence of the local costs J_i on the global parameters $\boldsymbol{\theta}$ is intrinsically related to the game and prevents us from trivially split (4) into N independent problems.

3.1 Multi-agent Newton Algorithm with Consensus

Nonetheless, without sharing their private cost function, the agents are willing to share some information with their neighbors to collaboratively solve (4). To this end, we propose a distributed Newton-like algorithm and combine techniques of dynamic consensus (Kia et al., 2019; Zhu and Martínez, 2010) with stochastic gradient-based learning (Mazumdar et al., 2020). First, we assign local copies $\boldsymbol{\theta}_i \in \mathbb{R}^d$ of the global decision variable $\boldsymbol{\theta}$ to each agent i, and collect them in a single vector $\boldsymbol{\theta}_{\mathcal{N}} \triangleq [\boldsymbol{\theta}_1^\top, \ldots, \boldsymbol{\theta}_N^\top]^\top \in \mathbb{R}^{dN}$. Then, inspired by (Varagnolo et al., 2015), we adapt Newton's dynamics (2) for problem (4) and introduce N independent dynamical updates for each $\boldsymbol{\theta}_i$ as

$$\boldsymbol{\theta}_i(k+1) = (1-\epsilon)\boldsymbol{\theta}_i(k) + \epsilon \phi_{\mathrm{DN}}(\boldsymbol{\theta}_{\mathcal{N}}(k)), \tag{5}$$

where the common driving force $\phi_{\mathrm{DN}}() : \mathbb{R}^{dN} \to \mathbb{R}^d$ is defined by

$$\phi_{\mathrm{DN}}(\boldsymbol{\theta}_{\mathcal{N}}) \triangleq \left[\frac{1}{N}\sum_{i \in \mathcal{N}} \boldsymbol{H}_i(\boldsymbol{\theta}_i)\right]^{-1} \left[\frac{1}{N}\sum_{i \in \mathcal{N}} \boldsymbol{g}_i(\boldsymbol{\theta}_i)\right], \tag{6}$$

with $\boldsymbol{H}_i(\boldsymbol{\theta}) \triangleq \boldsymbol{\nabla}^2 J_i(\boldsymbol{\theta})$ and $\boldsymbol{g}_i(\boldsymbol{\theta}) \triangleq \boldsymbol{\nabla}^2 J_i(\boldsymbol{\theta})\boldsymbol{\theta} + \boldsymbol{\nabla} J_i(\boldsymbol{\theta})$. To give an idea of the reason why (5) embeds the centralized dynamics (2), let's consider the average quantity $\bar{\boldsymbol{\theta}}(k) \triangleq N^{-1}\sum_{i \in \mathcal{N}} \boldsymbol{\theta}_i(k)$. It is easy to prove that dynamics of $\bar{\boldsymbol{\theta}}(k)$ evolve according to (2) when applied to solve (4), *i.e.*, :

$$\begin{aligned}\bar{\boldsymbol{\theta}}(k+1) &= (1-\epsilon)\bar{\boldsymbol{\theta}}(k) + \epsilon \phi_{\mathrm{DN}}(\mathbf{1}_N \otimes \bar{\boldsymbol{\theta}}) \\ &= (1-\epsilon)\bar{\boldsymbol{\theta}}(k) + \epsilon \left[\bar{\boldsymbol{\theta}}(k) + \left(\frac{1}{N}\sum_{i \in \mathcal{N}} \boldsymbol{\nabla}^2 J_i(\bar{\boldsymbol{\theta}}(k))\right)^{-1} \frac{1}{N}\sum_{i \in \mathcal{N}} \boldsymbol{\nabla} J_i(\bar{\boldsymbol{\theta}}(k))\right] \\ &= \bar{\boldsymbol{\theta}}(k) + \epsilon \phi_{\mathrm{N}}(\bar{\boldsymbol{\theta}}(k)).\end{aligned} \tag{7}$$

[3] Where not differently specified, in this paper the gradient is computed with respect to the full argument of the function.

The above dynamics are exponentially stable thanks to Newton's method properties and, under same initial conditions $\boldsymbol{\theta}(0) = \bar{\boldsymbol{\theta}}(0)$, the trajectories coincide, *i.e.*, $\boldsymbol{\theta}_i(k) = \bar{\boldsymbol{\theta}}(k)$, $\forall i \in \mathcal{N}$ and $\forall k \geq 0$. Moreover, even when $\boldsymbol{\theta}_i(0)$ may be different, also dynamics (5) are exponentially stable (see Theorem 4), and the convergence of $\boldsymbol{\theta}_i$ to $\boldsymbol{\theta}^\star$ is guaranteed under the same Newton's assumptions for the convergence of $\bar{\boldsymbol{\theta}}$.

The update rule (5) does not represent yet a fully distributed scheme since it cannot be computed independently by each agent. Following (Varagnolo et al., 2015), we employ a dynamic average consensus-based mechanism (Nedić and Liu, 2018) where the agents, starting only from the local available quantities $\boldsymbol{H}_i(\boldsymbol{\theta}_i)$ and $\boldsymbol{g}_i(\boldsymbol{\theta}_i)$, are able to recover their network-wise average for computing ϕ_{DN}. Practically, we introduce two sets of auxiliary variables $\widetilde{\boldsymbol{g}}_i \in \mathbb{R}^d$ and $\widetilde{\boldsymbol{H}}_i \in \mathbb{R}^{d \times d}$ to track, respectively, the averages of $\boldsymbol{H}_i(\boldsymbol{\theta}_i)$ and $\boldsymbol{g}_i(\boldsymbol{\theta}_i)$ based on the local $\boldsymbol{\nabla} J_i(\boldsymbol{\theta}_i)$ and $\boldsymbol{\nabla}^2 J_i(\boldsymbol{\theta}_i)$. If the dynamics (5) of the $\boldsymbol{\theta}_i(k)$ are sufficiently slow with respect to the update of $\widetilde{\boldsymbol{g}}_i, \widetilde{\boldsymbol{H}}_i$, then $\widetilde{\boldsymbol{g}}_i(k)$ and $\widetilde{\boldsymbol{H}}_i(k)$ variables tend to converge to their average value (see (Kia et al., 2019) for a rigorous treatment and a proof of convergence for the dynamic consensus algorithms). Consequently, the product $[\widetilde{\boldsymbol{H}}_i(k)]_c^{-1}\widetilde{\boldsymbol{g}}_i(k)$ is the same across the agents, and it represents the Newton's direction ϕ_{DN}, thus leading all the $\boldsymbol{\theta}_i$ to be updated by the same force according to (5). The overall procedure is reported in Algorithm 1, and its convergence analyzed in Section 4.

Algorithm 1: Multi-agent Newton Algorithm with Consensus

```
Initialize
$\epsilon \in (0, 1], m > 0$
$\boldsymbol{\theta}_i(0) \leftarrow 0; \widetilde{\boldsymbol{g}}_i(0) = \boldsymbol{g}_i(\boldsymbol{\theta}_i(-1)) \leftarrow 0; \widetilde{\boldsymbol{H}}_i(0) = \boldsymbol{H}_i(\boldsymbol{\theta}_i(-1)) \leftarrow 0;$
for $k = 1, 2, \ldots$ do
    Compute $\boldsymbol{g}_i(\boldsymbol{\theta}_i(k-1)), \boldsymbol{H}_i(\boldsymbol{\theta}_i(k-1))$          // See Section 3.2
    Gather $\widetilde{\boldsymbol{g}}_j(k-1), \boldsymbol{g}_j(\boldsymbol{\theta}_j(k-1)), \widetilde{\boldsymbol{H}}_j(k-1), \boldsymbol{H}_j(\boldsymbol{\theta}_j(k-1))$ from neighbours $j \in \mathcal{N}_i$:
    Update
    $\widetilde{\boldsymbol{g}}_i(k) \leftarrow \sum_{j=1}^N a_{ij}[\widetilde{\boldsymbol{g}}_j(k-1) + \boldsymbol{g}_j(\boldsymbol{\theta}_j(k-1)) - \boldsymbol{g}_j(\boldsymbol{\theta}_j(k-2))]$
    $\widetilde{\boldsymbol{H}}_i(k) \leftarrow \sum_{j=1}^N a_{ij}[\widetilde{\boldsymbol{H}}_j(k-1) + \boldsymbol{H}_j(\boldsymbol{\theta}_j(k-1)) - \boldsymbol{H}_j(\boldsymbol{\theta}_j(k-2)]$
    $\boldsymbol{\theta}_i(k) \leftarrow (1-\epsilon)\boldsymbol{\theta}_i(k-1) + \epsilon[\widetilde{\boldsymbol{H}}_i(k-1)]_m^{-1}\widetilde{\boldsymbol{g}}_i(k-1)$
end
```

3.2 Learning Newton's Direction in Cooperative MARL

In the above description of the multi-agent Newton's method, we employed the agents' gradient and Hessian estimates of $J_i(\boldsymbol{\theta}_i)$ and, for clarity of exposition, postponed here their derivations. In the RL literature, a variety of approaches for policy gradient methods have been yielded, including finite-differences, simultaneous perturbation methods, and likelihood-ration methods (see, *e.g.*, (Peters and Schaal, 2008) for an overview). Being the latter the most prominent approach nowadays, we employ its formulation and obtain Newton's direction for Cooperatives Markov Games along the following lines (proof can be found in the Appendix).

Theorem 1 *Given a Markov Game $\mathcal{M}$ and a continuously twice-differentiable policy $\pi_{\boldsymbol{\theta}}(a|s)$, the general likelihood ratio estimators for the gradient and the Hessian of the expected-discounted return $J_i(\boldsymbol{\theta})$ in (4) of the agent i-th are:*

$$\boldsymbol{\nabla} J_i(\boldsymbol{\theta}) = \mathop{\mathbb{E}}_{\tau \sim \pi_{\boldsymbol{\theta}}, \mathcal{P}}[\mathcal{R}_i(\tau)\boldsymbol{\nabla}\log p(\tau|\boldsymbol{\theta})], \tag{8a}$$

$$\boldsymbol{\nabla}^2 J_i(\boldsymbol{\theta}) = \mathop{\mathbb{E}}_{\tau \sim \pi_{\boldsymbol{\theta}}, \mathcal{P}}\left[\mathcal{R}_i(\tau)\big(\boldsymbol{\nabla}\log p(\tau|\boldsymbol{\theta})\boldsymbol{\nabla}\log p(\tau|\boldsymbol{\theta})^\top + \boldsymbol{\nabla}^2 \log p(\tau|\boldsymbol{\theta})\big)\right] \tag{8b}$$

Similar derivations for the Hessian can be found in the single-agent RL literature, as in (Furmston and Barber, 2012; Shen et al., 2019; Manganini et al., 2015), or in the multi-agent RL competitive setting (Foerster et al., 2017; Ramponi and Restelli).

Equations (8) offers an exact and general formulation for the gradient and the Hessian, which can be estimated only from the data collected during the task execution and without the need for a transition model, thanks to the identity

$\nabla \log p(\tau|\boldsymbol{\theta}) = \sum_{t=0}^{T} \nabla \log \pi_{\boldsymbol{\theta}}(a_t|s_t)$. In the literature, there are different related expressions for the policy gradient and the Hessian, including Monte-Carlo methods (Williams, 1992; Baxter and Bartlett, 2001) and actor-critic methods (Konda and Tsitsiklis, 2000; Sutton et al., 1999; Kober and Peters, 2014). All these approaches avoid the practically unfeasible expectation over all possible trajectories and propose different methods to compute the trajectory reward $\mathcal{R}(\tau)$, also employing optimal constant baselines and value functions to reduce the variance of the estimates at the cost of some bias. In particular, a derivation for a sample-based estimator of the Hessian can be found in (Baxter and Bartlett, 2001), and in (Furmston et al., 2016) with alternative equivalent formulations.

When computing the local quantities $\boldsymbol{g}_i(\boldsymbol{\theta}_i)$ and $\boldsymbol{H}_i(\boldsymbol{\theta}_i)$ in Algorithm 1, each agent needs to evaluate (or approximate) the expected value w.r.t its own parameters $\boldsymbol{\theta}_i$, which would require impractical on-policy evaluations with $\pi_{\boldsymbol{\theta}_i}$. On the contrary, to collect the trajectories from the environment, we propose an on-line off-policy scheme and employ a global behavior policy $\pi_{\boldsymbol{\mu}}$ with the same structure of $\pi_{\boldsymbol{\theta}}$ (and therefore $\pi_{\boldsymbol{\theta}_i}$), where the vector $\boldsymbol{\mu}$ contains the local parameters $\boldsymbol{\psi}_i$ of the local policies π_i as contained in the local variables $\boldsymbol{\theta}_i$, *i.e.*, $\boldsymbol{\mu} = [\boldsymbol{\psi}_1^\top, \dots, \boldsymbol{\psi}_N^\top]^\top \in \mathbb{R}^d$. By applying importance sampling (IS) technique, we obtain the identity

$$\underset{\tau \sim \pi_{\boldsymbol{\theta}_i}, \mathcal{P}}{\mathbb{E}} [\cdot] = \underset{\tau \sim \pi_{\boldsymbol{\mu}}, \mathcal{P}}{\mathbb{E}} \left[\frac{p(\tau|\pi_{\boldsymbol{\theta}_i})}{p(\tau|\pi_{\boldsymbol{\mu}})} \cdot \right], \tag{9}$$

with the assumption that if $\pi_{\boldsymbol{\theta}_i}(a|s) > 0$, then $\pi_{\boldsymbol{\mu}}(a|s) > 0$, and the convention that the ratio is 0 if both policies are 0. Using (9) in (8), we can define $\nabla J_i(\boldsymbol{\theta}_i)$ and $\nabla^2 J_i(\boldsymbol{\theta}_i)$ as an expected value w.r.t. $\pi_{\boldsymbol{\mu}}$, *i.e.*, we can sample from the distribution $\pi_{\boldsymbol{\mu}}$ and then simply re-weight the trajectories to obtain an unbiased estimate w.r.t. $\pi_{\boldsymbol{\theta}_i}$ of each $\boldsymbol{g}_i(\boldsymbol{\theta}_i)$ and $\boldsymbol{H}_i(\boldsymbol{\theta}_i)$ (for a complete and formal derivation, see Appendix A). In order for all the agents to have the same vector $\boldsymbol{\mu}$, each agent i is required to transmit his local parameters $\boldsymbol{\psi}_i$ to all the others in the network. This can be done at the additional cost of running some classical message broadcasting algorithm on the communication graph, as part of the "Compute" step in Algorithm 1. Broadcasting is completed when all agents are informed, which is done in finite time and without any approximation error (see (Attiya and Welch, 2004; Lynch, 1996) for technical details).

3.3 Computational and Communication requirements

The computational and communication requirements are the main concerns when implementing a second-order optimization algorithm like ours. Indeed, the local computation of the Hessian matrices in $\boldsymbol{H}_i(\boldsymbol{\theta})$ involves $\mathcal{O}(d^3)$ operations for each agent i, and the distributed consensus demand the agents to exchange information on $\mathcal{O}(d^2)$ scalars. This might pose some obstacles to the application and the scalability of the proposed Algorithm 1, particularly under limited computational capabilities and strict communication bandwidth constraints. Nonetheless, we remark that the additional computational cost of each single update should be evaluated in relation to the reduced total number of iterations required to reach the optimal solution, which is most often problem specific.

This discussion gives us the opportunity to sketch alternative and simplified versions of Algorithm 1 (more sophisticated quasi-Newton alternatives are currently under study), where we select different structures for $\boldsymbol{H}_i(\boldsymbol{\theta})$ and obtain different procedures with different computational/communication requisites:

- with $\boldsymbol{H}_i(\boldsymbol{\theta}) \triangleq \mathrm{diag}\left[\nabla^2 J_i(\boldsymbol{\theta})\right]$ we obtain a Jacobi-like direction that reduces the computational and communication burden to $\mathcal{O}(d)$. This scheme still converges to the global optimum, but with a convergence rate generally slower than the Newton's method (Nocedal and Wright, 2006; Becker et al., 1988; Varagnolo et al., 2015).

- with $\boldsymbol{H}_i(\boldsymbol{\theta}) \triangleq \boldsymbol{I}$ we obtain a Gradient Ascent direction, which does not require any computation nor communication but achieves only linear convergence rate. This alternative can be motivated in those cases where the computation of the local second derivatives is expensive or where they are not continuous (because of the chosen policy parametrization).

4. Convergence Analysis

Convergence analysis for Algorithm 1 will be split into two-part: in the first, we provide convergence guarantees in an exact setting, where the agents have oracle access to the gradient/Hessian of their cost functions; then, we extend these results in an approximate setting, where the agents learn unbiased estimators for their gradient/Hessian. We refer to the Appendix for all the necessary proofs.

Without loss of generality, we assume the origin to be the minimizer[4] of the average objective function $J(\boldsymbol{\theta})$ in (1), *i.e.*, $\boldsymbol{\theta}^\star = 0$. We also make the following assumptions necessary for global convergence. If only local convergence is sought, as common practice when working on learning in non-convex games, the subsequent theorems have local validity. In fact, local differentiability and Lipschitzianity of the functions $J_i(\boldsymbol{\theta})$ are sufficient to guarantee that the following assumptions are locally valid.

Assumption 2 *The local functions J_i in (1) are of class $\mathcal{C}^3$. Moreover the global function $J(\boldsymbol{\theta})$ has bounded positive definite Hessian, i.e., $0 < mI \leq \boldsymbol{\nabla}^2 J(\boldsymbol{\theta}) \leq MI$ for some $m, M \in \mathbb{R}_+$ and $\forall \boldsymbol{\theta} \in \mathbb{R}^d$. Moreover, without loss of generality, we assume $J(\boldsymbol{\theta}^\star) = 0$, $m \leq 1$ and $M \geq 1$.*

Assumption 3 *The local functions J_i in (4) are such that, $\forall \boldsymbol{\theta}, \boldsymbol{\theta}', \boldsymbol{\theta}_i, \boldsymbol{\theta}_i' \in \mathbb{R}^d$, the following conditions hold*

$$\begin{cases} mI \leq N^{-1} \sum_{i \in \mathcal{N}} \boldsymbol{H}_i(\boldsymbol{\theta}_i) \leq MI \\ \left\| N^{-1} \sum_{i \in \mathcal{N}} \boldsymbol{g}_i(\boldsymbol{\theta}_i) \right\| \leq a_1 \\ \|\boldsymbol{g}_i(\boldsymbol{\theta}) - \boldsymbol{g}_i(\boldsymbol{\theta}')\| \leq a_2 \|\boldsymbol{\theta} - \boldsymbol{\theta}'\| \\ \|\boldsymbol{H}_i(\boldsymbol{\theta}) - \boldsymbol{H}_i(\boldsymbol{\theta}')\| \leq a_3 \|\boldsymbol{\theta} - \boldsymbol{\theta}'\| \\ \|\phi_{DN}(\boldsymbol{\theta}_{\mathcal{N}}) - \phi_{DN}(\boldsymbol{\theta}'_{\mathcal{N}})\| \leq a_4 \|\boldsymbol{\theta}_{\mathcal{N}} - \boldsymbol{\theta}'_{\mathcal{N}}\| \end{cases}$$

for positive scalars a_1, a_2, a_3, a_4, and with m, M from Assumption 2.

Exact learning The convergence of Algorithm 1, in the case of exact knowledge of gradients and Hessians, involves the stability properties of intermediate systems which are a successive approximation of the distributed Newton's dynamics. Starting from the global exponential stability of the centralized Newton's dynamics (2) (see (Nocedal and Wright, 2006), and Theorem 3 in (Varagnolo et al., 2015)), we first predicate the stability of the multi-agent Newton's update (5) and, consequently, of its distributed version in Algorithm 1.

Theorem 4 *Consider the dynamics defined by (5). Under Assumptions 2 and 3 there exists a positive scalar $\bar{\epsilon} > 0$ such that the update rule can be considered a stable forward-Euler discretization of a globally exponentially stable continuous dynamics. In particular, under Assumption 1, the origin is a globally exponentially stable point for dynamics (5), $\forall \epsilon \in (0, \bar{\epsilon})$.*

Theorem 5 *Consider the dynamics defined by Algorithm 1 with possibly nonzero initial conditions. Under Assumptions 2 and 3 there exists a positive scalar $\bar{\epsilon} > 0$ such that the algorithm can be considered a stable forward-Euler discretization of a globally exponentially stable continuous dynamics associated to (5). Thus, due to Theorem 4, the local estimates $\boldsymbol{\theta}_i(k)$ produced by the algorithm exponentially converge to the global optimizer, i.e., $\lim_{k \to \infty} \boldsymbol{\theta}_i(k) = \boldsymbol{\theta}^\star$, $\forall i \in \mathcal{N}, \epsilon \in (0, \bar{\epsilon})$ and $\boldsymbol{\theta}_i(0) \in \mathbb{R}^d$.*

Approximate learning When agents need to estimate their gradients and Hessians from potentially noisy observations of the environment, we integrate the convergence analysis by resorting to the technical machinery of stochastic approximation theory (Shapiro et al., 2014) and obtain guarantees analogous to that for exact learning but asymptotic in nature. Specifically, we assume that agents have access to unbiased sample average approximation estimators $\widehat{\boldsymbol{\nabla}} J_i(\boldsymbol{\theta}), \widehat{\boldsymbol{\nabla}}^2 J_i(\boldsymbol{\theta})$ of their exact counterpart in Theorem 1, which can be obtained following one of the approaches mentioned in Section 3.2 and the off-policy scheme therein (see also Appendix A for additional details). By the Law of Large Numbers we have that, under some regularity conditions as the availability of L i.i.d. samples (*i.e.*, trajectories τ collected from the MDP), $\widehat{\boldsymbol{\nabla}} J_i(\boldsymbol{\theta}), \widehat{\boldsymbol{\nabla}}^2 J_i(\boldsymbol{\theta})$ converge point wise with probability 1 to $\boldsymbol{\nabla} J_i(\boldsymbol{\theta}), \boldsymbol{\nabla}^2 J_i(\boldsymbol{\theta})$ as $L \to \infty$, and uniformly if additional mild conditions are considered (see Section 7.2.5 in (Shapiro et al., 2014)). As a consequence, we can still apply Theorems 4 and 5 also in the case of approximated gradients.

Furthermore, we also present a non-asymptotic analysis for Algorithm 1 and provide a more practical result that can be realistically used in real-life applications and under finite-samples conditions. Let us then consider some perturbation terms in the dynamics $\phi_{\text{DN}}(\boldsymbol{\theta}_{\mathcal{N}})$ that can be interpreted as numerical errors introduced by the estimators of agents' gradients and Hessians, as per the following assumption.

[4] For ease of reading and comparison with respect to the classical stability and optimization theory, we take here the point of view of a minimization problem, which can be obtained by simply switching the sign of the objective function $J(\theta)$.

Assumption 6 *The local finite-samples estimators $\widehat{\boldsymbol{g}}_i(\boldsymbol{\theta}(k)) = \boldsymbol{g}_i(\boldsymbol{\theta}(k)) + \boldsymbol{\xi}_{g,i}(k)$ and $\widehat{\boldsymbol{H}}_i(\boldsymbol{\theta}(k)) = \boldsymbol{H}_i(\boldsymbol{\theta}(k)) + \boldsymbol{\xi}_{H,i}(k)$ are subject to zero-mean, finite variance stochastic processes $\{\boldsymbol{\xi}_{g,i}(k)\}$ and $\{\boldsymbol{\xi}_{H,i}(k)\}$.*

The perturbations $\boldsymbol{\xi}_{\cdot,i}$ introduce a perturbed version $\widehat{\phi}_{\mathrm{DN}}(\boldsymbol{\theta}_{\mathcal{N}}, \{\boldsymbol{\xi}_{g,i}, \boldsymbol{\xi}_{H,i}\}_{i\in\mathcal{N}})$, with $\widehat{\phi}_{\mathrm{DN}}(\boldsymbol{\theta}_{\mathcal{N}}, \mathbf{0}) = \phi_{\mathrm{DN}}(\boldsymbol{\theta}_{\mathcal{N}})$, of the multi-agent dynamics (5) which is globally exponentially stable too, but with a different equilibrium point that is a function of the latter perturbations, as formalized in the next theorem.

Theorem 7 *Let $\boldsymbol{\xi}_g \in \mathbb{R}^d$, $\boldsymbol{\xi}_H \in \mathbb{R}^{d\times d}$, $\boldsymbol{\xi} = (\boldsymbol{\xi}_g, \boldsymbol{\xi}_H)$, $\boldsymbol{\xi}_{\mathcal{N}} = 1_N \otimes \boldsymbol{\xi}$, and consider the perturbed multi-agent Newton's dynamics $\widehat{\phi}_{DN}$. Then, under Assumptions 2 and 3, there exists a positive scalar $r > 0$ and a unique continuously differentiable function $\widehat{\boldsymbol{\theta}} : \{\boldsymbol{\xi} \mid \|\boldsymbol{\xi}\| \leq r\} \to \mathbb{R}^d$ such that $\widehat{\boldsymbol{\theta}}(\mathbf{0}) = \mathbf{0}$ and which is an equilibrium point for $\widehat{\phi}_{DN}$. Specifically we have*

$$\widehat{\boldsymbol{\theta}}(\boldsymbol{\xi}) = \left(\frac{1}{N}\sum_{i\in\mathcal{N}} \boldsymbol{H}_i(\widehat{\boldsymbol{\theta}}(\boldsymbol{\xi})) + \boldsymbol{\xi}_H\right)^{-1} \left(\frac{1}{N}\sum_{i\in\mathcal{N}} \boldsymbol{g}_i(\widehat{\boldsymbol{\theta}}(\boldsymbol{\xi})) + \boldsymbol{\xi}_g - \boldsymbol{\xi}_H \boldsymbol{\theta}^\star\right). \tag{10}$$

Moreover, if we define a translated version of the perturbed multi-agent Newton's dynamics as $\widehat{\phi}'_{DN}(\boldsymbol{\theta}_{\mathcal{N}}, \boldsymbol{\xi}) \triangleq \widehat{\phi}_{DN}(\boldsymbol{\theta}_{\mathcal{N}} + \mathbf{1}_N \otimes \widehat{\boldsymbol{\theta}}(\boldsymbol{\xi}), \boldsymbol{\xi}_{\mathcal{N}})$, these have the origin as a new equilibrium point.
Under Assumptions 2, 3 and global Lipschitz conditions of $\widehat{\phi}'_{DN}$, such that $\left\|\widehat{\phi}'_{DN}(\boldsymbol{\theta}_{\mathcal{N}}, \boldsymbol{\xi}) - \widehat{\phi}'_{DN}(\boldsymbol{\theta}_{\mathcal{N}}, \mathbf{0})\right\| \leq k\|\boldsymbol{\xi}\|\|\boldsymbol{\theta}_{\mathcal{N}}\|$, with positive scalar k and $\|\boldsymbol{\xi}\| \leq r$, the origin is a globally exponentially stable equilibrium point.

5. Related work

Lately, several important advances have been made in the context of cooperative MARL algorithms (OroojlooyJadid and Hajinezhad, 2019; Zhang et al., 2019a; Lee et al., 2020), where problem (4) is solved with only local computation and communication with neighboring agents. However, our algorithm retains two unique characteristics when compared to the other approaches in the literature, namely *i)* the application of second-order Newton's search direction and *ii)* the direct consensus on the global optimal policy, without any auxiliary value–function representation. To the best of our knowledge, all the policy search approaches in the literature of cooperative MARL are built on first-order policy-gradient approaches, and the global optimal policy is indirectly achieved by first reaching consensus on the optimal global value–function, required to capture the agents' mutual dependencies in the environment. The combination of these two aspects has made Actor-Critic (AC) methods and their extensions quite popular in the MARL literature, where the Critic evaluates the current combined local policies, and the Actors use the feedback from the Critic to improve the policy functions. Recently introduced fully decentralized gradient-based AC methods are the most relevant works to the method presented here.

In (Zhang et al., 2018b,a) the authors let the agents estimate the global value–function in a distributed way through a consensus mechanism and perform the actor step individually, without the need to know or infer other agents' policies. In (Zhang and Zavlanos, 2019) the authors assume that the agents can have different tasks (thus different state and action spaces), and different local value–functions associated with them: the agents keep local estimates of the global policy function parameters and update them via a consensus step, while the local value-function estimates are instead updated independently. Zhang et al. (Zhang and Zavlanos, 2020) propose an actor-only policy optimization method that allows the agents to compute local policy gradients based on partial states and actions and to use local estimates of the global return that can be obtained using consensus. Suttle et al. (Suttle et al., 2020) investigate an off-policy AC approach based on the consensus-based version of a particular temporal difference algorithm for the critic update. Wai et al. (Wai et al., 2018) focuses on the policy evaluation step within an AC scheme and proposes a Fenchel primal-dual reformulation of the mean squared projected Bellman error minimization problem, with a consensus step on the global value-function parameters.

From the theoretical point of view, all the above algorithms come with convergence guarantees, which are almost invariably related to the case of a linear approximation for the value function, and based on the two-time scale technique (Borkar, 2009). The only notable differences are in (Zhang and Zavlanos, 2020) which employs biased gradient estimates, and hence the convergence is proved to a neighborhood of the global optimal policy (similarly to our analysis in the non-exact and non-asymptotic case), and (Wai et al., 2018) offers finite-time convergence on decentralized convex-concave saddle-point problems.

6. Experiments

This section is devoted to the empirical evaluation of the proposed algorithm. We provide results in two domains: a famous matrix game, the Prisoner's Dilemma, and a continuous Gridworld environment (Lowe et al., 2017). Our goal is to compare the convergence speed of Algorithm 1 (DN, in this section) against first-order approaches (DG, hereafter), which can be implemented by setting $\boldsymbol{H}_i(\boldsymbol{\theta}) = \boldsymbol{I}$ (as described in Section 3.3). Experiments were run on a 1,4 GHz Intel Core i5 quad-core with 8 GB RAM.

The Prisoner's Dilemma is a bimatrix game where two agents have to decide between cooperating and defecting. If both agents cooperate, they gain a reward of -1, if both defect of -2, and if one cooperates and the other defects, they gain respectively -3 and 0. The Nash equilibrium for this game is (defect, defect); however, the agents can learn to cooperate to achieve the optimal expected return in the cooperative scenario. In this experiment, we employed a Boltzmann parametrization for the agents' policies and set the parameters of NC as $\epsilon = 10^{-5}$, $m = 10^{-7}$, and for GC as $\epsilon = 10^{-1}$. First, we illustrate in Figure 1 the performances of the DN algorithm in the exact learning setting, where the gradients and the Hessian are computed without any approximation error. The results, randomized over 30 repetitions with different consensus matrices, show an evident and substantial increase in the speed convergence of DN, and that the algorithm is also able to reach the global optimal value of $J(\boldsymbol{\theta}^{\star}) = -1$.

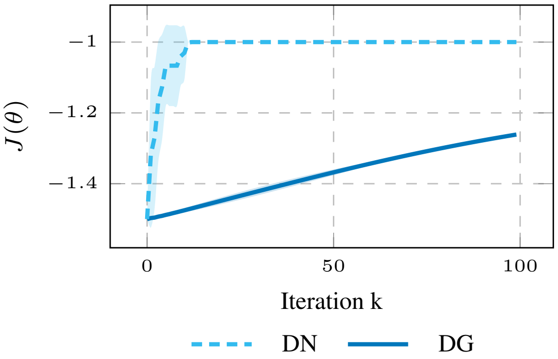

Figure 1: Exact learning in Prisoner's Dilemma. Global average long-term return $J(\boldsymbol{\theta})$ evolution with DN ($\epsilon = 10^{-5}, m = 10^-7$) and DG ($\epsilon = 10^{-1}$). Results are randomized over 30 casual consensus matrices, and showed with $\%98$ c.i.

The same positive outcome was confirmed in the approximate learning case, where gradients and Hessians are estimated from the interaction with the environment. Here the results were obtained with a fixed consensus matrix and randomized over 30 repetitions for the stochastic estimations of the gradients and Hessians. In the plot on the right of Figure 2 we can see how DN outperforms DG when IS technique is properly applied to obtain unbiased estimations, as discussed in Section 3.2. On the contrary, the plot on the left shows the negative impact of using wrongly estimated directions, especially for the DN algorithm. Another interesting result of Figure 2 is the empirical evidence of Theorem 7, which implies that, with numerical approximation errors in the finite-sample scenario, the algorithm converges to a slightly shifted global optimum (see equation 10) and is not able to reach the optimal value -1.

The continuous Gridworld is a planar cooperative environment where two agents are initialized in the two opposite lower corners and have to reach the opposite upper corner, with a positive reward when the agent gets to his goal. The two agents have to keep a distance from each other of no less than 0.5, and get stuck otherwise so that they need to coordinate to reach their goals safely. Agents' policies were parametrized as Gaussian, linear in a set of 18 radial basis functions, which generate the angle for the step direction. Figure 3 reports a comparison in the approximate learning settings (with IS properly utilized), randomized over 30 repetitions, where 100 trajectories of 15 steps were used to estimate the gradients and Hessians. Coherently with the previous domain, the DN algorithm exceeds the DG in the convergence speed and reaches the local optimum in very few iterations of the algorithm.

7. Conclusions and Future Directions

Albeit the recent growth of approaches in multi-agent RL and the cornerstone position of the Newton algorithm in the optimization literature, to the best of authors' knowledge, there are currently no works that take advantage of the good

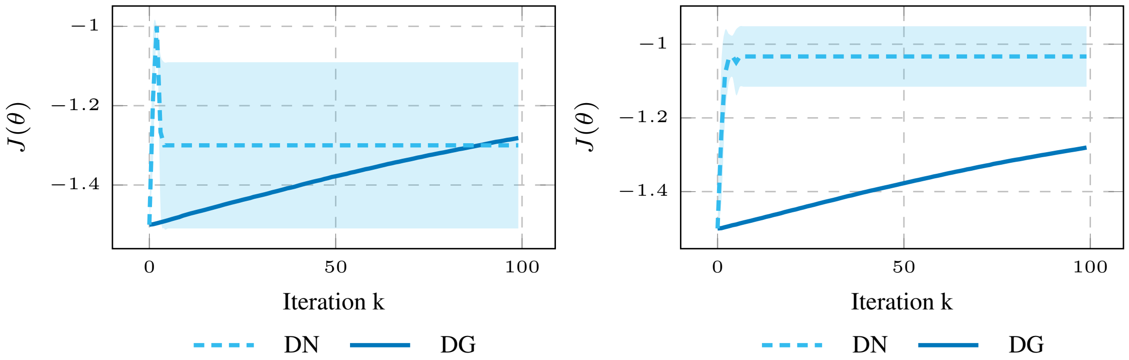

Figure 2: Approximate learning in Prisoner's Dilemma. Comparison of global average long-term return $J(\boldsymbol{\theta})$ evolution with DN ($\epsilon = 10^{-5}, m = 10^{-}7$) and DG ($\epsilon = 10^{-1}$) under approximate learning without IS (left) and with IS (right). Results are randomized over 30 repetitions, and showed with $\%98$ c.i..

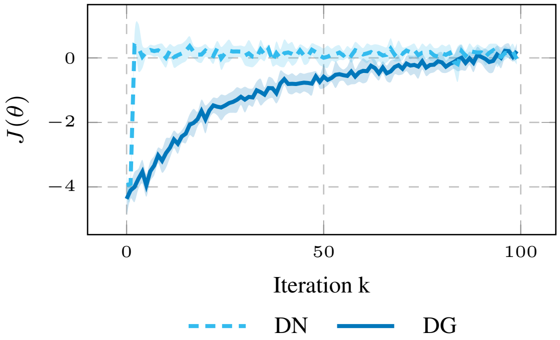

Figure 3: Approximate learning in continuous Gridworld. Global average long-term return $J(\boldsymbol{\theta})$ evolution with DN ($\epsilon = 10^{-5}, m = 10^{-6}$) and DG ($\epsilon = 10^{-3}$). Results are randomized over 10 repetitions, and showed with $\%98$ c.i..

Newton's method properties to solve collaborative MARL problems. In this paper, we proposed a novel Newton-based policy search algorithm for the networked MARL cooperative settings. The algorithm does not rely on the representation of the global value function and directly seeks the optimal parameters of the global policy. In our scheme, each agent locally computes the gradient and Hessian of the global objective function, which, by means of a consensus algorithm, are properly combined to infer the global Newton's direction used by the agents to update the policy parameters. The estimated directions are learned by off-policy interactions with the environment and do not require the knowledge of the underlying model. The described algorithm comes with theoretical guarantees of convergence, which remarkably hold also in the finite-sample case, and was numerically proved to outperform simpler first-order gradient approaches. Among possible future research paths, we intend to investigate the properties and the theoretical guarantees of different quasi-Newton directions within the distributed MARL setting so as to trade the computational complexity with convergence speed, and hence increase the scalability potentials of the algorithm. Another possible extension concerns the applicability of the proposed algorithm to the case of time-varying communication networks or asynchronous communication protocols.

Acknowledgments

This work was partially supported by the PON - Programma Operativo Nazionale Ricerca e Innovazione 2014-2020, AIM1880573 Smart, Secure and Inclusive Communities.

References

Hagit Attiya and Jennifer Welch. *Distributed computing: fundamentals, simulations, and advanced topics*, volume 19. John Wiley & Sons, 2004.

Jonathan Baxter and Peter L Bartlett. Infinite-horizon policy-gradient estimation. *Journal of Artificial Intelligence Research*, 15:319–350, 2001.

Sue Becker, Yann Le Cun, et al. Improving the convergence of back-propagation learning with second order methods. In *Proceedings of the 1988 connectionist models summer school*, pages 29–37, 1988.

Vivek S Borkar. *Stochastic approximation: a dynamical systems viewpoint*, volume 48. Springer, 2009.

Peter Corke, Ron Peterson, and Daniela Rus. Networked robots: Flying robot navigation using a sensor net. In *Robotics research. The eleventh international symposium*, pages 234–243. Springer, 2005.

Jorge Cortés, Sonia Martınez, Timur Karatas, and Francesco Bullo. Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation*, 20(2), 2004.

J Alexander Fax and Richard M Murray. Information flow and cooperative control of vehicle formations. *IEEE transactions on automatic control*, 49(9):1465–1476, 2004.

Jakob N Foerster, Richard Y Chen, Maruan Al-Shedivat, Shimon Whiteson, Pieter Abbeel, and Igor Mordatch. Learning with opponent-learning awareness. *arXiv preprint arXiv:1709.04326*, 2017.

Thomas Furmston and David Barber. A unifying perspective of parametric policy search methods for markov decision processes. Neural Information Processing Systems Foundation, 2012.

Thomas Furmston, Guy Lever, and David Barber. Approximate newton methods for policy search in markov decision processes. *Journal of Machine Learning Research*, 17, 2016.

Hassan K Khalil and Jessy W Grizzle. *Nonlinear systems*, volume 3. Prentice hall Upper Saddle River, NJ, 2002.

Solmaz S Kia, Bryan Van Scoy, Jorge Cortes, Randy A Freeman, Kevin M Lynch, and Sonia Martinez. Tutorial on dynamic average consensus: The problem, its applications, and the algorithms. *IEEE Control Systems Magazine*, 39(3):40–72, 2019.

Jens Kober and Jan Peters. Policy search for motor primitives in robotics. In *Learning Motor Skills*, pages 83–117. Springer, 2014.

Vijay R Konda and John N Tsitsiklis. Actor-critic algorithms. In *Advances in neural information processing systems*, pages 1008–1014. Citeseer, 2000.

Steven G Krantz and Harold R Parks. *The implicit function theorem: history, theory, and applications*. Springer Science & Business Media, 2012.

Donghwan Lee, Niao He, Parameswaran Kamalaruban, and Volkan Cevher. Optimization for reinforcement learning: From a single agent to cooperative agents. *IEEE Signal Processing Magazine*, 37(3):123–135, 2020.

Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *Neural Information Processing Systems (NIPS)*, 2017.

Nancy A Lynch. *Distributed algorithms*. Elsevier, 1996.

Giorgio Manganini, Matteo Pirotta, Marcello Restelli, and Luca Bascetta. Following newton direction in policy gradient with parameter exploration. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2015.

Eric Mazumdar, Lillian J Ratliff, and S Shankar Sastry. On gradient-based learning in continuous games. *SIAM Journal on Mathematics of Data Science*, 2(1):103–131, 2020.

Angelia Nedić and Ji Liu. Distributed optimization for control. *Annual Review of Control, Robotics, and Autonomous Systems*, 1:77–103, 2018.

Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

Afshin OroojlooyJadid and Davood Hajinezhad. A review of cooperative multi-agent deep reinforcement learning. *arXiv preprint arXiv:1908.03963*, 2019.

Jan Peters and Stefan Schaal. Reinforcement learning of motor skills with policy gradients. *Neural networks*, 21(4): 682–697, 2008.

Giorgia Ramponi and Marcello Restelli. Newton optimization on helmholtz decomposition for continuous games.

Alexander Shapiro, Darinka Dentcheva, and Andrzej Ruszczyński. *Lectures on stochastic programming: modeling and theory*. SIAM, 2014.

Zebang Shen, Alejandro Ribeiro, Hamed Hassani, Hui Qian, and Chao Mi. Hessian aided policy gradient. In *International Conference on Machine Learning*, pages 5729–5738. PMLR, 2019.

Wesley Suttle, Zhuoran Yang, Kaiqing Zhang, Zhaoran Wang, Tamer Başar, and Ji Liu. A multi-agent off-policy actor-critic algorithm for distributed reinforcement learning. *IFAC-PapersOnLine*, 53(2):1549–1554, 2020.

Richard S Sutton, David A McAllester, Satinder P Singh, Yishay Mansour, et al. Policy gradient methods for reinforcement learning with function approximation. In *NIPs*, volume 99, pages 1057–1063. Citeseer, 1999.

Philip S Thomas and Emma Brunskill. Policy gradient methods for reinforcement learning with function approximation and action-dependent baselines. *arXiv preprint arXiv:1706.06643*, 2017.

Damiano Varagnolo, Filippo Zanella, Angelo Cenedese, Gianluigi Pillonetto, and Luca Schenato. Newton-raphson consensus for distributed convex optimization. *IEEE Transactions on Automatic Control*, 61(4):994–1009, 2015.

Hoi-To Wai, Zhuoran Yang, Zhaoran Wang, and Mingyi Hong. Multi-agent reinforcement learning via double averaging primal-dual optimization. *arXiv preprint arXiv:1806.00877*, 2018.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

Kaiqing Zhang, Zhuoran Yang, and Tamer Basar. Networked multi-agent reinforcement learning in continuous spaces. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 2771–2776. IEEE, 2018a.

Kaiqing Zhang, Zhuoran Yang, Han Liu, Tong Zhang, and Tamer Basar. Fully decentralized multi-agent reinforcement learning with networked agents. In *International Conference on Machine Learning*, pages 5872–5881. PMLR, 2018b.

Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. Decentralized multi-agent reinforcement learning with networked agents: Recent advances. *arXiv preprint arXiv:1912.03821*, 2019a.

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

Yan Zhang and Michael M Zavlanos. Distributed off-policy actor-critic reinforcement learning with policy consensus. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 4674–4679. IEEE, 2019.

Yan Zhang and Michael M Zavlanos. Cooperative multi-agent reinforcement learning with partial observations. *arXiv preprint arXiv:2006.10822*, 2020.

Minghui Zhu and Sonia Martínez. Discrete-time dynamic average consensus. *Automatica*, 46(2):322–329, 2010.

Appendix A. Appendix: Gradient and Hessian Estimators

We present a brief but formal overview of the two most widespread gradient estimators, namely (Williams, 1992) and (Baxter and Bartlett, 2001), and show how we apply them also for the Hessian estimation. For clarity of exposition, we introduce first the on-policy setting, and then we show the off-policy version used in our work (see Section 3.2). The exact gradient and Hessian formulations are given in Theorem 1. All the derivations are from the point of view of the i-th agent.

On-policy setting Let $\mathcal{D} = \{\tau^l\}_{l=1}^L$ be a batch of trajectories $\tau^l = \{s_t^l, a_t^l\}_{t=0}^T$ (the subscripts denote the time step, superscripts denote the trajectory) collected using the policy $\pi_{\boldsymbol{\theta}_i}$. The REINFORCE gradient estimator provides a simple, unbiased way of estimating the gradient:

$$\widehat{\boldsymbol{\nabla}} J_i(\boldsymbol{\theta}_i) = \frac{1}{L} \sum_{l=1}^{L} \left(\sum_{t=0}^{T} \boldsymbol{\nabla} \log \pi_{\boldsymbol{\theta}_i}(a_t^l | s_t^l) \right) \left(\sum_{t=0}^{T} \gamma^t \mathcal{R}_i(s_t^l, a_t^l) - b(s_t^l, a_t^l) \right),$$

where $b : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is baseline function that may be used for reducing the variance of the estimation (see (Williams, 1992; Peters and Schaal, 2008; Thomas and Brunskill, 2017)). Similarly, we derive the Hessian estimator as

$$\widehat{\boldsymbol{\nabla}}^2 J_i(\boldsymbol{\theta}_i) = \frac{1}{L} \sum_{l=1}^{L} \left(\sum_{t=0}^{T} \boldsymbol{\nabla} \log \pi_{\boldsymbol{\theta}_i}(a_t^l | s_t^l) \boldsymbol{\nabla} \log \pi_{\boldsymbol{\theta}_i}(a_t^l | s_t^l)^\top + \boldsymbol{\nabla}^2 \log \pi_{\boldsymbol{\theta}_i}(a_t^l | s_t^l) \right) \left(\sum_{t=0}^{T} \gamma^t \mathcal{R}_i(a_t^l, s_t^l) - b(a_t^l, s_t^l) \right).$$

The G(PO)MDP gradient estimator is a more efficient implementation of the REINFORCE algorithm, and it is subject to less variance. Whereas the latter ignores that the reward at time t does not depend on the action performed after time t, the former explicitly takes into account the causality of rewards in the estimation procedure. Thus we obtain for the gradient

$$\widehat{\boldsymbol{\nabla}} J_i(\boldsymbol{\theta}_i) = \frac{1}{L} \sum_{l=1}^{L} \sum_{t=0}^{T} \left(\sum_{k=0}^{t} \boldsymbol{\nabla} \log p_{\boldsymbol{\theta}_i}(a_k^l | s_k^l) \right) \left(\gamma^t \mathcal{R}_i(s_t^l, a_t^l) - b(s_t^l, a_t^l) \right),$$

and for the Hessian

$$\widehat{\boldsymbol{\nabla}}^2 J_i(\boldsymbol{\theta}_i) = \frac{1}{L} \sum_{l=1}^{L} \sum_{t=0}^{T} \left(\sum_{k=0}^{t} \boldsymbol{\nabla} \log \pi_{\boldsymbol{\theta}_i}(a_k^l | s_k^l) \boldsymbol{\nabla} \log \pi_{\boldsymbol{\theta}_i}(a_k^l | s_k^l)^\top + \boldsymbol{\nabla}^2 \log \pi_{\boldsymbol{\theta}_i}(a_k^l | s_k^l) \right) \left(\gamma^t \mathcal{R}_i(a_t^l, s_t^l) - b(a_t^l, s_t^l) \right).$$

Off-policy setting In the off-policy setting we have two policies involved, $\pi_{\boldsymbol{\mu}}$ (the behavioural one, see Section 3.2 for its definition in our specific context of Cooperative MARL) and $\pi_{\boldsymbol{\theta}_i}$ (the target one). The first one, as explained in Section 3.2, is used to interact with the environment, whereas the second one is used to evaluate the agent performance and it is improved in each iteration of the algorithm. When we would like to estimate the performance of the target policy $\pi_{\boldsymbol{\theta}_i}$, but we have samples collected using policy $\pi_{\boldsymbol{\mu}}$, we can use importance sampling to correct the shift in the distribution and obtain an unbiased estimate of $J_i(\boldsymbol{\theta}_i)$ as

$$J_i(\boldsymbol{\theta}_i) = \mathop{\mathbb{E}}_{\tau \sim \pi_{\boldsymbol{\theta}_i}, \mathcal{P}} [\mathcal{R}_i(\tau)] = \mathop{\mathbb{E}}_{\tau \sim \pi_{\boldsymbol{\mu}}, \mathcal{P}} \left[\frac{p(\tau | \pi_{\boldsymbol{\theta}_i})}{p(\tau | \pi_{\boldsymbol{\mu}})} \mathcal{R}_i(\tau) \right]. \tag{11}$$

If we compactly introduce the correction importance weight $\omega_{\boldsymbol{\theta}_i/\boldsymbol{\mu}}(\tau) = \frac{p(\tau|\pi_{\boldsymbol{\theta}_i})}{p(\tau|\pi_{\boldsymbol{\mu}})}$, we can write the off-policy versions of the gradient and Hessian (8) as

$$\boldsymbol{\nabla} J_i(\boldsymbol{\theta}_i) = \mathop{\mathbb{E}}_{\tau \sim \pi_{\boldsymbol{\mu}}, \mathcal{P}} \left[\omega_{\boldsymbol{\theta}_i/\boldsymbol{\mu}}(\tau) \mathcal{R}_i(\tau) \boldsymbol{\nabla} \log p(\tau | \boldsymbol{\theta}_i) \right],$$

$$\boldsymbol{\nabla}^2 J_i(\boldsymbol{\theta}_i) = \mathop{\mathbb{E}}_{\tau \sim \pi_{\boldsymbol{\mu}}, \mathcal{P}} \left[\omega_{\boldsymbol{\theta}_i/\boldsymbol{\mu}}(\tau) \mathcal{R}_i(\tau) \left(\boldsymbol{\nabla} \log p(\tau | \boldsymbol{\theta}_i) \boldsymbol{\nabla} \log p(\tau | \boldsymbol{\theta}_i)^\top + \boldsymbol{\nabla}^2 \log p(\tau | \boldsymbol{\theta}_i) \right) \right].$$

Let us assume then to have a batch $\mathcal{D} = \{\tau^l\}_{l=1}^L$ of trajectories $\tau^l = \{s_t^l, a_t^l\}_{t=0}^T$ collected using the policy $\pi_{\boldsymbol{\mu}}$. The off-policy versions of REINFORCE are easily obtained by taking the empirical average of the previous functions:

$$\widehat{\boldsymbol{\nabla}} J_i(\boldsymbol{\theta}_i) = \frac{1}{L} \sum_{l=1}^{L} \omega_{\boldsymbol{\theta}_i/\boldsymbol{\mu}}(\tau^l) \left(\sum_{t=0}^{T} \boldsymbol{\nabla} \log \pi_{\boldsymbol{\theta}_i}(a_t^l | s_t^l) \right) \left(\sum_{t=0}^{T} \gamma^t \mathcal{R}_i(s_t^l, a_t^l) - b(s_t^l, a_t^l) \right),$$

$$\widehat{\boldsymbol{\nabla}}^2 J_i(\boldsymbol{\theta}_i) = \frac{1}{L} \sum_{l=1}^{L} \omega_{\boldsymbol{\theta}_i/\boldsymbol{\mu}}(\tau^l) \left(\sum_{t=0}^{T} \boldsymbol{\nabla} \log \pi_{\boldsymbol{\theta}_i}(a_t^l | s_t^l) \boldsymbol{\nabla} \log \pi_{\boldsymbol{\theta}_i}(a_t^l | s_t^l)^\top + \boldsymbol{\nabla}^2 \log \pi_{\boldsymbol{\theta}_i}(a_t^l | s_t^l) \right) \left(\sum_{t=0}^{T} \gamma^t \mathcal{R}_i(a_t^l, s_t^l) - b(a_t^l, s_t^l) \right).$$

Finally, the G(PO)MDP off-policy estimators can be computed as follows:

$$\widehat{\boldsymbol{\nabla}} J_i(\boldsymbol{\theta}_i) = \frac{1}{L}\sum_{l=1}^{L}\sum_{t=0}^{T}\left(\sum_{k=0}^{t} \boldsymbol{\nabla}\log p_{\boldsymbol{\theta}_i}(a_k^l|s_k^l)\right)\big(\gamma^t \mathcal{R}_i(s_t^l, a_t^l) - b(s_t^l, a_t^l)\big)\omega_{\boldsymbol{\theta}_i/\boldsymbol{\mu}}(\tau_{0:t}^l),$$

$$\widehat{\boldsymbol{\nabla}}^2 J_i(\boldsymbol{\theta}_i) = \frac{1}{L}\sum_{l=1}^{L}\sum_{t=0}^{T}\left(\sum_{k=0}^{t} \boldsymbol{\nabla}\log \pi_{\boldsymbol{\theta}_i}(a_k^l|s_k^l)\boldsymbol{\nabla}\log \pi_{\boldsymbol{\theta}_i}(a_k^l|s_k^l)^\top + \boldsymbol{\nabla}^2\log \pi_{\boldsymbol{\theta}_i}(a_k^l|s_k^l)\right)\big(\gamma^t \mathcal{R}_i(a_t^l, s_t^l) - b(a_t^l, s_t^l)\big)\omega_{\boldsymbol{\theta}_i/\boldsymbol{\mu}}(\tau_{0:t}^l),$$

where we indicated with $\omega_{\boldsymbol{\theta}_i/\boldsymbol{\mu}}(\tau_{0:t})$ the truncated importance weight at time t, that is $\omega_{\boldsymbol{\theta}_i/\boldsymbol{\mu}}(\tau_{0:t}) = \prod_{t'=0}^{t} \frac{\pi_{\boldsymbol{\theta}_i}(a_{t'}|s_{t'})}{\pi_{\boldsymbol{\mu}}(a_{t'}|s_{t'})}$.

Appendix B. Appendix: Proofs

In this appendix, we report the proofs and derivations of the results presented in the main paper.

Proof of Theorem 1

Proof First we derive the gradient of the cost function $J_i(\boldsymbol{\theta})$ w.r.t. $\boldsymbol{\theta}$:

$$\begin{aligned}
\boldsymbol{\nabla} J_i(\boldsymbol{\theta}) &= \int_{\mathcal{T}} \boldsymbol{\nabla} p(\tau|\pi_{\boldsymbol{\theta}})\mathcal{R}_i(\tau)\mathrm{d}\tau \\
&\overset{(a)}{=} \int_{\mathcal{T}} p(\tau|\pi_{\boldsymbol{\theta}})\boldsymbol{\nabla}\log p(\tau|\pi_{\boldsymbol{\theta}})\mathcal{R}_i(\tau)\mathrm{d}\tau \\
&= \mathop{\mathbb{E}}_{\tau\sim\pi_{\boldsymbol{\theta}},\mathcal{P}}[\boldsymbol{\nabla}\log p(\tau|\pi_{\boldsymbol{\theta}})\mathcal{R}_i(\tau)]
\end{aligned}$$

where in (a) we used the log derivative trick $\boldsymbol{\nabla}_x f(x) = f(x)\boldsymbol{\nabla}_x \log f(x)$.

The derivation of the Hessian follows from the application of the second derivative to the gradient vector:

$$\begin{aligned}
\boldsymbol{\nabla}^2 J_i(\boldsymbol{\theta}) &= \int_{\mathcal{T}} \boldsymbol{\nabla}(p(\tau|\pi_{\boldsymbol{\theta}})\boldsymbol{\nabla}\log p(\tau|\pi_{\boldsymbol{\theta}}))\mathcal{R}_i(\tau)\mathrm{d}\tau \\
&= \int_{\mathcal{T}} \big(\boldsymbol{\nabla} p(\tau|\pi_{\boldsymbol{\theta}})\boldsymbol{\nabla}\log p(\tau|\pi_{\boldsymbol{\theta}}) + p(\tau|\pi_{\boldsymbol{\theta}})\boldsymbol{\nabla}^2\log p(\tau|\pi_{\boldsymbol{\theta}})\big)\mathcal{R}_i(\tau)\mathrm{d}\tau \\
&\overset{(a)}{=} \int_{\mathcal{T}} \big(p(\tau|\pi_{\boldsymbol{\theta}})\boldsymbol{\nabla}\log p(\tau|\pi_{\boldsymbol{\theta}})\boldsymbol{\nabla}\log p(\tau|\pi_{\boldsymbol{\theta}})^\top + p(\tau|\pi_{\boldsymbol{\theta}})\boldsymbol{\nabla}^2\log p(\tau|\pi_{\boldsymbol{\theta}})\big)\mathcal{R}_i(\tau)\mathrm{d}\tau \\
&= \mathop{\mathbb{E}}_{\tau\sim\pi_{\boldsymbol{\theta}},\mathcal{P}}\big[\big(\boldsymbol{\nabla}\log p(\tau|\pi_{\boldsymbol{\theta}})\boldsymbol{\nabla}\log p(\tau|\pi_{\boldsymbol{\theta}})^\top + \boldsymbol{\nabla}^2\log p(\tau|\pi_{\boldsymbol{\theta}})\big)\mathcal{R}_i(\tau)\big]
\end{aligned}$$

where in (a) we used again the same log derivative trick. ■

Proof of Theorem 4

Let us consider the continuous-time and aggregated form of equation (5), that reads

$$\dot{\boldsymbol{\theta}}_{\mathcal{N}} = -\boldsymbol{\theta}_{\mathcal{N}} + \mathbf{1}_N \otimes \phi_{\text{DN}}(\boldsymbol{\theta}_{\mathcal{N}}) \triangleq \Phi(\boldsymbol{\theta}_{\mathcal{N}}). \tag{12}$$

The logical flow of the demonstration profs first that, under the existence of a proper Lyapunov function, the origin is globally exponentially stable for the previous system (12), and then that the forward-Euler discretization (5) of its dynamics are globally exponentially stable.

Let $V(\cdot) : \mathbb{R}^{dN} \to \mathbb{R}$ be a Lyapunov function for (12):

$$V(\boldsymbol{\theta}_{\mathcal{N}}) \triangleq J(\bar{\boldsymbol{\theta}}) + \frac{1}{2}\big\|\boldsymbol{\theta}_{\mathcal{N}} - 1_N \otimes \bar{\boldsymbol{\theta}}\big\|^2. \tag{13}$$

It is easily noted that $V(\mathbf{0}) = 0$ and $V(\boldsymbol{\theta}_{\mathcal{N}}) > 0$ for $\boldsymbol{\theta}_{\mathcal{N}} \neq \mathbf{0}$, which follow from the fact that (by Assumption 2) $J(\mathbf{0}) = 0$ and $J(\bar{\boldsymbol{\theta}}) > 0$ for $\bar{\boldsymbol{\theta}} \neq 0$. Thanks to Theorem 6 in (Varagnolo et al., 2015), we can assume also that there exist positive scalars b_1, b_2, b_3, b_4 such that, $\forall \boldsymbol{\theta}_{\mathcal{N}} \in \mathbb{R}^{dN}$

$$\begin{cases} b_1 I \leq \boldsymbol{\nabla}^2 V(\boldsymbol{\theta}_{\mathcal{N}}) \leq b_2 I \\ \frac{\partial V}{\partial \boldsymbol{\theta}_N} \Phi(\boldsymbol{\theta}_{\mathcal{N}}) \leq -b_3 \|\boldsymbol{\theta}_{\mathcal{N}}\|^2 \\ \|\Phi(\boldsymbol{\theta}_{\mathcal{N}})\| \leq b_4 \|\boldsymbol{\theta}_{\mathcal{N}}\|. \end{cases}$$

Integrating the first condition twice implies $\frac{1}{2}\|\boldsymbol{\theta}_{\mathcal{N}}\|^2 \leq V(\boldsymbol{\theta}_{\mathcal{N}}) \leq \frac{1}{2}\|\boldsymbol{\theta}_{\mathcal{N}}\|^2$ that, jointly with the second condition, guarantees global exponential stability for (12) (Theorem 4.10 in (Khalil and Grizzle, 2002)).

The third conditions is instead used to proof the stability of the forward-Euler discretization of (12), *i.e.*, equation (5), for which we need to show that $V(\boldsymbol{\theta}_{\mathcal{N}}(k+1)) - V(\boldsymbol{\theta}_{\mathcal{N}}(k)) \leq -b\|\boldsymbol{\theta}_{\mathcal{N}}\|^2$ for some positive scalar b. To this end, expand $V(\boldsymbol{\theta}_{\mathcal{N}}(k+1))$ with a second order Taylor expansion around $\boldsymbol{\theta}_{\mathcal{N}}(k)$, to obtain

$$V(\boldsymbol{\theta}_{\mathcal{N}} + \epsilon\Phi(\boldsymbol{\theta}_{\mathcal{N}})) = V(\boldsymbol{\theta}_{\mathcal{N}}) + \epsilon \frac{\partial V}{\partial \boldsymbol{\theta}_N}\Phi(\boldsymbol{\theta}_{\mathcal{N}}) + \frac{1}{2}\epsilon^2 \Phi(\boldsymbol{\theta}_{\mathcal{N}})^\top \boldsymbol{\nabla}^2 V(\boldsymbol{\theta}_{\mathcal{N}} + \epsilon'\Phi(\boldsymbol{\theta}_{\mathcal{N}}))\Phi(\boldsymbol{\theta}_{\mathcal{N}})$$

with $\epsilon' \in [0, \epsilon]$. Using again the above inequalities on the Lyapunov function, we derive

$$V(\boldsymbol{\theta}_{\mathcal{N}}(k+1)) - V(\boldsymbol{\theta}_{\mathcal{N}}(k)) \leq -\epsilon b_3 \|\boldsymbol{\theta}_{\mathcal{N}}(k)\|^2 + \frac{1}{2}\epsilon^2 b_2 b_4^2 \|\boldsymbol{\theta}_{\mathcal{N}}(k)\|^2 = -\epsilon\left(b_3 - \epsilon\frac{1}{2}b_2 b_4^2\right)\|\boldsymbol{\theta}_{\mathcal{N}}(k)\|^2.$$

It follows that, for all $\epsilon < \bar{\epsilon} = 2b_3/b_2 b_4^2$ the origin is globally exponentially stable.

Proof of Theorem 5

The proof can be found in Theorem 12 of (Varagnolo et al., 2015).

Proof of Theorem 7

Let's consider the following perturbed version of the continuous-time counterpart $\dot{\boldsymbol{\theta}}_{\mathcal{N}} = -\boldsymbol{\theta}_{\mathcal{N}} + \mathbf{1}_N \otimes \phi_{\text{DN}}(\boldsymbol{\theta}_{\mathcal{N}})$ of the multi-agent independent Newton's dynamics (5),

$$\begin{aligned} \dot{\boldsymbol{\theta}}_{\mathcal{N}} &= -\boldsymbol{\theta}_{\mathcal{N}} + \frac{\boldsymbol{\xi}_{\boldsymbol{g},\boldsymbol{\mathcal{N}}} + \mathbf{1}_N \otimes \left(\frac{1}{N}\sum_{i\in\mathcal{N}} \boldsymbol{g}_i(\boldsymbol{\theta}_i) + \frac{1}{N}\sum_{i\in\mathcal{N}} \boldsymbol{H}_i(\boldsymbol{\theta}_i)\boldsymbol{\theta}^\star\right)}{\left[\boldsymbol{\xi}_{\boldsymbol{H},\boldsymbol{\mathcal{N}}} + \mathbf{1}_N \otimes \left(\frac{1}{N}\sum_{i\in\mathcal{N}} \boldsymbol{H}_i(\boldsymbol{\theta}_i)\right)\right]_m} - \mathbf{1}_N \otimes\ \boldsymbol{\theta}^\star \\ &= -\boldsymbol{\theta}_{\mathcal{N}} + \widehat{\phi}_{\text{DN}}(\boldsymbol{\theta}_{\mathcal{N}}, \boldsymbol{\xi}_{\mathcal{N}}) \\ &\triangleq \widehat{\Phi}(\boldsymbol{\theta}_{\mathcal{N}}, \boldsymbol{\xi}_{\mathcal{N}}), \end{aligned}$$

where $\boldsymbol{\xi}_{\boldsymbol{g},\boldsymbol{\mathcal{N}}} = 1_N \otimes \boldsymbol{\xi}_g$, $\boldsymbol{\xi}_{\boldsymbol{H},\boldsymbol{\mathcal{N}}} = 1_N \otimes \boldsymbol{\xi}_H$, and the division is a Hadamard element-wise division.

We start by first assuming the existence of the equilibrium $\widehat{\boldsymbol{\theta}}(\boldsymbol{\xi})$ such that, for $\|\boldsymbol{\xi}\| \leq r$,

$$\widehat{\Phi}(\mathbf{1}_N \otimes \widehat{\boldsymbol{\theta}}(\boldsymbol{\xi}), \boldsymbol{\xi}_{\mathcal{N}}) = 0 \quad \text{and} \quad \widehat{\boldsymbol{\theta}}(\mathbf{0}) = \mathbf{0}, \tag{14}$$

and prove that it must satisfy (10). Since $\sum_{i\in\mathcal{N}} \boldsymbol{H}_i(\boldsymbol{\theta}_i) \geq mI$ by Assumption 2, then

$$\left[\boldsymbol{\xi}_{\boldsymbol{H},\boldsymbol{\mathcal{N}}} + \mathbf{1}_N \otimes \left(\frac{1}{N}\sum_{i\in\mathcal{N}} \boldsymbol{H}_i(\boldsymbol{\theta}_i)\right)\right]_m = \boldsymbol{\xi}_{\boldsymbol{H},\boldsymbol{\mathcal{N}}} + \mathbf{1}_N \otimes \left(\frac{1}{N}\sum_{i\in\mathcal{N}} \boldsymbol{H}_i(\boldsymbol{\theta}_i)\right) = \mathbf{1}_N \otimes \left(\frac{1}{N}\sum_{i\in\mathcal{N}} \boldsymbol{H}_i(\boldsymbol{\theta}_i) + \boldsymbol{\xi}_{\boldsymbol{H}}\right),$$

which implies that

$$\widehat{\Phi}(\mathbf{1}_N \otimes \widehat{\boldsymbol{\theta}}(\boldsymbol{\xi}), \boldsymbol{\xi}_{\mathcal{N}}) = -1_N \otimes \left(\widehat{\boldsymbol{\theta}}(\boldsymbol{\xi}) + \boldsymbol{\theta}^\star - \left(\boldsymbol{\xi}_H + \frac{1}{N}\sum_{i\in\mathcal{N}} \boldsymbol{H}_i(\widehat{\boldsymbol{\theta}}(\boldsymbol{\xi}))\right)^{-1}\left(\boldsymbol{\xi}_g + \frac{1}{N}\sum_{i\in\mathcal{N}} \boldsymbol{g}_i(\widehat{\boldsymbol{\theta}}(\boldsymbol{\xi})) + \frac{1}{N}\sum_{i\in\mathcal{N}} \boldsymbol{H}_i(\widehat{\boldsymbol{\theta}}(\boldsymbol{\xi}))\boldsymbol{\theta}^\star\right)\right)$$

equals zeros if and only if

$$\widehat{\boldsymbol{\theta}}(\boldsymbol{\xi}) = -\boldsymbol{\theta}^\star + \left(\boldsymbol{\xi}_H + \frac{1}{N}\sum_{i\in\mathcal{N}} \boldsymbol{H}_i(\widehat{\boldsymbol{\theta}}(\boldsymbol{\xi}))\right)^{-1}\left(\boldsymbol{\xi}_g + \frac{1}{N}\sum_{i\in\mathcal{N}} \boldsymbol{g}_i(\widehat{\boldsymbol{\theta}}(\boldsymbol{\xi})) + \frac{1}{N}\sum_{i\in\mathcal{N}} \boldsymbol{H}_i(\widehat{\boldsymbol{\theta}}(\boldsymbol{\xi}))\boldsymbol{\theta}^\star\right).$$

The equilibrium value (10) can be retrieve immediately from the above equation
(since $-\boldsymbol{\theta}^\star = \left(\boldsymbol{\xi}_H + N^{-1}\sum_{i\in\mathcal{N}} \boldsymbol{H}_i(\widehat{\boldsymbol{\theta}}(\boldsymbol{\xi}))\right)^{-1}\left(-\boldsymbol{\xi}_H\boldsymbol{\theta}^\star - N^{-1}\sum_{i\in\mathcal{N}} \boldsymbol{H}_i(\widehat{\boldsymbol{\theta}}(\boldsymbol{\xi}))\boldsymbol{\theta}^\star\right)$).

We now prove (14) by exploiting the Implicit Function Theorem (Krantz and Parks, 2012). From the definition of $\widehat{\Phi}$ applied to $1_N \otimes \boldsymbol{\theta}(\boldsymbol{\xi})$ we obtain N equivalent equations of the form

$$\widehat{\boldsymbol{\theta}}(\boldsymbol{\xi}) + \boldsymbol{\theta}^\star = \left(\frac{1}{N}\sum_{i\in\mathcal{N}} \boldsymbol{H}_i(\widehat{\boldsymbol{\theta}}(\boldsymbol{\xi})) + \boldsymbol{\xi}_H\right)^{-1}\left(\frac{1}{N}\sum_{i\in\mathcal{N}} \boldsymbol{g}_i(\widehat{\boldsymbol{\theta}}(\boldsymbol{\xi})) + \boldsymbol{\xi}_g + \frac{1}{N}\sum_{i\in\mathcal{N}} \boldsymbol{H}_i(\widehat{\boldsymbol{\theta}}(\boldsymbol{\xi}))\boldsymbol{\theta}^\star\right).$$

For the continuity of Assumption2, and the fact that $N^{-1}\sum_{i\in\mathcal{N}} \boldsymbol{H}_i(\boldsymbol{\theta}^\star) \geq mI$ thanks to Assumption 3, there exists a sufficiently small $r > 0$ such that if $\|\boldsymbol{\xi}_H\| \leq \|\boldsymbol{\xi}\| \leq r$ then $N^{-1}\sum i \in \mathcal{N}\boldsymbol{H}_i(\boldsymbol{\theta}^\star) + \boldsymbol{\xi}_H$ is still invertible, leading to

$$\frac{1}{N}\sum_{i\in\mathcal{N}} \boldsymbol{g}_i(\widehat{\boldsymbol{\theta}}(\boldsymbol{\xi})) + \boldsymbol{\xi}_g + \frac{1}{N}\sum_{i\in\mathcal{N}} \boldsymbol{H}_i(\widehat{\boldsymbol{\theta}}(\boldsymbol{\xi}))\boldsymbol{\theta}^\star = \frac{1}{N}\sum_{i\in\mathcal{N}} \boldsymbol{H}_i(\widehat{\boldsymbol{\theta}}(\boldsymbol{\xi}))(\widehat{\boldsymbol{\theta}}(\boldsymbol{\xi}) + \boldsymbol{\theta}^\star) + \boldsymbol{\xi}_g(\widehat{\boldsymbol{\theta}}(\boldsymbol{\xi}) + \boldsymbol{\theta}^\star).$$

By recalling the definition of $\boldsymbol{g}_i$, and in particular the fact that $N^{-1}\sum_{i\in\mathcal{N}} \boldsymbol{g}_i(\widehat{\boldsymbol{\theta}}(\boldsymbol{\xi})) = N^{-1}\sum_{i\in\mathcal{N}} \boldsymbol{H}_i(\widehat{\boldsymbol{\theta}}(\boldsymbol{\xi}))\widehat{\boldsymbol{\theta}}(\boldsymbol{\xi}) - \boldsymbol{\nabla} J(\widehat{\boldsymbol{\theta}}(\boldsymbol{\xi}))$, it follows that $\widehat{\boldsymbol{\theta}}(\boldsymbol{\xi})$ must satisfy the following condition:

$$\boldsymbol{\nabla} J(\widehat{\boldsymbol{\theta}}(\boldsymbol{\xi})) - \boldsymbol{\xi}_g + \boldsymbol{\xi}_H(\widehat{\boldsymbol{\theta}}(\boldsymbol{\xi}) + \boldsymbol{\theta}^\star) = 0.$$

Given Assumption 2, the left-hand side of the previous equation is a continuously differentiable function, for which we can write

$$\frac{\partial\Big(\boldsymbol{\nabla} J(\widehat{\boldsymbol{\theta}}(\boldsymbol{\xi})) - \boldsymbol{\xi}_g + \boldsymbol{\xi}_H(\widehat{\boldsymbol{\theta}}(\boldsymbol{\xi}) + \boldsymbol{\theta}^\star)\Big)}{\partial\widehat{\boldsymbol{\theta}}(\boldsymbol{\xi})} = \boldsymbol{\nabla}^2 J(\widehat{\boldsymbol{\theta}}(\boldsymbol{\xi})) + \boldsymbol{\xi}_H.$$

Again, if r is sufficiently small and thanks to Assumption 2, the differentiation is an invertible matrix and, by the Implicit Function Theorem, $\widehat{\boldsymbol{\theta}}(\boldsymbol{\xi})$ exists, is unique and continuously differentiable.

For what concerns the translated perturbed multi-agent Newton's dynamics $\widehat{\phi}'_{\text{DN}}(\boldsymbol{\theta}_\mathcal{N}, \boldsymbol{\xi}) \triangleq \widehat{\phi}_{\text{DN}}(\boldsymbol{\theta}_\mathcal{N} + \mathbf{1}_N \otimes \widehat{\boldsymbol{\theta}}(\boldsymbol{\xi}), \boldsymbol{\xi}_\mathcal{N})$, it is immediate to prove that the origin is an equilibrium point, *i.e.*, $\widehat{\phi}'_{\text{DN}}(\mathbf{0}, \boldsymbol{\xi}) = 0, \forall \|\boldsymbol{\xi}\| \leq r$. To establish the stability of this translated version $\widehat{\phi}'_{\text{DN}}$ of the perturbed multi-agent Newton's dynamics, we need to study the dynamics

$$\dot{\boldsymbol{\theta}}_\mathcal{N} = -\boldsymbol{\theta}_\mathcal{N} + \widehat{\phi}'_{\text{DN}}(\boldsymbol{\theta}_\mathcal{N}, \boldsymbol{\xi}) \triangleq \widehat{\Phi}'(\boldsymbol{\theta}_\mathcal{N}, \boldsymbol{\xi}). \tag{15}$$

Similarly to what was done for Theorem 4, we employ the same Lyapunov function (13) and prove, under the made assumptions, that exist positive scalars r, c_1, c_2 such that, $\forall \boldsymbol{\theta}_\mathcal{N} \in \mathbb{R}^{dN}$ and $\boldsymbol{\xi}$ satisfying $\|\boldsymbol{\xi}\| \leq r$,

$$\begin{cases} \frac{\partial V}{\partial \boldsymbol{\theta}_\mathcal{N}}\widehat{\Phi}'(\boldsymbol{\theta}_\mathcal{N}, \boldsymbol{\xi}) \leq -c_1\|\boldsymbol{\theta}_N\|^2 \\ \left\|\widehat{\Phi}'(\boldsymbol{\theta}_\mathcal{N}, \boldsymbol{\xi})\right\| \leq c_2\|\boldsymbol{\theta}_\mathcal{N}\|. \end{cases}$$

For the first condition, we have that, $\forall \boldsymbol{\theta} \in \mathbb{R}^{dN}$,

$$\begin{aligned} \frac{\partial V}{\partial \boldsymbol{\theta}_\mathcal{N}}\widehat{\Phi}'_{\text{DN}}(\boldsymbol{\theta}_\mathcal{N}, \boldsymbol{\xi}) &= \frac{\partial V}{\partial \boldsymbol{\theta}_\mathcal{N}}\widehat{\Phi}'_{\text{DN}}(\boldsymbol{\theta}_\mathcal{N}, \mathbf{0}) + \frac{\partial V}{\partial \boldsymbol{\theta}_\mathcal{N}}\Big(\widehat{\Phi}'_{\text{DN}}(\boldsymbol{\theta}_\mathcal{N}, \boldsymbol{\xi}) - \widehat{\Phi}'_{\text{DN}}(\boldsymbol{\theta}_\mathcal{N}, \mathbf{0})\Big) \\ &\leq \frac{\partial V}{\partial \boldsymbol{\theta}_\mathcal{N}}\Phi(\boldsymbol{\theta}_\mathcal{N}) + \left\|\frac{\partial V}{\partial \boldsymbol{\theta}_\mathcal{N}}\right\|\left\|\widehat{\Phi}'_{\text{DN}}(\boldsymbol{\theta}_\mathcal{N}, \boldsymbol{\xi}) - \widehat{\Phi}'_{\text{DN}}(\boldsymbol{\theta}_\mathcal{N}, \mathbf{0})\right\| \\ &\leq -b_3\|\boldsymbol{\theta}_\mathcal{N}\|^2 + b_2\|\boldsymbol{\theta}_\mathcal{N}\|k\|\boldsymbol{\xi}\|\|\boldsymbol{\theta}_\mathcal{N}\| \\ &\leq -(b_3 - b_2kr)\|\boldsymbol{\theta}_\mathcal{N}\|^2 \\ &\leq -c_1\|\boldsymbol{\theta}_\mathcal{N}\|^2. \end{aligned}$$

The inequality is meaningful for $r < b_3/(b_2 k)$.

The second condition follows by considering that, $\forall \boldsymbol{\theta}_{\mathcal{N}} \in \mathbb{R}^{dN}$,

$$\begin{aligned}\left\|\widehat{\Phi}'(\boldsymbol{\theta}_{\mathcal{N}}, \boldsymbol{\xi})\right\| &\leq \left\|\widehat{\Phi}'(\boldsymbol{\theta}_{\mathcal{N}}, \mathbf{0})\right\| + \left\|\widehat{\Phi}'(\boldsymbol{\theta}_{\mathcal{N}}, \boldsymbol{\xi}) - \widehat{\Phi}'(\boldsymbol{\theta}_{\mathcal{N}}, \mathbf{0})\right\| \\ &\leq (b_4 + kr)\|\boldsymbol{\theta}_{\mathcal{N}}\| \\ &\leq c_2\|\boldsymbol{\theta}_{\mathcal{N}}\|\end{aligned}$$

Following the same reasoning in Theorem 4, we can claim that (15) and its discrete-time counterpart are globally exponentially stable.