Dynamic Games and Applications (2023) 13:25–55
https://doi.org/10.1007/s13235-022-00449-9

Multi-Agent Natural Actor-Critic Reinforcement Learning Algorithms

Prashant Trivedi[1] · **Nandyala Hemachandra**[1]

Accepted: 11 April 2022 / Published online: 16 June 2022

Abstract
Multi-agent actor-critic algorithms are an important part of the Reinforcement Learning (RL) paradigm. We propose three fully decentralized multi-agent natural actor-critic (MAN) algorithms in this work. The objective is to collectively find a joint policy that maximizes the average long-term return of these agents. In the absence of a central controller and to preserve privacy, agents communicate some information to their neighbors via a time-varying communication network. We prove convergence of all the three MAN algorithms to a globally asymptotically stable set of the ODE corresponding to actor update; these use linear function approximations. We show that the Kullback–Leibler divergence between policies of successive iterates is proportional to the objective function's gradient. We observe that the minimum singular value of the Fisher information matrix is well within the reciprocal of the policy parameter dimension. Using this, we theoretically show that the optimal value of the deterministic variant of the MAN algorithm at each iterate dominates that of the standard gradient-based multi-agent actor-critic (MAAC) algorithm. To our knowledge, it is the first such result in multi-agent reinforcement learning (MARL). To illustrate the usefulness of our proposed algorithms, we implement them on a bi-lane traffic network to reduce the average network congestion. We observe an almost 25% reduction in the average congestion in 2 MAN algorithms; the average congestion in another MAN algorithm is on par with the MAAC algorithm. We also consider a generic 15 agent MARL; the performance of the MAN algorithms is again as good as the MAAC algorithm.

Keywords Natural Gradients · Actor-Critic Methods · Networked Agents · Traffic Network Control · Stochastic Approximations · Function Approximations · Fisher Information Matrix · Non-Convex Optimization · Quasi second-order methods · Local optima value comparison · Algorithms for better local minima

This article is part of the topical collection "Multi-agent Dynamic Decision Making and Learning" edited by Konstantin Avrachenkov, Vivek S. Borkar and U. Jayakrishnan Nair.

✉ Prashant Trivedi
trivedi.prashant15@iitb.ac.in

Nandyala Hemachandra
nh@iitb.ac.in

[1] Industrial Engineering and Operations Research IIT Bombay, Mumbai, India

1 Introduction

Reinforcement learning (RL) has been explored in recent years and is of great interest to researchers because of its broad applicability in many real-life scenarios. In RL, agents interact with the environment and take decisions sequentially. It is applied successfully to various problems, including elevator scheduling, robot control, etc. There are many instances where RL agents surpass human performance, such as openAI beating the world champion DOTA player, DeepMind beating the world champion of Alpha Star.[1]

The sequential decision-making problems are generally modeled via Markov decision process (MDP). It requires the knowledge of system transitions and rewards. In contrast, RL is a data-driven MDP framework for sequential decision-making tasks; the transition probability matrices and the reward functions are not assumed, but their realizations are available as observed data.

In RL, the purpose of an agent is to learn an optimal or nearly-optimal policy that maximizes the "reward function" or functions of other user-provided "reinforcement signals" from the observed data. However, in many realistic scenarios, there is more than one agent. To this end, researchers explore the multi-agent reinforcement learning (MARL) methods, but most are centralized and relatively slow. Furthermore, these MARL algorithms use the standard/vanilla gradient, which has limitations. For example, the standard gradients cannot capture the angles in the state space and may not be effective in many scenarios. The natural gradients are more suitable choices because they capture the intrinsic curvature in the state space. In this work, we are incorporating natural gradients in the MARL framework.

In the multi-agent setup that we consider, the agents have some private information and a common goal. This goal could be achieved by deploying a central controller and converting the MARL problem into a single-agent RL problem. However, deploying a central controller often leads to scalability issues. On the other hand, if there is no central controller and the agents do not share any information, then there is almost no hope of achieving the common goal. An intermediate model is to share some parameters via (possibly) a time-varying, and sparse communication matrix [53]. The algorithms based on such intermediate methods are often attributed as consensus-based algorithms.

The consensus-based algorithm models can also be considered as intermediate between dynamic non-cooperative and cooperative game models. Non-cooperative games, as multi-agent systems, model situations where the agents do not have a common goal and do not communicate. On the contrary, cooperative games model situations where a central controller achieves a common goal using complete information.

Algorithm 2 of [53] is a consensus-based actor-critic algorithm. We call it MAAC (multi-agent actor-critic) algorithm. The MAAC algorithm uses the standard gradient and hence lacks in capturing the intrinsic curvature present in the state space. We propose three multi-agent natural actor-critic (MAN) algorithms and incorporate the curvatures via natural gradients. These algorithms use the linear function approximations for the state value and reward functions. We prove the convergence of all the three MAN algorithms to a globally asymptotically stable equilibrium set of ordinary differential equations (ODEs) obtained from the actor updates.

Here is a brief overview of our two time-scale approach. Let $J(\theta)$ be the global MARL objective function of n agents, where $\theta = (\theta^1, \ldots, \theta^n)$ is the actor (or policy) parameter. For a given policy parameter θ of each MAN algorithm, we first show in Theorem 4 the convergence of critic parameters (to be defined later) on a faster time scale. Note that these

[1] A detailed version of this paper is available at: arXiv:2109.01654.

critic parameters are updated via the communication matrix. We then show the convergence of each agent's actor parameters to an asymptotically stable attractor set of its ODE. These actor updates use the natural gradients in the form of Fisher information matrix and advantage parameters (Theorem 6, 8 and 10). The actor parameter θ is shown to converge on the slower time scale.

Our MAN algorithms use a log-likelihood function via the Fisher information matrix and incorporate the curvatures. We show that this log-likelihood function is indeed the KL divergence between the consecutive policies, and it is the gradient of the objective function up to scaling (Lemma 1). Unlike standard gradient methods, where the updates are restricted to the parameter space only, the natural gradient-based methods allow the updates to factor in the curvature of the policy distribution prediction space via the KL divergence between them. Thus, two of our MAN algorithms, FI-MAN and FIAP-MAN, use a certain *representation* of the objective function gradient in terms of the gradient of this KL divergence (Lemma 1). It turns out these two algorithms have much better empirical performance (Sect. 5.1).

We now point out a couple of important consequences of the representation learning aspect of our MAN algorithms for reinforcement learning. First, we show that under some conditions, our deterministic version of the FI-MAN algorithm converges to local minima with a better objective function value than the deterministic counterpart of the MAAC algorithm, Theorem 3. To the best of our knowledge, this is a new result in non-convex optimization; we are not aware of any algorithm that is *proven* to converge to a better local maxima [13, 37]. This relies on the important observation, which can be of independent interest, that $1/m$ is *uniform* upper bound on the smallest singular value of Fisher information matrix $G(\theta)$, Lemma 2; here, m is the common dimension of the compatible policy parameter θ and the Fisher information matrix $G(\theta)$.

The natural gradient-based methods can be viewed as quasi-second order methods, as the Fisher information matrix $G(\cdot)$ is an invertible linear transformation of basis that is used in first-order optimization methods [1]. However, they are not regarded as second-order methods because the Fisher information matrix is not the Hessian of the *objective function*.

To validate the usefulness of our proposed algorithms, we perform a comprehensive set of computational experiments in two settings: a bi-lane traffic network and an abstract MARL model. On a bi-lane traffic network model, the objective is to find the traffic signaling plan that reduces the overall network congestion. We consider two different arrival patterns between various origin-destination (OD) pairs. With the suitable linear function approximations to incorporate the humongous state space (50^{16}) and action space (3^4), we observe a significant reduction ($\approx 25\%$) in the average network congestion in two of our MAN algorithms. One of our MAN algorithms that are only based on the advantage parameters and never estimate the Fisher information matrix inverse is on-par with the MAAC algorithm. In the abstract MARL model, we consider 15 agents with 15 states and two actions in each state and generic reward functions [18, 53]. Each agent's reward is private information and hence not known to other agents. Our MAN algorithms either outperform or are on-par with the MAAC algorithm with high confidence.

2 MARL Framework and Natural Gradients

Let $N = \{1, 2, \ldots, n\}$ denote the set of agents. Each agent independently interacts with a stochastic environment and takes a local action. We consider a fully decentralized setup in which a communication network connect the agents.

This network is used to exchange information among agents in the absence of a central controller so that agents' privacy remains intact. The communication network is possibly time-varying and sparse. We assume the communication among agents is synchronized, and hence there are no information delays. Moreover, only some parameters (that we define later) are shared among the neighbors of each agent. It also addresses an important aspect of the privacy protection of such agents. Formally, the communication network is characterized by an undirected graph $\mathcal{G}_t = (N, \mathcal{E}_t)$, where N is the set of all nodes (or agents) and $\mathcal{E}_t$ is the set of communication links available at time $t \in \mathbb{N}$. We say, agents $i, j \in N$ communicate at time t if $(i, j) \in \mathcal{E}_t$.

Let $\mathcal{S}$ denote the common state space available to all the agents. At any time t, each agent observes a common state $s_t \in \mathcal{S}$, and takes a local action a_t^i from the set of available actions $\mathcal{A}^i$. We assume that for any agent $i \in N$, the entire action set $\mathcal{A}^i$ is feasible in every state $s \in \mathcal{S}$. The action a_t^i is taken as per a local policy $\pi^i : \mathcal{S} \times \mathcal{A}^i \to [0, 1]$, where $\pi^i(s_t, a_t^i)$ is the probability of taking action a_t^i in state s_t by agent $i \in N$. Let $\mathcal{A} := \prod_{i=1}^n \mathcal{A}^i$ be the joint action space of all the agents. To each state and action pair, every agent receives a finite reward from the local reward function $R^i : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$. Note that the reward is private information of the agent, and it is not known to other agents. The state transition probability of MDP is given by $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$. Using only local rewards and actions, it is hard for any classical reinforcement learning algorithm to maximize the averaged reward determined by the joint action of all the agents. To this end, we consider the multi-agent networked MDP given in [53]. The multi-agent networked MDP is defined as $(\mathcal{S}, \{\mathcal{A}^i\}_{i\in N}, P, \{R^i\}_{i\in N}, \{\mathcal{G}_t\}_{t\geq 0})$, with each component described as above. Let joint policy of all agents be denoted by $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$ satisfying $\pi(s, a) = \prod_{i\in N} \pi^i(s, a^i)$. Let $a_t = (a_t^1, \ldots, a_t^n)$ be the action taken by all the agents at time t. Depending on the action a_t^i the agent i receives a random reward r_{t+1}^i with the expected value $R^i(s_t, a_t)$. Moreover, with probability $P(s_{t+1}|s_t, a_t)$ the multi-agent MDP shifts to next state $s_{t+1} \in \mathcal{S}$.

Due to the large state and action space, it is often helpful to consider the parameterized policies [23, 45]. We parameterize the local policy, $\pi^i(\cdot, \cdot)$ by $\theta^i \in \Theta^i \subseteq \mathbb{R}^{m_i}$, where Θ^i is the compact set. To find the global policy parameters, we can pack all the local policy parameters as $\theta = [(\theta^1)^\top, \ldots, (\theta^n)^\top]^\top \in \Theta \subseteq \mathbb{R}^m$, where $\Theta := \prod_{i\in N} \Theta^i$, and $m = \sum_{i=1}^n m_i$. The parameterized joint policy is then given by $\pi_\theta(s, a) = \prod_{i\in N} \pi_{\theta^i}^i(s, a^i)$. The objective of the agents is to collectively find a joint policy π_θ that maximizes the averaged long-term return $J(\theta)$, provided each agent has local information only, i.e.,

$$max_\theta\ J(\theta) = \lim_{T\to\infty} \frac{1}{T} \mathbb{E}\left(\sum_{t=0}^{T-1} \frac{1}{n} \sum_{i\in N} r_{t+1}^i \right) = \sum_{s\in\mathcal{S}} d_\theta(s) \sum_{a\in\mathcal{A}} \pi_\theta(s, a) \bar{R}(s, a), \tag{1}$$

where $\bar{R}(s, a) = \frac{1}{n} \sum_{i\in N} R^i(s, a)$ is the globally averaged reward function. Let $\bar{r}_t = \frac{1}{n} \sum_{i\in N} r_t^i$. Thus, $\bar{R}(s, a) = \mathbb{E}[\bar{r}_{t+1}|s_t = s, a_t = a]$.

Like single-agent RL [7], we also require following regularity assumption on networked multi-agent MDP and parameterized policies.

X 1 *For each agent $i \in N$, the local policy function $\pi_{\theta^i}^i(s, a^i) > 0$ for any $s \in \mathcal{S}, a^i \in \mathcal{A}^i$ and $\theta^i \in \Theta^i$. Also $\pi_{\theta^i}^i(s, a^i)$ is continuously differentiable with respect to parameters θ^i over Θ^i. Moreover, for any $\theta \in \Theta$, let P^θ be the transition matrix for the Markov chain $\{s_t\}_{t\geq 0}$ induced by policy π_θ, that is, $P^\theta(s'|s) = \sum_{a\in\mathcal{A}} \pi_\theta(s, a) P(s'|s, a)$ for any $s, s' \in \mathcal{S}$. Further, the Markov chain $\{s_t\}_{t\geq 0}$ is ergodic under π_θ with stationary distribution $d_\theta(s)$ over $\mathcal{S}$.*

The regularity assumption X. 1 on a multi-agent networked MDP is standard in the work of single agent actor-critic algorithms with function approximations [7, 27]. The continuous differentiability of policy $\pi_\theta(\cdot,\cdot)$ with respect to θ is required in policy gradient theorem [45], and it is commonly satisfied by well-known class of functions such as neural networks or deep neural networks. Moreover, assumption X. 1 also implies that the Markov chain $\{(s_t, a_t)\}_{t\geq 0}$ has stationary distribution $\tilde{d}_\theta(s,a) = d_\theta(s)\cdot\pi_\theta(s,a)$ for any $s\in\mathcal{S}, a\in\mathcal{A}$.

Based on the objective function given in Eq. (1), the global state-action value function associated with state-action pair (s,a) for a given policy π_θ is defined as $Q_\theta(s,a) = \sum_{t\geq 0}\mathbb{E}[\bar{r}_{t+1} - J(\theta)|s_0 = s, a_0 = a, \pi_\theta]$. Note that $Q_\theta(s,a)$ is motivated from the gain and bias relation for average reward criteria of the single-agent MDP as given in say, Sect. 8.2.1 in [41]. It captures the expected sum of fluctuations of the global rewards about the globally averaged objective function ("average adjusted sum of rewards" [30]) when action a is taken in state $s\in\mathcal{S}$ at time $t=0$, and thereafter the policy π_θ is followed. The global state value function is defined as $V_\theta(s) = \sum_{a\in\mathcal{A}}\pi_\theta(s,a)\cdot Q_\theta(s,a)$.

Let $A_\theta(s,a) := Q_\theta(s,a) - V_\theta(s)$ be the global advantage function. Note that the advantage function captures the benefit of taking action a in state s and thereafter following the policy π_θ over the case when policy π_θ is followed from state s itself. For the multi-agent setup, we define the local advantage function for each agent $i\in N$ as $A^i_\theta(s,a) := Q_\theta(s,a) - \widetilde{V}^i_\theta(s,a^{-i})$, where $\widetilde{V}^i_\theta(s,a^{-i}) := \sum_{a^i\in A^i}\pi^i_{\theta^i}(s,a^i)Q_\theta(s,a^i,a^{-i})$. Note that $\widetilde{V}^i_\theta(s,a^{-i})$ represents the value of state s to an agent $i\in N$ when policy $\pi(\cdot,\cdot)$ is parameterized by θ, and all other agents are taking action $a^{-i} = (a^1,\ldots,a^{i-1},a^{i+1},\ldots,a^n)$.

Theorem 1 (Policy gradient theorem for MARL [53]) *Under assumption X. 1, for any $\theta\in\Theta$, and each agent $i\in N$, the gradient of $J(\theta)$ with respect to θ^i is given by*

$$\nabla_{\theta^i} J(\theta) = \mathbb{E}[\nabla_{\theta^i}\log\pi^i_{\theta^i}(s,a^i)\cdot A_\theta(s,a)] = \mathbb{E}[\nabla_{\theta^i}\log\pi^i_{\theta^i}(s,a^i)\cdot A^i_\theta(s,a)].$$

We refer to Appendix A. 1 of [49] for complete proof. The idea of the proof is as follows: we first recall the policy gradient theorem for single agent. Now using the fact that for multi-agent case, the global policy is product of local policies, i.e., $\pi_\theta(s,a) = \prod_{i=1}^n \pi^i_{\theta^i}(s,a^i)$, and $\sum_{a^i\in\mathcal{A}^i}\pi^i_{\theta^i}(s,a^i) = 1$, hence $\nabla_{\theta^i}\left[\sum_{a^i\in\mathcal{A}^i}\pi^i_{\theta^i}(s,a^i)\right] = 0$, we show $\nabla_{\theta^i}J(\theta) = \mathbb{E}_{s\sim d_\theta, a\sim\pi_\theta}[\nabla_{\theta^i}\log\pi^i_{\theta^i}(s,a^i)\cdot Q_\theta(s,a)]$. Now, observe that adding/subtracting any function Λ that is independent of the action a^i taken by agent $i\in N$ to $Q_\theta(s,a)$ does not make any difference in the above expected value. In particular, considering two such Λ functions $V_\theta(s)$ and $\widetilde{V}^i_\theta(s,a^{-i})$, we have desired results.

We call $\psi^i(s,a^i) := \nabla_{\theta^i}\log\pi^i_{\theta^i}(s,a^i)$, the score function. We will see in Sect. 3.2 that the same score function is called the compatible features. This is because the above policy gradient theorem with linear function approximations require the compatibility condition (Theorem 2 [46]). The policy gradient theorem for MARL relates the gradient of the global objective function w.r.t. θ^i and the local advantage function $A^i_\theta(\cdot,\cdot)$. It also suggests that the global objective function's gradients can be obtained solely using the local score function $\psi^i(s,a^i)$, if agent $i\in N$ has an unbiased estimate of the advantage functions A^i_θ or A_θ. However, estimating the advantage function requires the rewards r^i_t of *all* the agents $i\in N$; therefore, these functions cannot be well-estimated by any agent $i\in N$ alone. To this end, [53] have proposed two fully decentralized actor-critic algorithms based on the consensus network. These algorithms work in a fully decentralized fashion and empirically achieve the same performance as a centralized algorithm in the long run. We use algorithm 2 of [53] which we are calling as multi-agent actor-critic (MAAC) algorithm.

In the fully decentralized setup, we consider the weight matrix $C_t = [c_t(i, j)]$, depending on the network topology of communication network $\mathcal{G}_t$. Here, $c_t(i, j)$ represents the weight of the message transmitted from agent i to agent j at time t. For generality, we take the weight matrix C_t to be random. This is either because $\mathcal{G}_t$ is a time-varying graph or the randomness in the consensus algorithm [14]. Following are the assumptions on the matrix C_t.

X 2 *The sequence of nonnegative random matrices* $\{C_t\}_{t\geq 0} \subseteq \mathbb{R}^{n\times n}$ *satisfy*

1. C_t *is row stochastic, i.e.,* $C_t\mathbb{1} = \mathbb{1}$. *Moreover,* $\mathbb{E}(C_t)$ *is column stochastic, i.e.,* $\mathbb{1}^\top\mathbb{E}(C_t) = \mathbb{1}^\top$. *Furthermore, there exists a constant* $\gamma \in (0, 1)$ *such that for any* $c_t(i, j) > 0$, *we have* $c_t(i, j) \geq \gamma$.
2. *Weight matrix* C_t *respects* $\mathcal{G}_t$, *i.e.,* $c_t(i, j) = 0$, *if* $(i, j) \notin \mathcal{E}_t$.
3. *The spectral norm of* $\mathbb{E}[C_t^\top(I - \mathbb{1}\mathbb{1}^\top/n)C_t]$ *is smaller than one.*
4. *Given the* σ*-algebra generated by the random variables before time* t, C_t *is conditionally independent of* r_{t+1}^i *for each agent* $i \in N$.

Assumption X. 2(1) of considering a doubly stochastic matrix is standard in the work of consensus-based algorithms [10, 32]. To prove the stability of the consensus update (see Appendix A of [53] for detailed proof), we require the lower bound on the weights of the matrix [35]. Assumption X. 2(2) is required for the connectivity of $\mathcal{G}_t$. For geometric convergence in distributed optimization, authors in [34] provide the connection between the time-varying network and the spectral norm property. The same is required for convergence in our work also (assumption X. 2(3)). Assumption X. 2(4) on the conditional independence of C_t and r_{t+1} is common in many practical multi-agent systems.

Next, we outline the actor-critic algorithm using linear function approximations in a fully decentralized setting. The actor-critic algorithm consists of two steps—critic step and actor step. At each time t, the actor suggests a policy parameter θ_t. The critic evaluates its value using the policy parameters and criticizes or gives the feedback to the actor. Using this feedback, actor then update the policy parameters, and this continues until convergence. Let the global state value temporal difference (TD) error be defined as $\bar{\delta}_t = \bar{r}_{t+1} - J(\theta) + V_\theta(s_{t+1}) - V_\theta(s_t)$. It is known that the state value temporal difference error is an unbiased estimate of the advantage function A_θ [45], i.e., $\mathbb{E}[\bar{\delta}_t \mid s_t = s, a_t = a, \pi_\theta] = A_\theta(s, a),\ \forall\, s \in \mathcal{S},\ a \in \mathcal{A}$. The TD error specifies how different the new value is from the old prediction. In many applications [16, 17], the state space is either large or infinite. To this end, we use the linear function approximations for state value function. Later in Sect. 3.2, we use linear function approximation for advantage function also.

Let the state value function $V_\theta(s)$ be approximated using the linear function as $V_\theta(s; v) := v^\top\varphi(s)$, where $\varphi(s) = [\varphi_1(s), \ldots, \varphi_L(s)]^\top \in \mathbb{R}^L$ is the feature associated with state s, and $v \in \mathbb{R}^L$. Note that $L << |\mathcal{S}|$; hence, the value function is approximated using very small number of features. Moreover, let μ_t^i be the estimate of the global objective function $J(\theta)$ by agent $i \in N$ at time t. Note that μ_t^i tracks the long-term return to each agent $i \in N$. The MAAC algorithm (Appendix C. 4 of [49]) uses consensus network and consists of the following updates for objective function estimate and the critic parameters

$$\tilde{\mu}_t^i = (1 - \beta_{v,t}) \cdot \mu_t^i + \beta_{v,t} \cdot r_{t+1}^i; \quad \mu_{t+1}^i = \sum_{j\in N} c_t(i, j)\tilde{\mu}_t^j \tag{2}$$

$$\tilde{v}_t^i = v_t^i + \beta_{v,t} \cdot \delta_t^i \cdot \nabla_v V_t(v_t^i); \quad v_{t+1}^i = \sum_{j\in N} c_t(i, j)\tilde{v}_t^j, \tag{3}$$

where $\beta_{v,t} > 0$ is the critic step-size and $\delta_t^i = r_{t+1}^i - \mu_t^i + V_{t+1}(v_t^i) - V_t(v_t^i)$ is the local TD error. Here, $V_{t+1}(v_t^i) := v_t^{i\top}\varphi(s_{t+1})$, and hence, $V_{t+1}(v_t^i) = V_\theta(s_{t+1}; v_t^i)$. It is a linear

function approximation of the state value function, $V_\theta(s_{t+1})$ by agent $i \in N$. Note that the estimate of the advantage function $A_\theta(s,a)$ require $\bar{r}_{t+1}$ which is not available to each agent $i \in N$. Therefore, we parameterize the reward function $\bar{R}(s,a)$ used in the critic update as well.

Let $\bar{R}(s,a)$ be approximated using a linear function as $\bar{R}(s,a;\lambda) = \lambda^\top f(s,a)$, where $f(s,a) = [f_1(s,a),\ldots,f_M(s,a)]^\top \in \mathbb{R}^M$, $M << |\mathcal{S}||\mathcal{A}|$ are the features associated with state action pair (s,a). To obtain the estimate of $\bar{R}(s,a)$, we use the following least square minimization:

$$min_\lambda \sum_{s\in\mathcal{S},a\in\mathcal{A}} \tilde{d}_\theta(s,a)[\bar{R}(s,a) - \bar{R}(s,a;\lambda)]^2,$$

where $\bar{R}(s,a) := \frac{1}{n}\sum_{i\in N} R^i(s,a)$, and $\tilde{d}_\theta(s,a) = d_\theta(s)\cdot\pi_\theta(s,a)$. This optimization problem can be *equivalently characterized* as follows:

$$min_\lambda \sum_{i\in N}\sum_{s\in\mathcal{S},a\in\mathcal{A}} \tilde{d}_\theta(s,a)[R^i(s,a) - \bar{R}(s,a;\lambda)]^2,$$

i.e., both the optimization problems have the same stationary points.

For more details on stationary points, see Appendix A.2 of [49]. Taking first-order derivative with respect to λ implies that we should also do the following in critic update:

$$\tilde{\lambda}^i_t = \lambda^i_t + \beta_{v,t}\cdot[r^i_{t+1} - \bar{R}_t(\lambda^i_t)]\cdot\nabla_\lambda\bar{R}_t(\lambda^i_t);\;\; \lambda^i_{t+1} = \sum_{j\in N} c_t(i,j)\tilde{\lambda}^j_t, \tag{4}$$

where $\bar{R}_t(\lambda^i_t)$ is the linear function approximation of the global reward $\bar{R}_t(s_t,a_t)$ by agent $i \in N$ at time t, i.e., $\bar{R}_t(\lambda^i_t) = \lambda^{i^\top}_t f(s_t,a_t)$. The TD error with parameterized reward $\bar{R}_t(\cdot)$ is given by $\tilde{\delta}^i_t := \bar{R}_t(\lambda^i_t) - \mu^i_t + V_{t+1}(v^i_t) - V_t(v^i_t)$.

Note that each agent $i \in N$ know its local reward function $r^i_{t+1}(s_t,a_t)$, but at the same time also seeks to get some information about the global reward, $\bar{r}_{t+1}(s_t,a_t)$ because the objective is to maximize the globally averaged reward function. Therefore, in Eq. (4), each agent $i \in N$ uses $\bar{R}_t(\lambda^i_t)$ as an estimate of the global reward function. Let $\beta_{\theta,t} > 0$ be the actor step-size, then each agent $i \in N$ updates the policy/actor parameters as

$$\theta^i_{t+1} = \theta^i_t + \beta_{\theta,t}\cdot\tilde{\delta}^i_t\cdot\psi^i_t.$$

Note that we have used $\tilde{\delta}^i_t\cdot\psi^i_t$ instead of gradient of the objective function $\nabla_{\theta^i}J(\theta)$ in the actor update. However, $\tilde{\delta}^i_t\cdot\psi^i_t$ may not be an unbiased estimate of $\nabla_{\theta^i}J(\theta)$. That is,

$\mathbb{E}_{s_t\sim d_\theta,a_t\sim\pi_\theta}[\tilde{\delta}^i_t\cdot\psi^i_t] = \nabla_{\theta^i}J(\theta)+b$, where $b = \mathbb{E}_{s_t\sim d_\theta,a_t\sim\pi_\theta}[(\bar{R}_t(\lambda^i_t)-\bar{R}(s_t,a_t))\cdot\psi^i_t] + \mathbb{E}_{s_t\sim d_\theta}[(V_\theta(s_t) - V_t(v^i_t))\cdot\psi^i_t]$ is the bias term. (For more details, please refer to page 6 of [49].)

The bias term captures the sum of the expected linear approximation errors in the reward and value functions. If these approximation errors are small, the convergence point of the ODE corresponding to the actor update (as given in Sect. 4) is close to the local optima of $J(\theta)$. In fact, in Sect. 4, we show that the actor parameters converge to asymptotically stable equilibrium set of the ODEs corresponding to actor updates, hence possibly nullifying the bias.

To prove the convergence of actor-critic algorithm, we require $\beta_{\theta,t} = o(\beta_{v,t})$, and $lim_t\frac{\beta_{v,t+1}}{\beta_{v,t}} = 1$. Moreover, we also require (a) $\sum_t\beta_{v,t} = \sum_t\beta_{\theta,t} = \infty$; (b) $\sum_t\left(\beta^2_{v,t}+\beta^2_{\theta,t}\right) < \infty$, i.e., critic update is made at the faster time scale than the actor update. Condition in (a) ensures that the discrete time steps $\beta_{v,t}$, $\beta_{\theta,t}$ used in the critic and

actor steps do cover the entire time axis while retaining $\beta_{v,t}, \beta_{\theta,t} \to 0$. We also require the error due to the estimates used in the critic and actor updates are asymptotically negligible almost surely. So, condition in (b) asymptotically suppresses the variance in the estimates [11]; see [47] for some recent developments that do away with this requirement.

The MAAC algorithm uses standard gradients. However, they are most useful for the reward functions that have single optima and whose gradients are isotropic in magnitude for any direction away from its optimum [3]. None of these properties are valid in typical reinforcement learning environments. Apart from this, the performance of standard gradient-based RL algorithms depends on the coordinate system used to define the objective function. It is one of the most significant drawbacks of the standard gradient [26].

Moreover, in many applications such as robotics, the state space contains angles, so the state space has manifolds (curvatures). The objective function will then be defined in that curved space, making the policy gradients methods inefficient. Thus, we require a method that incorporates the knowledge about curvature of the space into the gradient. The natural gradients are the most "natural" choices in such cases.

2.1 Natural Gradients and the Fisher Information Matrix

For single agent actor-critic methods, the natural gradients of the objective function $J(\theta)$ are defined in [7, 39] as $\widetilde{\nabla}_\theta J(\theta) = G(\theta)^{-1}\nabla_\theta J(\theta)$, where $G(\theta) := \mathbb{E}[\nabla_\theta \log \pi_\theta(s,a)\nabla_\theta \log \pi_\theta(s,a)^\top]$ is the Fisher information matrix, and $\nabla J(\theta)$ is the standard gradient. The Fisher information matrix is the covariance of score function. It can also be interpreted via KL divergence between the policy $\pi(\cdot,\cdot)$ parameterize at θ and $\theta + \Delta\theta$ as [31, 42]

$$KL(\pi_\theta(\cdot,\cdot)||\pi_{\theta+\Delta\theta}(\cdot,\cdot)) \approx \frac{1}{2}\Delta\theta^\top \cdot G(\theta) \cdot \Delta\theta. \tag{5}$$

The above expression is obtained from the second-order Taylor expansion of $\log \pi_{\theta+\Delta\theta}(s,a)$, and using the fact that the sum of the probabilities is one. In above, the right-hand term is a quadratic involving positive definite matrix $G(\theta)$, and hence $G(\theta)$ approximately captures the curvature of the KL divergence between policy distributions at θ and $\theta + \Delta\theta$.

Lemma 1 *The gradient of the KL divergence between two consecutive policies is approximately proportional to the gradient of the objective function, i.e.,* $\nabla KL(\pi_{\theta_t}(\cdot,\cdot)||\pi_{\theta_t+\Delta\theta_t}(\cdot,\cdot)) \propto \nabla J(\theta_t)$.

Proof From Eq. (5), the KL divergence is a function of the Fisher information matrix and delta change in the policy parameters.

We find the optimal step-size $\Delta\theta_t^\star$ via the following optimization problem:

$$\Delta\theta_t^\star = argmax_{\Delta\theta_t}\ J(\theta_t + \Delta\theta_t) \quad s.t. \quad KL(\pi_{\theta_t}(\cdot,\cdot)||\pi_{\theta_t+\Delta\theta_t}(\cdot,\cdot)) = c.$$

Writing the Lagrange function $\mathcal{L}(\theta_t + \Delta\theta_t; \rho_t)$ (where ρ_t is the Lagrangian multiplier) of the above optimization problem and using the first-order Taylor approximation along with the KL divergence approximation as given in Eq. (5), we have

$$\begin{aligned}\Delta\theta_t^\star &= argmax_{\Delta\theta_t}\ J(\theta_t + \Delta\theta_t) + \rho_t(KL(\pi_{\theta_t}(\cdot,\cdot)||\pi_{\theta_t+\Delta\theta_t}(\cdot,\cdot)) - c)\\ &\approx argmax_{\Delta\theta_t}\ J(\theta_t) + \Delta\theta_t^\top \nabla_\theta J(\theta_t) + \frac{1}{2}\cdot\rho_t\cdot\Delta\theta_t^\top\cdot G(\theta_t)\cdot\Delta\theta_t - \rho_t c.\end{aligned}$$

Setting the derivative (w.r.t. $\Delta\theta_t$) of Lagrangian to zero, we have $\nabla_\theta J(\theta_t) + \rho_t \cdot \Delta\theta_t^{\star\top} \cdot G(\theta_t) = 0 \implies \Delta\theta_t^\star = -\frac{1}{\rho_t}G(\theta_t)^{-1}\nabla J(\theta_t)$, i.e., upto the factor of $-\frac{1}{\rho_t}$, we get an

optimal step-size in terms of the standard gradients and the Fisher information matrix at point θ_t. Moreover, from Eq. (5), we have $\nabla KL(\pi_{\theta_t}(\cdot,\cdot)||\pi_{\theta_t+\Delta\theta_t}(\cdot,\cdot)) \approx G(\theta_t)\Delta\theta_t^\star$ and $G(\theta_t)\Delta\theta_t^\star = -\frac{1}{\rho_t}\nabla J(\theta_t)$. Hence, $\nabla KL(\pi_{\theta_t}(\cdot,\cdot)||\pi_{\theta_t+\Delta\theta_t}(\cdot,\cdot)) \approx -\frac{1}{\rho_t}\nabla J(\theta_t)$. This ends the proof. □

The above lemma relates the gradient of the objective function to the gradient of the KL divergence between the policies separated by $\Delta\theta_t$. It provides a valuable observation because we can adjust the updates (of actor parameter θ_t) just by moving in the prediction space of the parameterized policy distributions. Thus, those MAN algorithms discussed later that rely on Fisher information matrix $G(\cdot)$ *implicitly use the above representation* for $\nabla J(\cdot)$. We recall these aspects in Sect. 3.6 for the Boltzmann policies.

2.2 Multi-Agent Natural Policy Gradient Theorem and Rank-One Update of $G_{t+1}^{i^{-1}}$

In this section, we provide some details of natural policy gradient methods and the Fisher information matrix in the multi-agent setup. Similar to single agent setup, in a multi-agent model the natural gradient of the objective function is $\widetilde{\nabla}_{\theta^i} J(\theta) = G(\theta^i)^{-1}\nabla_{\theta^i} J(\theta),\ \forall\, i \in N$, where the Fisher information matrix $G(\theta^i) := \mathbb{E}_{s\sim d_\theta, a\sim\pi_\theta}[\nabla_{\theta^i}\log\pi_{\theta^i}^i(s,a^i)\nabla_{\theta^i}\log\pi_{\theta^i}^i(s,a^i)^\top]$ is a positive definite matrix for each agent $i \in N$. We now present the policy gradient theorem for multi-agent setup involving the natural gradients.

Theorem 2 (Policy gradient theorem for MARL with natural gradients) *Under assumption X. 1, the natural gradient of $J(\theta)$ with respect to θ^i for each $i \in N$ is*

$$\begin{aligned}\widetilde{\nabla}_{\theta^i} J(\theta) &= G(\theta^i)^{-1}\mathbb{E}_{s\sim d_\theta, a\sim\pi_\theta}[\nabla_{\theta^i}\log\pi_{\theta^i}^i(s,a^i)\cdot A_\theta(s,a)]\\ &= G(\theta^i)^{-1}\mathbb{E}_{s\sim d_\theta, a\sim\pi_\theta}[\nabla_{\theta^i}\log\pi_{\theta^i}^i(s,a^i)\cdot A_\theta^i(s,a)].\end{aligned}$$

The proof follows from the multi-agent policy gradient Theorem 1 and definition of natural gradients, i.e., $\widetilde{\nabla}_{\theta^i} J(\theta) = G(\theta^i)^{-1}\nabla_{\theta^i} J(\theta),\ \forall\, i \in N$.

It is known that inverting a Fisher information matrix is computationally heavy [26, 40], whereas in our natural gradient-based multi-agent actor-critic methods, we require $G(\theta^i)^{-1},\ \forall\, i \in N$. To this end, for every $t > 0$, let $G_{t+1}^i = \frac{1}{t+1}\sum_{l=0}^t \psi_l\psi_l^\top$ be the sample average of the Fisher information matrix $G(\theta^i)$ by agent $i \in N$. Using the Sherman–Morrison–Woodbury matrix inversion [43] (see also [7, 37]), we recursively estimate the $G(\theta^i)^{-1}$ for each agent $i \in N$ at the faster time scale (more details are available in Sect. 2.2 in [49]).

$$G_{t+1}^{i^{-1}} = \frac{1}{1-\beta_{v,t}}\left[G_t^{i^{-1}} - \beta_{v,t}\frac{(G_t^{i^{-1}}\psi_t^i)(G_t^{i^{-1}}\psi_t^i)^\top}{1-\beta_{v,t}+\beta_{v,t}\psi_t^{i^\top}G_t^{i^{-1}}\psi_t^i}\right]. \tag{6}$$

The following section provides three multi-agent natural actor-critic (MAN) RL algorithms involving consensus matrices. Moreover, we will also investigate the relations among these algorithms and their effect on the quality of the local optima they attain.

3 Multi-Agent Natural Actor-Critic (MAN) Algorithms

This section provides three multi-agent natural actor-critic (MAN) reinforcement learning algorithms. Two of the three MAN algorithms explicitly use the Fisher information matrix inverse, whereas one uses the linear function approximation of the advantage parameters.

3.1 FI-MAN: Fisher Information Based Multi-Agent Natural Actor-Critic Algorithm

Our first multi-agent natural actor-critic algorithm uses the fact that natural gradients can be obtained via the Fisher information matrix and the standard gradients. The updates of the objective function estimate, critic, and the rewards parameters in FI-MAN algorithm are the same as given in Eqs. (2), (3), and (4), respectively. The major difference between the MAAC and the FI-MAN algorithm is in the actor update. FI-MAN algorithm uses the following actor update:

$$\theta^i_{t+1} \leftarrow \theta^i_t + \beta_{\theta,t} \cdot G^{i^{-1}}_t \cdot \tilde{\delta}^i_t \cdot \psi^i_t, \ \forall\, i \in N.$$

FI-MAN: Fisher information based multi-agent natural actor critic

Input: Initial values of $\mu^i_0, \tilde{\mu}^i_0, v^i_0, \tilde{v}^i_0, \lambda^i_0, \tilde{\lambda}^i_0, \theta^i_0, G^{i^{-1}}_0, \ \forall\, i \in N$; initial state s_0; stepsizes $\{\beta_{v,t}\}_{t\geq 0}, \{\beta_{\theta,t}\}_{t\geq 0}$.

Each agent i implements $a^i_0 \sim \pi_{\theta^i_0}(s_0, \cdot)$.

Initialize the step counter $t \leftarrow 0$.

repeat

for *all* $i \in N$ **do**

Observe state s_{t+1}, and reward r^i_{t+1}.

Update: $\tilde{\mu}^i_t \leftarrow (1-\beta_{v,t}) \cdot \mu^i_t + \beta_{v,t} \cdot r^i_{t+1}$.

$\tilde{\lambda}^i_t \leftarrow \lambda^i_t + \beta_{v,t} \cdot [r^i_{t+1} - \bar{R}_t(\lambda^i_t)] \cdot \nabla_\lambda \bar{R}_t(\lambda^i_t)$, where $\bar{R}_t(\lambda^i_t) = \lambda^{i^\top}_t f(s_t, a_t)$.

Update: $\delta^i_t \leftarrow r^i_{t+1} - \mu^i_t + V_{t+1}(v^i_t) - V_t(v^i_t)$, where $V_{t+1}(v^i_t) = v^{i^\top}_t \varphi(s_{t+1})$.

Critic Step: $\tilde{v}^i_t \leftarrow v^i_t + \beta_{v,t} \cdot \delta^i_t \cdot \nabla_v V_t(v^i_t)$,

Update: $\tilde{\delta}^i_t \leftarrow \bar{R}_t(\lambda^i_t) - \mu^i_t + V_{t+1}(v^i_t) - V_t(v^i_t)$; $\psi^i_t \leftarrow \nabla_{\theta^i} \log \pi^i_{\theta^i_t}(s_t, a^i_t)$.

Actor Step: $\theta^i_{t+1} \leftarrow \theta^i_t + \beta_{\theta,t} \cdot G^{i^{-1}}_t \cdot \tilde{\delta}^i_t \cdot \psi^i_t$.

Send $\tilde{\mu}^i_t, \tilde{\lambda}^i_t, \tilde{v}^i_t$ to the neighbors over $\mathcal{G}_t$.

for *all* $i \in N$ **do**

Consensus Update: $\mu^i_{t+1} \leftarrow \sum_{j\in N} c_t(i,j)\tilde{\mu}^j_t$;

$\lambda^i_{t+1} \leftarrow \sum_{j\in N} c_t(i,j)\tilde{\lambda}^j_t$; $v^i_{t+1} \leftarrow \sum_{j\in N} c_t(i,j)\tilde{v}^j_t$.

Fisher Update: $G^{i^{-1}}_{t+1} \leftarrow \frac{1}{1-\beta_{v,t}} \left[G^{i^{-1}}_t - \beta_{v,t} \frac{(G^{i^{-1}}_t \psi^i_t)(G^{i^{-1}}_t \psi^i_t)^\top}{1-\beta_{v,t}+\beta_{v,t}\psi^{i^\top}_t G^{i^{-1}}_t \psi^i_t} \right]$.

Update: $t \leftarrow t+1$.

until *Convergence*;

FI-MAN algorithm explicitly uses $G^{i^{-1}}_t$ in the actor update. Though the Fisher information inverse matrix is updated according to the Sherman–Morrison inverse at a faster time scale, it may be better to avoid explicit use of the Fisher inverse in the actor update. To this end, we use the linear function approximation of the advantage function. This leads to the AP-MAN algorithm, i.e., advantage parameters based multi-agent natural actor-critic algorithm.

3.2 AP-MAN: Advantage Parameters-Based Multi-Agent Natural Actor Critic Algorithm

Consider the local advantage function $A^i(s,a^i) : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ for each agent $i \in N$. Let the local advantage function $A^i(s,a^i)$ be linearly approximated as $A^i(s,a^i;w^i) := {w^i}^\top \psi^i(s,a^i)$, where $\psi^i(s,a^i) = \nabla_{\theta^i} \log \pi^i_{\theta^i}(s,a^i)$ are the compatible features, and $w^i \in \mathbb{R}^{m_i}$ are the advantage function parameters. Recall, the same $\psi^i(s,a^i)$ was used to represent the score function in the policy gradient theorem, Theorem 1. However, it also serves as the compatible feature while approximating the advantage function as it satisfies the compatibility condition in the policy gradient theorem with linear function approximations (Theorem 2 [46]). The compatibility condition as given in [46] is for single agent; however, we are using it explicitly for each agent $i \in N$. Whenever there is no confusion, we write ψ^i instead of $\psi^i(s,a^i)$, to save space.

We can tune w^i in such a way that the estimate of least squared error in linear function approximation of advantage function is minimized, i.e., $\mathcal{E}^{\pi_\theta}(w^i) = \frac{1}{2}\sum_{s\in\mathcal{S},a^i\in\mathcal{A}^i} \tilde{d}_\theta(s,a^i)[{w^i}^\top\psi^i(s,a^i) - A^i(s,a^i)]^2$ is minimized. Here, $\tilde{d}_\theta(s,a^i) = d_\theta(s)\cdot\pi^i_{\theta^i}(s,a^i)$ as defined earlier. Taking the derivative of above equation w.r.t w^i, we have $\nabla_{w^i}\mathcal{E}^{\pi_\theta}(w^i) = \sum_{s\in\mathcal{S},a^i\in\mathcal{A}^i}\tilde{d}_\theta(s,a^i)[{w^i}^\top\psi^i - A^i(s,a^i)]\psi^i$. Noting that parameterized TD error $\tilde{\delta}^i_t$ is an unbiased estimate of the local advantage function $A^i(s,a^i)$, we will use $\widehat{\nabla_{w^i}}\mathcal{E}^{\pi_\theta}(w^i_t) = \psi^i_t{\psi^i_t}^\top w^i_t - \tilde{\delta}^i_t\psi^i_t$ as an estimate of $\nabla_{w^i}\mathcal{E}^{\pi_\theta}(w^i)$. Hence, the update of advantage parameter w^i in the AP-MAN algorithm is $w^i_{t+1} = w^i_t - \beta_{v,t}\widehat{\nabla_{w^i}}\mathcal{E}^{\pi_\theta}(w^i_t) = (I - \beta_{v,t}\psi^i_t{\psi^i_t}^\top)w^i_t + \beta_{v,t}\tilde{\delta}^i_t\psi^i_t$. The updates of the objective function estimate, critic, and reward parameters in the AP-MAN algorithm are the same as given in Eqs. (2), (3), and (4), respectively. Additionally, in the critic step, we update the advantage parameters as given above. For single-agent RL with natural gradients [39, 40], show that $\widetilde{\nabla}_\theta J(\theta) = w$. In MARL with natural gradient, we separately verified and hence use $\widetilde{\nabla}_{\theta^i} J(\theta) = w^i$ for each agent $i \in N$ in the actor update of AP-MAN algorithm. The AP-MAN actor-critic algorithm thus uses $\theta^i_{t+1} \leftarrow \theta^i_t + \beta_{\theta,t}\cdot w^i_{t+1}$ in the actor update. The algorithm's pseudo-code involving advantage parameters is given in the AP-MAN algorithm.

Remark 1 We want to emphasize that the AP-MAN algorithm does not explicitly use $G(\theta^i)^{-1}$ in the actor update (as also in [7]); hence, it requires fewer computations. However, it involves the linear function approximation of the advantage function that itself requires $\psi^i_t{\psi^i_t}^\top$ which is an unbiased estimate of the Fisher information matrix. We will see later in Sect. 3.4 that the performance of the AP-MAN algorithm is almost the same as the MAAC algorithm. We empirically verify this observation in Sect. 5.

Remark 2 The advantage function is a linear combination of $Q_\theta(s,a)$ and $V_\theta(s)$; therefore, the linear function approximation of the advantage function alone enjoys the benefit of approximating the $Q_\theta(s,a)$ or $V_\theta(s)$. Moreover, MAAC uses the linear function approximation of $V_\theta(s)$; hence, we expect the behavior of AP-MAN to be similar to that of MAAC; this comes out in our computational experiments in Sect. 5.

The FI-MAN algorithm is based solely on the Fisher information matrix and the AP-MAN algorithm on the advantage function approximation. Our next algorithm, FIAP-MAN algorithm, i.e., Fisher information and advantage parameter-based multi-agent natural actor-critic algorithm combines them in a certain way. We see the benefits of this combination in Sects. 3.4 and 5.1.

AP-MAN: Advantage parameters based multi-agent natural actor critic

Input: Initial values of $\mu_0^i, \tilde{\mu}_0^i, v_0^i, \tilde{v}_0^i, \lambda_0^i, \tilde{\lambda}_0^i, \theta_0^i, w_0^i, \forall i \in N$; initial state s_0; stepsizes $\{\beta_{v,t}\}_{t\geq 0}, \{\beta_{\theta,t}\}_{t\geq 0}$.

Each agent i implements $a_0^i \sim \pi_{\theta_0^i}(s_0, \cdot)$.

Initialize the step counter $t \leftarrow 0$.

repeat

  for *all* $i \in N$ **do**

    Observe state s_{t+1}, and reward r_{t+1}^i.

    Update: $\tilde{\mu}_t^i \leftarrow (1-\beta_{v,t})\cdot \mu_t^i + \beta_{v,t}\cdot r_{t+1}^i$.

    $\tilde{\lambda}_t^i \leftarrow \lambda_t^i + \beta_{v,t}\cdot [r_{t+1}^i - \bar{R}_t(\lambda_t^i)]\cdot \nabla_\lambda \bar{R}_t(\lambda_t^i)$, where $\bar{R}_t(\lambda_t^i) = {\lambda_t^i}^\top f(s_t, a_t)$.

    Update: $\delta_t^i \leftarrow r_{t+1}^i - \mu_t^i + V_{t+1}(v_t^i) - V_t(v_t^i)$, where $V_{t+1}(v_t^i) = {v_t^i}^\top \varphi(s_{t+1})$.

    Critic Step: $\tilde{v}_t^i \leftarrow v_t^i + \beta_{v,t}\cdot \delta_t^i \cdot \nabla_v V_t(v_t^i)$.

    Update: $\tilde{\delta}_t^i \leftarrow \bar{R}_t(\lambda_t^i) - \mu_t^i + V_{t+1}(v_t^i) - V_t(v_t^i)$; $\psi_t^i \leftarrow \nabla_{\theta^i} \log \pi_{\theta_t^i}^i(s_t, a_t^i)$.

    Update: $w_{t+1}^i \leftarrow (I - \beta_{v,t}\psi_t^i {\psi_t^i}^\top) w_t^i + \beta_{v,t}\tilde{\delta}_t^i \psi_t^i$.

    Actor Step: $\theta_{t+1}^i \leftarrow \theta_t^i + \beta_{\theta,t}\cdot w_{t+1}^i$.

    Send $\tilde{\mu}_t^i, \tilde{\lambda}_t^i, \tilde{v}_t^i$ to the neighbors over $\mathcal{G}_t$.

  for *all* $i \in N$ **do**

    Consensus Update: $\mu_{t+1}^i \leftarrow \sum_{j\in N} c_t(i,j)\tilde{\mu}_t^j$;

    $\lambda_{t+1}^i \leftarrow \sum_{j\in N} c_t(i,j)\tilde{\lambda}_t^j$; $v_{t+1}^i \leftarrow \sum_{j\in N} c_t(i,j)\tilde{v}_t^j$.

  Update: $t \leftarrow t+1$.

until *Convergence*;

3.3 FIAP-MAN: Fisher Information and Advantage Parameter Based Multi-Agent Natural Actor-Critic Algorithm

Recall in Sect. 3.2, for each agent $i \in N$, the local advantage function has linear function approximation $A^i(s, a^i; w^i) = {w^i}^\top \psi^i(s, a^i)$, where $\psi^i(s, a^i)$ are the compatible features as before, and $w^i \in \mathbb{R}^{m_i}$ are the advantage function parameters. In AP-MAN algorithm the Fisher inverse $G(\theta^i)^{-1}$ is not estimated explicitly; however, in FIAP-MAN algorithm, we explicitly estimate $G(\theta^i)^{-1}$, and hence use $\widehat{\nabla_{w^i}} \mathcal{E}^{\pi_\theta}(w_t^i) = G_t^{i^{-1}}(\psi_t^i {\psi_t^i}^\top w_t^i - \tilde{\delta}_t^i \psi_t^i)$, $\forall\, i \in N$ as an estimate of $\nabla_{w^i} \mathcal{E}^{\pi_\theta}(w^i)$. The update of advantage parameters w^i along with the critic update in the FIAP-MAN algorithm is $w_{t+1}^i = w_t^i - \beta_{v,t}\widehat{\nabla_{w^i}} \mathcal{E}^{\pi_\theta}(w_t^i) = w_t^i - \beta_{v,t} G_t^{i^{-1}}(\psi_t^i {\psi_t^i}^\top w_t^i - \tilde{\delta}_t^i \psi_t^i) = (1-\beta_{v,t}) w_t^i + \beta_{v,t} G_t^{i^{-1}} \tilde{\delta}_t^i \psi_t^i$.

Remark 3 Note that we take $G_t^{i^{-1}} \psi_t^i {\psi_t^i}^\top = I$, $\forall\, i \in N$, though $G_{t+1} = \frac{1}{t+1}\sum_{l=0}^t \psi_l {\psi_l}^\top$. A similar approximation is also implicitly made in natural gradient algorithms in [7, 8] for single-agent RL. Convergence of FIAP-MAN algorithm with above approximate update in MARL is given in Sect. 4. Later, we use these updates in our computations to demonstrate their superior performance in multiple instances of traffic network (Sect. 5).

The updates of the objective function estimate, critic, and reward parameters in the FIAP-MAN algorithm are the same as given in Eqs. (2), (3), and (4), respectively. Similar to the AP-MAN algorithm, the actor update in FIAP-MAN algorithm is $\theta_{t+1}^i \leftarrow \theta_t^i + \beta_{\theta,t}\cdot w_{t+1}^i$, $\forall\, i \in N$. Again for the same reason as in the AP-MAN algorithm, we take $\widetilde{\nabla}_{\theta^i} J(\theta) = w^i$.

FIAP-MAN: Fisher information and advantage parameters based multi-agent natural actor-critic

Input: Initial values of $\mu_0^i, \tilde{\mu}_0^i, v_0^i, \tilde{v}_0^i, \lambda_0^i, \tilde{\lambda}_0^i, \theta_0^i, w_0^i, G_0^{i^{-1}}$, $\forall\, i \in N$; initial state s_0; stepsizes $\{\beta_{v,t}\}_{t\geq 0}, \{\beta_{\theta,t}\}_{t\geq 0}$.

Each agent i implements $a_0^i \sim \pi_{\theta_0^i}(s_0, \cdot)$.

Initialize the step counter $t \leftarrow 0$.

repeat

  for *all* $i \in N$ **do**

    Observe state s_{t+1}, and reward r_{t+1}^i.

    Update: $\tilde{\mu}_t^i \leftarrow (1-\beta_{v,t})\cdot \mu_t^i + \beta_{v,t}\cdot r_{t+1}^i$.

    $\tilde{\lambda}_t^i \leftarrow \lambda_t^i + \beta_{v,t}\cdot[r_{t+1}^i - \bar{R}_t(\lambda_t^i)]\cdot \nabla_\lambda \bar{R}_t(\lambda_t^i)$, where $\bar{R}_t(\lambda_t^i) = \lambda_t^{i^\top} f(s_t, a_t)$.

    Update: $\delta_t^i \leftarrow r_{t+1}^i - \mu_t^i + V_{t+1}(v_t^i) - V_t(v_t^i)$, where $V_{t+1}(v_t^i) = v_t^{i^\top}\varphi(s_{t+1})$.

    Critic Step: $\tilde{v}_t^i \leftarrow v_t^i + \beta_{v,t}\cdot \delta_t^i \cdot \nabla_v V_t(v_t^i)$.

    Update: $\tilde{\delta}_t^i \leftarrow \bar{R}_t(\lambda_t^i) - \mu_t^i + V_{t+1}(v_t^i) - V_t(v_t^i)$; $\psi_t^i \leftarrow \nabla_{\theta^i} \log \pi_{\theta_t^i}^i(s_t, a_t^i)$.

    Update: $w_{t+1}^i \leftarrow (1-\beta_{v,t}) w_t^i + \beta_{v,t} G_t^{i^{-1}} \tilde{\delta}_t^i \psi_t^i$.

    Actor Step: $\theta_{t+1}^i \leftarrow \theta_t^i + \beta_{\theta,t}\cdot w_{t+1}^i$.

    Send $\tilde{\mu}_t^i, \tilde{\lambda}_t^i, \tilde{v}_t^i$ to the neighbors over $\mathcal{G}_t$.

  for *all* $i \in N$ **do**

    Consensus Update: $\mu_{t+1}^i \leftarrow \sum_{j\in N} c_t(i,j)\tilde{\mu}_t^j$;

    $\lambda_{t+1}^i \leftarrow \sum_{j\in N} c_t(i,j)\tilde{\lambda}_t^j$; $v_{t+1}^i \leftarrow \sum_{j\in N} c_t(i,j)\tilde{v}_t^j$.

    Fisher Update: $G_{t+1}^{i^{-1}} \leftarrow \frac{1}{1-\beta_{v,t}}\left[G_t^{i^{-1}} - \beta_{v,t}\frac{(G_t^{i^{-1}}\psi_t^i)(G_t^{i^{-1}}\psi_t^i)^\top}{1-\beta_{v,t}+\beta_{v,t}\psi_t^{i^\top} G_t^{i^{-1}}\psi_t^i}\right]$.

  Update: $t \leftarrow t+1$.

until *Convergence*;

3.4 Relationship Between Actor Updates in Algorithms

Recall, the actor update for each agent $i \in N$ in FIAP-MAN algorithm is $\theta_{t+1}^i = \theta_t^i + \beta_{\theta,t} w_{t+1}^i$, where $w_{t+1}^i = (1-\beta_{v,t}) w_t^i + \beta_{v,t} G_t^{i^{-1}} \tilde{\delta}_t^i \psi_t^i$. Therefore, the actor update of FIAP-MAN algorithm is $\theta_{t+1}^i = \theta_t^i + \beta_{\theta,t}(1-\beta_{v,t}) w_t^i + \beta_{v,t}\left(\beta_{\theta,t} G_t^{i^{-1}} \tilde{\delta}_t^i \psi_t^i\right)$. The above update is almost the same as the actor update of the FI-MAN algorithm with an additional term involving advantage parameter w_t^i. However, the contribution of the second term is negligible after some time t. Moreover, the third term is a positive fraction of the second term in the actor update of FI-MAN algorithm. Therefore, the actor parameters in FIAP-MAN and FI-MAN algorithms are almost the same after time t. Hence, both the algorithms are expected to converge almost to the same local optima.

Similarly, consider the actor update of the AP-MAN algorithm, i.e., $\theta_{t+1}^i = \theta_t^i + \beta_{\theta,t} w_{t+1}^i$, where $w_{t+1}^i = (I - \beta_{v,t}\psi_t^i \psi_t^{i^\top}) w_t^i + \beta_{v,t}\tilde{\delta}_t^i \psi_t^i$. Therefore, the actor update of AP-MAN algorithm is $\theta_{t+1}^i = \theta_t^i + \beta_{\theta,t}(I - \beta_{v,t}\psi_t^i \psi_t^{i^\top}) w_t^i + \beta_{v,t}\left(\beta_{\theta,t}\tilde{\delta}_t^i \psi_t^i\right)$. Again, the second term in the above equation is negligible after some time t, and the third term is a positive fraction of the second term in the actor update of the MAAC algorithm. Hence, the actor update in AP-MAN algorithm is almost the same as the MAAC algorithm; therefore, AP-MAN and MAAC are expected to converge to the same local optima.

3.5 Comparison of Variants of MAN and MAAC Algorithms

In this section, we show that under some conditions the objective function of a variant of the FI-MAN algorithm dominates that of the corresponding variant of the MAAC algorithm for all $t \geq t_0$, for some $t_0 < \infty$. For this purpose, we propose a model to evaluate the "efficiency" of MAAC and FI-MAN algorithms in terms of their goal; maximization of MARL objective function, $J(\theta)$. This comparison exploits the intrinsic property of the Fisher information matrix $G(\theta)$ (Lemma 2, an uniform upper bound on its minimum eigenvalue).

Let θ_t^M and θ_t^N be the actor parameters in MAAC and FI-MAN algorithms, respectively. Recall the actor updates in MAAC and FI-MAN algorithms were $\theta_{t+1}^M = \theta_t^M + \beta_{\theta,t}\tilde{\delta}_t \cdot \psi_t$, and $\theta_{t+1}^N = \theta_t^N + \beta_{\theta,t} G_t^{-1}\tilde{\delta}_t \cdot \psi_t$, respectively. However, these updates use the biased estimate of $\nabla J(\cdot)$. Moreover, the Fisher information matrix inverse is updated via the Sherman–Morrison iterative method. In this section, we work with the deterministic variants of these algorithms where we use $\nabla J(\cdot)$ instead of $\tilde{\delta}_t \cdot \psi_t$, and $G(\theta_t^N)^{-1}$ instead of using G_t^{-1} in the actor updates. This avoids the approximation errors; however, the same is not possible in the computations since the gradient of the objective function is not known. For ease of notation, we denote the actor parameters in the deterministic variants of MAAC and FI-MAN algorithms by $\tilde{\theta}^M$, and $\tilde{\theta}^N$, respectively. In particular, we consider the following actor updates.

$$\begin{aligned} &\textbf{Deterministic MAAC } \tilde{\theta}_{t+1}^M = \tilde{\theta}_t^M + \beta_{\tilde{\theta},t} \nabla J(\tilde{\theta}_t^M); \\ &\textbf{Deterministic FI-MAN } \tilde{\theta}_{t+1}^N = \tilde{\theta}_t^N + \beta_{\tilde{\theta},t} G(\tilde{\theta}_t^N)^{-1} \nabla J(\tilde{\theta}_t^N). \end{aligned} \tag{7}$$

We give sufficient conditions when the objective function value of the limit point of the deterministic FI-MAN algorithm is not worse off than the value by deterministic MAAC algorithm while using the above actor updates. We want to emphasize that with these updates, the actor parameters in Eq. (7) will converge to a local maxima under some conditions (for example, the strong Wolfe's conditions) on the step-size [37]. Let $\tilde{\theta}^{M^\star}$ and $\tilde{\theta}^{N^\star}$ be the corresponding local maxima. The existence of the local maxima for the deterministic MAN algorithms is also guaranteed via the Wolfe's conditions in the natural gradients space. Note that these local maxima need not be the same as the one obtained from actor updates of MAAC and FI-MAN algorithms. However, the result given below may be valid for the MAAC and FI-MAN algorithms because both $\tilde{\delta} \cdot \psi$ and $\nabla J(\cdot)$ go to zero asymptotically.

We also assume that both algorithms use the same sequence of step-sizes, $\{\beta_{\tilde{\theta},t}\}$. The results in this section uses Taylor's series expansion and comparison of the objective function, $J(\cdot)$, rather than its estimate μ. Similar ideas are used in [4, 38] where the certainty equivalence principle holds, i.e., the random variables are replaced by their expected values. However, we work with the estimates in convergence theory/proofs and computations since the value of global objective is unknown to the agents. We first bound the minimum singular value of the Fisher information matrix in the following Lemma.

Lemma 2 *For $G(\theta) = \mathbb{E}[\psi\psi^\top]$, such that $||\psi|| \leq 1$, the minimum singular value $\sigma_{\min}(G(\theta))$ is upper bounded by $\frac{1}{m}$, i.e., $\sigma_{\min}(G(\theta)) \leq \frac{1}{m}$.*

The proof of this Lemma is based on the observation that the trace of matrix $\psi\psi^\top$ is $||\psi||^2$. Though this is a new result, we defer its detailed proof to Appendix A. 3 of [49] due to space considerations.

Remark 4 In the literature, the compatible features are assumed to be uniformly bounded, Assumption X. 3. For the linear architecture of features that we are using, assuming this bound to be 1 is not restrictive. The features ψ that we use in our computational experiments in Sect. 5 automatically meet the condition of being normalized by 1, i.e., $||\psi|| \leq 1$.

Lemma 3 *Let $J(\cdot)$ be twice continuously differentiable function on a compact set Θ, so that $|\{\nabla^2 J(\tilde{\theta}_t^M)\}_{(i,j)}| \leq H,\ \forall\, i,j \in [m]$ for some $H < \infty$. Moreover, let $J(\tilde{\theta}_t^M) \leq J(\tilde{\theta}_t^N)$, $||\nabla J(\tilde{\theta}_t^M)|| \leq ||\nabla J(\tilde{\theta}_t^N)||$, and $\beta_{\tilde{\theta},t}\frac{mH}{2} + 1 - m^2 \leq 0$. Then, $J(\tilde{\theta}_{t+1}^M) \leq J(\tilde{\theta}_{t+1}^N)$.*

Proof The Taylor series expansion of a twice differentiable function $J(\tilde{\theta}_{t+1}^M)$ with Lagrange form of remainder [22] is $J(\tilde{\theta}_{t+1}^M) = J(\tilde{\theta}_t^M + \Delta\tilde{\theta}_t^M) = J(\tilde{\theta}_t^M) + \Delta\tilde{\theta}_t^{M^\top}\nabla J(\tilde{\theta}_t^M) + R_M(\Delta\tilde{\theta}_t^M)$, where $R_M(\Delta\tilde{\theta}_t^M) = \frac{1}{2!}\Delta\tilde{\theta}_t^{M^\top}\nabla^2 J(\tilde{\theta}_t^M + c_M \cdot \Delta\tilde{\theta}_t^M)\Delta\tilde{\theta}_t^M$ for some $c_M \in (0,1)$.

Similarly, the Taylor series expansion of $J(\tilde{\theta}_{t+1}^N)$ with Lagrange remainder form is $J(\tilde{\theta}_{t+1}^N) = J(\tilde{\theta}_t^N + \Delta\tilde{\theta}_t^N) = J(\tilde{\theta}_t^N) + \Delta\tilde{\theta}_t^{N^\top}\widetilde{\nabla} J(\tilde{\theta}_t^N) + R_N(\Delta\tilde{\theta}_t^N)$, where $R_N(\Delta\tilde{\theta}_t^N) = \frac{1}{2!}\Delta\tilde{\theta}_t^{N^\top}\widetilde{\nabla}^2 J(\tilde{\theta}_t^N + c_N \cdot \Delta\tilde{\theta}_t^N)\Delta\tilde{\theta}_t^N$ for some $c_N \in (0,1)$.

Now, consider the difference

$$
\begin{aligned}
&J(\tilde{\theta}_{t+1}^M) - J(\tilde{\theta}_{t+1}^N)\\
&= J(\tilde{\theta}_t^M) - J(\tilde{\theta}_t^N) + \Delta\tilde{\theta}_t^{M^\top}\nabla J(\tilde{\theta}_t^M)\\
&\quad -\Delta\tilde{\theta}_t^{N^\top}\widetilde{\nabla} J(\tilde{\theta}_t^N) + R_M(\Delta\tilde{\theta}_t^M) - R_N(\Delta\tilde{\theta}_t^N)\\
&\overset{(i)}{\leq} \Delta\tilde{\theta}_t^{M^\top}\nabla J(\tilde{\theta}_t^M) - \Delta\tilde{\theta}_t^{N^\top} G(\tilde{\theta}_t^N)^{-1}\nabla J(\tilde{\theta}_t^N) + R_M(\Delta\tilde{\theta}_t^M)\\
&\overset{(ii)}{=} (\tilde{\theta}_{t+1}^M - \tilde{\theta}_t^M)^\top J(\tilde{\theta}_t^M) - (\tilde{\theta}_{t+1}^N - \tilde{\theta}_t^N)^\top G(\tilde{\theta}_t^N)^{-1}\nabla J(\tilde{\theta}_t^N) + R_M(\Delta\tilde{\theta}_t^M)\\
&\overset{(iii)}{=} \beta_{\tilde{\theta},t}\left(||\nabla J(\tilde{\theta}_t^M)||^2 - ||G(\tilde{\theta}_t^N)^{-1}\nabla J(\tilde{\theta}_t^N))||^2\right) + R_M(\Delta\tilde{\theta}_t^M)\\
&\overset{(iv)}{\leq} \beta_{\tilde{\theta},t}\left(||\nabla J(\tilde{\theta}_t^N)||^2 - ||G(\tilde{\theta}_t^N)^{-1}\nabla J(\tilde{\theta}_t^N))||^2\right) + R_M(\Delta\tilde{\theta}_t^M),
\end{aligned}
\tag{8}
$$

where (i) follows because $J(\tilde{\theta}_t^M) \leq J(\tilde{\theta}_t^N)$, $R_N(\Delta\tilde{\theta}_t^N) \geq 0$ and $\widetilde{\nabla} J(\tilde{\theta}_t^N) = G(\tilde{\theta}_t^N)^{-1}\nabla J(\tilde{\theta}_t^N)$. (ii) uses the fact that $\Delta\tilde{\theta}_t^M = \tilde{\theta}_{t+1}^M - \tilde{\theta}_t^M$; $\Delta\tilde{\theta}_t^N = \tilde{\theta}_{t+1}^N - \tilde{\theta}_t^N$. (iii) is consequence of the updates in Eq. (7). Finally, (iv) follows from the fact that $||\nabla J(\tilde{\theta}_t^M)|| \leq ||\nabla J(\tilde{\theta}_t^N)||$.

Now, from Eq. (8) and using the fact that for any positive definite matrix $\mathbf{A}$ and a vector $\mathbf{v}$, $||\mathbf{Av}|| \geq \sigma_{\min}(\mathbf{A})||\mathbf{v}||$, we have $||G(\tilde{\theta}_t^N)^{-1}\nabla J(\tilde{\theta}_t^N))||^2 \geq \sigma_{\min}^2(G(\tilde{\theta}_t^N)^{-1})||\nabla J(\tilde{\theta}_t^N))||^2$. Therefore, from Eq. (8), we have

$$
\begin{aligned}
J(\tilde{\theta}_{t+1}^M) - J(\tilde{\theta}_{t+1}^N) &\leq \beta_{\tilde{\theta},t}\left(1 - \frac{1}{\sigma_{\min}^2(G(\tilde{\theta}_t^N))}\right)||\nabla J(\tilde{\theta}_t^N))||^2 + R_M(\Delta\tilde{\theta}_t^M)\\
&\overset{(v)}{\leq} \beta_{\tilde{\theta},t}\left(1 - m^2\right)||\nabla J(\tilde{\theta}_t^N))||^2 + R_M(\Delta\tilde{\theta}_t^M),
\end{aligned}
\tag{9}
$$

(v) follows from Lemma 2 as $\sigma_{\min}(G(\tilde{\theta}_t^N)) \leq \frac{1}{m}$, implies $-\frac{1}{\sigma_{\min}^2(G(\tilde{\theta}_t^N))} \leq -m^2$.

Since $J(\cdot)$ is twice continuously differentiable function on the compact set Θ, we have for all $i,j \in [m]$, $|\{\nabla^2 J(\tilde{\theta}_t^M\}_{(i,j)}| \leq H < \infty$. Therefore, we have $|R_M(\Delta\tilde{\theta}_t^M)| \leq \frac{H}{2!}||\Delta\tilde{\theta}_t^M||_1^2 \overset{(vii)}{=} \beta_{\tilde{\theta},t}^2\frac{H}{2}||\nabla J(\tilde{\theta}_t^M)||_1^2 \overset{(viii)}{\leq} \beta_{\tilde{\theta},t}^2\frac{mH}{2}||\nabla J(\tilde{\theta}_t^M)||^2 \overset{(ix)}{\leq} \beta_{\tilde{\theta},t}^2\frac{mH}{2}||\nabla J(\tilde{\theta}_t^N)||^2$, where (vii) follows from actor update of the FI-MAN algorithm. $(viii)$ holds because for any $\mathbf{x} \in \mathbb{R}^l$, the following is true:

$||\mathbf{x}||_2 \leq ||\mathbf{x}||_1 \leq \sqrt{l}\,||\mathbf{x}||_2$ [25]. (ix) comes from the assumption that $||\nabla J(\tilde{\theta}_t^M)||^2 \leq ||\nabla J(\tilde{\theta}_t^N)||^2$. From Eq. (9) and upper bound on $|R_M(\Delta\tilde{\theta}_t^M)|$, we have

$$J(\tilde{\theta}^M_{t+1}) - J(\tilde{\theta}^N_{t+1}) \leq \beta_{\tilde{\theta},t}\left(1-m^2\right)||\nabla J(\tilde{\theta}^N_t))||^2 + \beta^2_{\tilde{\theta},t}\frac{mH}{2}||\nabla J(\tilde{\theta}^N_t)||^2$$
$$= \beta_{\tilde{\theta},t}\left\{\beta_{\tilde{\theta},t}\frac{mH}{2} + 1 - m^2\right\}||\nabla J(\tilde{\theta}^N_t)||^2 \leq 0,$$

where the last inequality follows from the assumption $\beta_{\tilde{\theta},t}\frac{mH}{2} + 1 - m^2 \leq 0$. Therefore, $J(\tilde{\theta}^M_{t+1}) \leq J(\tilde{\theta}^N_{t+1})$. □

Theorem 3 *Let $J(\cdot)$ be twice continuously differentiable function on a compact set Θ, so that $|\{\nabla^2 J(\tilde{\theta}^M_t)\}_{(i,j)}| \leq H,\ \forall\, i, j \in [m]$ for some $H < \infty$. Moreover, let $J(\tilde{\theta}^M_{t_0}) \leq J(\tilde{\theta}^N_{t_0})$ for some $t_0 > 0$, and for every $t \geq t_0$, let $||\nabla J(\tilde{\theta}^M_t)|| \leq ||\nabla J(\tilde{\theta}^N_t)||$, and $\beta_{\tilde{\theta},t}\frac{mH}{2} + 1 - m^2 \leq 0$. Then, $J(\tilde{\theta}^M_t) \leq J(\tilde{\theta}^N_t)$, for all $t \geq t_0$. Further, for the local maxima $\tilde{\theta}^{M^\star}$, and $\tilde{\theta}^{N^\star}$ of the updates in Eq. (7), we have $J(\tilde{\theta}^{M^\star}) \leq J(\tilde{\theta}^{N^\star})$.*

Proof We prove this theorem via principle of mathematical induction (PMI). From assumption, we have $J(\tilde{\theta}^M_{t_0}) \leq J(\tilde{\theta}^N_{t_0})$. Now, using $t = t_0$ in Lemma 3, we have $J(\tilde{\theta}^M_{t_0+1}) \leq J(\tilde{\theta}^N_{t_0+1})$. Thus, the base case of PMI is true.

Next, we assume that $J(\tilde{\theta}^M_t) \leq J(\tilde{\theta}^N_t)$ for any $t = t_0 + k$, where $k \in \mathbb{Z}^+$. Also from assumption, for every $t \geq t_0 + k$, we have $||\nabla J(\tilde{\theta}^M_t)|| \leq ||\nabla J(\tilde{\theta}^N_t)||$, and $\beta_{\tilde{\theta},t}\frac{mH}{2} + 1 - m^2 \leq 0$. Therefore, again from Lemma 3, we have $J(\tilde{\theta}^M_{t_0+k+1}) \leq J(\tilde{\theta}^N_{t_0+k+1})$. From PMI, we have $J(\tilde{\theta}^M_t) \leq J(\tilde{\theta}^N_t),\ \ \forall\ t \geq t_0$.

Finally, consider the limiting case. Taking the limit $t \to \infty$ in the above equation and using the fact that $J(\cdot)$ is continuous on the compact set Θ, we have $lim_{t\to\infty} J(\tilde{\theta}^M_t) = J(\tilde{\theta}^{M^\star}),\ and\ lim_{t\to\infty} J(\tilde{\theta}^N_t) = J(\tilde{\theta}^{N^\star})$, so that $J(\tilde{\theta}^{M^\star}) \leq J(\tilde{\theta}^{N^\star})$. This ends the proof. □

3.6 KL Divergence-Based Natural Gradients for Boltzmann Policy

One specific policy that is often used in RL literature is the Boltzmann policy [20]. Recall, the parameterized Boltzmann policy is $\pi_{\theta_t}(s,a) = \frac{\exp(q^\top_{s,a}\theta_t)}{\sum_{b\in\mathcal{A}}\exp(q^\top_{s,b}\theta_t)}$, where $q^\top_{s,a}$ is the feature for any state-action pair (s,a). Here, the features $q_{s,a}$ are assumed to be uniformly bounded by 1.

Lemma 4 *For the Boltzmann policy, we have $KL(\pi_{\theta_t}(s,a)||\pi_{\theta_t+\Delta\theta_t}(s,a)) = \mathbb{E}\left[\log\left(\sum_{b\in\mathcal{A}}\pi_{\theta_t}(s,b)\exp(\Delta q^\top_{s,ba}\Delta\theta_t)\right)\right]$, where $\Delta q^\top_{s,ba} = q^\top_{s,b} - q^\top_{s,a}$.*

The proof of this Lemma just uses the definition of KL divergence and the Boltzmann policy. So, we defer it to Appendix A.4 of [49]. The above KL divergence suggests that we have a nonzero curvature if the action taken is better than the averaged action. Moreover, $\exp(\Delta q_{s,ba}\Delta\theta_t) \neq 1$ if and only if $\Delta q_{s,ba}$ is orthogonal to $\Delta\theta_t$. So, except when they are orthogonal, $\log(\sum_{b\in\mathcal{A}}\pi_{\theta_t}(s,b)\cdot\exp(\Delta q_{s,ba}\Delta\theta_t)) \neq 0$ as $\sum_{b\in\mathcal{A}}\pi_{\theta_t}(s,b) = 1$. Thus, the curvature is nonzero, larger or smaller depends on the direction $\Delta\theta_t$ makes with the feature difference $q^\top_{s,b} - q^\top_{s,a}$; if the angle is zero, it is better.

Lemma 5 *For the Boltzmann policy, we have $\nabla KL(\pi_{\theta_t}(\cdot,\cdot)||\pi_{\theta_t+\Delta\theta_t}(\cdot,\cdot)) = -\mathbb{E}[\nabla\log\pi_{\theta_t+\Delta\theta_t}(s,a)]$.*

So, $\psi_{\theta_{t+1}} = \nabla\log\pi_{\theta_t+\Delta\theta_t}$ is an unbiased estimate of $\nabla KL(\pi_{\theta_t}(\cdot,\cdot)||\pi_{\theta_t+\Delta\theta_t}(\cdot,\cdot))$. The proof uses the fact that the action set is finite and hence expectation and gradients can be interchanged. Moreover, for Boltzmann policies, the compatible features are same as the

features associated to policy, except normalized to be mean zero for each state. Proof follows from the definition of KL divergence and the Boltzmann policy. For details, we refer to Appendix A.5 of [49].

Recall from Eq. (5), we have $\nabla KL(\pi_{\theta_t}(\cdot,\cdot)||\pi_{\theta_t+\Delta\theta_t}(\cdot,\cdot)) \approx G(\theta_t)\Delta\theta_t$. Also, in Lemma 1, we obtain that $G(\theta_t)\Delta\theta_t = -\frac{1}{\rho_t}\nabla_\theta J(\theta_t)$. Moreover, we obtain $\nabla KL(\pi_{\theta_t}(\cdot,\cdot)||\pi_{\theta_t+\Delta\theta_t}(\cdot,\cdot)) \approx -\frac{1}{\rho_t}\nabla J(\theta_t)$. Thus, from Lemma 5, and above equations, we have $\mathbb{E}[\nabla \log \pi_{\theta_t+\Delta\theta_t}(s,a)] \approx \frac{1}{\rho_t}\nabla J(\theta_t)$. So, $\nabla \log \pi_{\theta_t+\Delta\theta_t} = \psi_{\theta_{t+1}}$ is approximately an unbiased estimate of $\nabla J(\theta_t)$ upto scaling of $\frac{1}{\rho_t}$ for the Boltzmann policies. It is a valuable observation because to move along the gradient of objective function $J(\cdot)$, we can adjust the updates (of actor parameter) just by moving in the π_{θ_t} prediction space via the compatible features.

We now prove the convergence of FI-MAN, AP-MAN, and FIAP-MAN algorithms. The proofs majorly use the idea of two-time scale stochastic approximations from [11].

4 Convergence Analysis

We now provide the convergence proof of all the three MAN algorithms. To this end, we need following assumptions on the features $\varphi(s)$, and $f(s,a)$ for the value and rewards function, respectively, for any $s \in \mathcal{S}, a \in \mathcal{A}$. This assumption is similar to [53], and also used in single-agent natural actor-critic methods [7].

X 3 *The feature vectors $\varphi(s)$, and $f(s,a)$ are uniformly bounded for any $s \in \mathcal{S}, a \in \mathcal{A}$. Moreover, let the feature matrix $\Phi \in \mathbb{R}^{|\mathcal{S}|\times L}$ have $[\varphi_l(s), s \in \mathcal{S}]^\top$ as its l-th column for any $l \in [L]$, and feature matrix $F \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|\times M}$ have $[f_m(s,a), s \in \mathcal{S}, a \in \mathcal{A}]^\top$ as its m-th column for any $m \in [M]$, then Φ and F have full column rank, and for any $\omega \in \mathbb{R}^L$, we have $\Phi\omega \neq \mathbb{1}$.*

Apart from assumption X. 3, let $D_\theta^s = [d_\theta(s), s \in \mathcal{S}]$, and $\bar{R}_\theta = [\bar{R}_\theta(s), s \in \mathcal{S}]^\top \in \mathbb{R}^{|\mathcal{S}|}$ with $\bar{R}_\theta(s) = \sum_a \pi_\theta(s,a)\cdot\bar{R}(s,a)$. Define the operator $T_\theta^V : \mathbb{R}^{|\mathcal{S}|} \to \mathbb{R}^{|\mathcal{S}|}$ for any state value vector $X \in \mathbb{R}^{|\mathcal{S}|}$ as $T_\theta^V(X) = \bar{R}_\theta - J(\theta)\mathbb{1} + P^\theta X$. The proof of all the three MAN algorithms are done in two steps: (a) convergence of the objective function estimate, critic update, and rewad parameters keeping the actor parameters θ^i fixed for all agents $i \in N$, and (b) convergence of the actor parameters to an asymptotically stable equilibrium set of the ODE corresponding to the actor update. So, we require the following assumption on G_t^i and its inverse $G_t^{i^{-1}}$. This assumption is used for single-agent natural actor-critic algorithms in [7]; here, we have modified it for multi-agent setup.

X 4 *The recursion of Fisher information matrix G_t^i and its inverse $G_t^{i^{-1}}$ satisfy $sup_{t,\theta^i,s,a^i}||G_t^i|| < \infty$; $sup_{t,\theta^i,s,a^i}||G_t^{i^{-1}}|| < \infty$ for each agent $i \in N$.*

Assumption X. 4 ensures that the FI-MAN and FIAP-MAN actor-critic algorithms does not get stuck in a non-stationary point. To ensure the existence of local optima of $J(\theta)$, we make the following assumptions on policy parameters θ_t^i, for each agent $i \in N$.

X 5 *The policy parameters $\{\theta_t^i\}_{t\geq 0}$ of the actor update include a projection operator $\Gamma^i : \mathbb{R}^{m_i} \to \Theta^i \subset \mathbb{R}^{m_i}$ that projects θ_t^i onto a compact set Θ^i. Moreover, $\Theta = \prod_{i=1}^n \Theta^i$ is large enough to include a local optima of $J(\theta)$.*

For each agent $i \in N$, let $\hat{\Gamma}^i$ be the transformed projection operator defined for any $\theta \in \Theta$ with $h : \Theta \to \mathbb{R}^{\sum_{i\in N} m_i}$ being a continuous function as $\hat{\Gamma}^i(h(\theta)) =$

$lim_{0<\eta\to 0}\frac{\Gamma^i(\theta^i+\eta h(\theta))-\theta^i}{\eta}$. If the above limit is not unique, $\hat{\Gamma}^i(h(\theta))$ denotes the set of all possible limit points. This projection operator is useful in convergence proof of the policy parameters. It is an often-used technique to ensure boundedness of iterates in stochastic approximation algorithms. However, we do not require a projection operator in computations because the iterates remain bounded.

We begin by proving the convergence of the critic updates given in Eqs. (2), (3), and (4), respectively. The following theorem will be common in the proof of all the three MAN algorithms. For proof see Appendix A. 6 of [49].

Theorem 4 [53] *Under assumptions X. 1, X. 2, and X. 3, for any policy π_θ, with sequences $\{\lambda_t^i\}, \{\mu_t^i\}, \{v_t^i\}$, we have $lim_t\ \mu_t^i = J(\theta)$, $lim_t\ \lambda_t^i = \lambda_\theta$, and $lim_t\ v_t^i = v_\theta$ a.s. for each agent $i \in N$, where $J(\theta)$, λ_θ, and v_θ are unique solutions to $F^\top D_\theta^{s,a}(\bar{R} - F\lambda_\theta) = 0$; $\Phi^\top D_\theta^s[T_\theta^V(\Phi v_\theta) - \Phi v_\theta] = 0$.*

4.1 Convergence of FI-MAN Actor-Critic Algorithm

To prove the convergence of FI-MAN algorithm, we first show the convergence of recursion for the Fisher information matrix inverse as in Eq. (6).

Theorem 5 *For each agent $i \in N$, and given parameter θ^i, we have $G_t^{i^{-1}} \to G(\theta^i)^{-1}$ as $t \to \infty$ with probability one.*

Please refer to Theorem 5 of [49] for detailed proof. Next, we prove the convergence of actor update. To this end, we can view $-r_{t+1}^i$ as the cost incurred at time t. Hence, transform the actor recursion in the FI-MAN algorithm as $\theta_{t+1}^i \leftarrow \theta_t^i - \beta_{\theta,t} \cdot G_t^{i^{-1}} \cdot \tilde{\delta}_t^i \cdot \psi_t^i$. The convergence of the FI-MAN actor-critic algorithm with linear function approximation is given in the following theorem.

Theorem 6 *Under the assumptions X. 1 - X. 5, the sequence $\{\theta_t^i\}_{t\geq 0}$ obtained from the actor step of the FI-MAN algorithm converges almost surely to asymptotically stable equilibrium set of the ODE*

$$\dot{\theta}^i = \hat{\Gamma}^i[-G(\theta^i)^{-1}\mathbb{E}_{s_t\sim d_\theta, a_t\sim\pi_\theta}(\tilde{\delta}_{t,\theta}^i\psi_{t,\theta}^i)], \quad \forall\, i \in N. \tag{10}$$

Proof Let $\mathcal{F}_{t,1} = \sigma(\theta_\tau, \tau \leq t)$ be the σ-field generated by $\{\theta_\tau\}_{\tau\leq t}$. Moreover, let $\xi_{t+1,1}^i = -G(\theta_t^i)^{-1}\left\{\tilde{\delta}_t^i\psi_t^i - \mathbb{E}_{s_t\sim d_{\theta_t}, a_t\sim\pi_{\theta_t}}(\tilde{\delta}_t^i\psi_t^i|\mathcal{F}_{t,1})\right\}$, and $\xi_{t+1,2}^i = -G(\theta_t^i)^{-1}\mathbb{E}_{s_t\sim d_{\theta_t}, a_t\sim\pi_{\theta_t}}((\tilde{\delta}_t^i - \tilde{\delta}_{t,\theta_t}^i)\psi_t^i|\mathcal{F}_{t,1})$, where $\tilde{\delta}_{t,\theta_t}^i = f_t^\top\lambda_{\theta_t} - J(\theta_t) + \varphi_{t+1}^\top v_{\theta_t} - \varphi_t^\top v_{\theta_t}$. The actor update in the FI-MAN algorithm with local projection then become $\theta_{t+1}^i = \Gamma^i[\theta_t^i - \beta_{\theta,t}G(\theta_t^i)^{-1}\mathbb{E}_{s_t\sim d_{\theta_t}, a_t\sim\pi_{\theta_t}}(\tilde{\delta}_t^i\psi_t^i|\mathcal{F}_{t,1}) + \beta_{\theta,t}\xi_{t+1,1}^i + \beta_{\theta,t}\xi_{t+1,2}^i]$. For a.s. convergence to the asymptotically stable equilibria set of the ODE Eq. (10) for each $i \in N$, we appeal to Kushner–Clark lemma (see appendix C. 3 of [49] and references therein), and we verify its three main conditions below.

First, note that $\xi_{t+1,2} = o(1)$ since critic converges at the faster time scale, i.e., $\tilde{\delta}_t^i \to \tilde{\delta}_{t,\theta_t}^i$ a.s. Next, let $M_t^{1,i} = \sum_{\tau=0}^t \beta_{\theta,\tau}\xi_{\tau+1,1}^i$; then $\{M_t^{1,i}\}$ is a martingale sequence. The sequences $\{z_t^i\}, \{\psi_t^i\}, \{G_t^{i^{-1}}\}$, and $\{\varphi_t^i\}$ are all bounded (by assumptions), and so is the sequence $\{\xi_{t,1}^i\}$ (Here $z_t^i = [\mu_t^i, (\lambda_t^i)^\top, (v_t^i)^\top]^\top$ is the same vector used in the proof of Theorem 4). Hence, $\sum_t \mathbb{E}[||M_{t+1}^{1,i} - M_t^{1,i}||^2 \mid \mathcal{F}_{t,1}] < \infty$ a.s., so the martingale sequence $\{M_t^{1,i}\}$ converges a.s. [36]. So, for any $\epsilon > 0$, we have $\lim_{t\to\infty} \mathbb{P}[sup_{p\geq t}||\sum_{\tau=t}^p \beta_{\theta,\tau}\xi_{\tau,1}^i|| \geq \epsilon] = 0$, as needed.

Regarding continuity of $g^{1,i}(\theta_t) = -G(\theta_t^i)^{-1}\mathbb{E}_{s_t\sim d_{\theta_t},a_t\sim\pi_{\theta_t}}(\tilde{\delta}_t^i\psi_t^i|\mathcal{F}_{t,1})$, we note that $g^{1,i}(\theta_t) = -G(\theta_t^i)^{-1}\sum_{s_t\in\mathcal{S},a_t\in\mathcal{A}} d_{\theta_t}(s_t)\cdot\pi_{\theta_t}(s_t,a_t)\cdot\tilde{\delta}_{t,\theta_t}^i\cdot\psi_{t,\theta_t}^i$. Firstly, ψ_{t,θ_t}^i is continuous by assumption X. 1. The term $d_{\theta_t}(s_t)\cdot\pi_{\theta_t}(s_t,a_t)$ is continuous in θ_t^i since it is the stationary distribution and solution to $d_{\theta_t}(s)\cdot\pi_{\theta_t}(s,a) = \sum_{s'\in\mathcal{S},a'\in\mathcal{A}} P^{\theta_t}(s',a'|s,a)\cdot d_{\theta_t}(s')\cdot\pi_{\theta_t}(s',a')$ and $\sum_{s\in\mathcal{S},a\in\mathcal{A}} d_{\theta_t}(s)\cdot\pi_{\theta_t}(s,a) = 1$, where $P^{\theta_t}(s',a'|s,a) = P(s'|s,a)\cdot\pi_{\theta_t}(s',a')$. From assumption X. 1, $\pi_{\theta_t}(s,a) > 0$ and hence the above set of linear equations has a unique continuous solution in θ_t by assumption X. 1. Moreover, $\tilde{\delta}_{t,\theta_t}^i$ is continuous in θ_t^i since v_{θ_t} as the unique solution to the linear equation $\Phi^\top D_\theta^s[T_\theta^V(\Phi v_\theta) - \Phi v_\theta] = 0$ is continuous in θ_t. Thus, $g^{1,i}(\theta_t)$ is continuous in θ_t^i, as needed in Kushner–Clark lemma. □

4.2 Convergence of AP-MAN Actor-Critic Algorithm

The convergence of critic step, the reward parameters and the objective function estimate are the same as in Theorem 4. So, we show the convergence of advantage parameters and actor updates as given in the AP-MAN algorithm. Similar to the FI-MAN algorithm we again consider the transformed problem; rewards replaced with costs. Thus, the transformed recursion is $w_{t+1}^i \leftarrow (I - \beta_{v,t}\psi_t^i\psi_t^{i^\top})w_t^i - \beta_{v,t}\tilde{\delta}_t^i\psi_t^i$. Section 4.2 of [49] has proof details.

Theorem 7 *Under the assumptions X. 3 and X. 4, for each agent $i \in N$, with actor parameters θ^i, we have $w_t^i \to -G(\theta^i)^{-1}\mathbb{E}[\tilde{\delta}_{t,\theta}^i\psi_t^i]$ as $t \to \infty$ with probability one.*

We now consider the convergence of actor update of the AP-MAN algorithm.

Theorem 8 *Under the assumptions X. 1 - X. 5, the sequence $\{\theta_t^i\}$ obtained from the actor step of AP-MAN algorithm converges a.s. to asymptotically stable equilibrium set of $\dot{\theta}^i = \hat{\Gamma}^i[-G(\theta^i)^{-1}\mathbb{E}_{s_t\sim d_\theta,a_t\sim\pi_\theta}(\tilde{\delta}_{t,\theta}^i\psi_{t,\theta}^i)]$, $\forall\, i \in N$.*

4.3 Convergence of FIAP-MAN Actor-Critic Algorithm

The critic convergence, the convergence of reward parameters, and objective function estimate are the same as in Theorem 4. Like FI-MAN and AP-MAN algorithms, we again consider the transformed problem; rewards are replaced with costs. Therefore, we consider the following recursion: $w_{t+1}^i = (1 - \beta_{v,t})w_t^i - \beta_{v,t}G_t^{i^{-1}}\tilde{\delta}_t^i\psi_t^i$. Again, we refer to Sect. 4.3 of [49] for detailed proofs.

Theorem 9 *Under the assumptions X. 3 and X. 4, for each agent $i \in N$, with actor parameters θ^i, we have $w_t^i \to -G(\theta^i)^{-1}\mathbb{E}[\tilde{\delta}_{t,\theta}^i\psi_t^i]$ as $t \to \infty$ with probability one.*

Theorem 10 *Under assumptions X. 1 - X. 5, the sequence $\{\theta_t^i\}$ obtained from the actor step of FIAP-MAN algorithm converges a.s. to asymptotically stable equilibrium set of the ODE $\dot{\theta}^i = \hat{\Gamma}^i[-G(\theta^i)^{-1}\mathbb{E}_{s_t\sim d_\theta,a_t\sim\pi_\theta}(\tilde{\delta}_{t,\theta}^i\psi_{t,\theta}^i)]$, $\forall\, i \in N$.*

Remark 5 Though the ODEs corresponding to actor update in all MAN algorithms seem similar, we emphasize that they come from three different algorithms, each with a different critic update implicitly. Moreover, all the three MAN algorithms have their ways of updating the advantage parameters. Also, the objective function $J(\theta)$ can have multiple stationary points and local optima. Thus, all the three algorithms *can potentially* attain different optima, and this was clearly illustrated in our comprehensive computational experiments in Sect. 5.1. See also the discussion in Sects. 3.4 and 3.5.

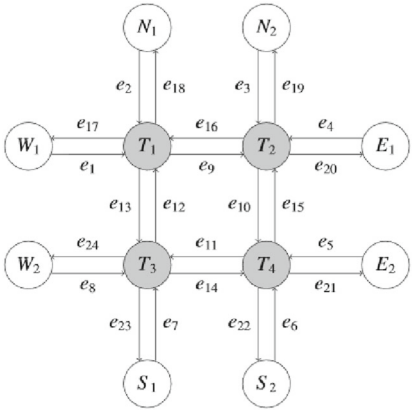


Fig. 1 A bi-lane traffic network with four traffic lights T_1, T_2, T_3, T_4. All the other nodes $N_1, N_2, S_1, S_2, E_1, E_2, W_1$, and W_2 act as the source and destination nodes

To validate the usefulness of our proposed MAN algorithms, we implement them on a bi-lane traffic network and an abstract multi-agent RL model. The detailed computational experiments follow in the next section.

5 Performance of Algorithms in Traffic Network and Abstract MARL Models

This section provides comparative and comprehensive experiments in two different setups. Firstly, we model traffic network control as a multi-agent reinforcement learning problem. A similar model is available in [9] in a related but different context. Another setup is an abstract multi-agent RL model with 15 agents, 15 states, and 2 actions in each state. The model, parameters, rewards, and the linear function approximation features are as given in [18, 53].

All the computations are done in python 3.8 on a machine equipped with 8 GB RAM and an Intel i5 processor. For the traffic network control, we use TraCI, i.e., "Traffic Control Interface." TraCI uses a TCP-based client/server architecture to provide access to sumo-gui, thereby sumo act as a server [28].

5.1 Performance of Algorithms for Traffic Network Controls

Consider the bi-lane traffic network as shown in Fig. 1. The network consists of $N_1, N_2, S_1, S_2, E_1, E_2, W_1$, and W_2 that act as the source and the destination nodes. T_1, T_2, T_3, and T_4 represents the traffic lights and act as agents. All the edges in the network are assumed to be of equal length. The agent's objective is to find a traffic signaling plan to minimize the overall network congestion. Note that the congestion to each traffic light is private information and hence not available to other traffic lights.

The sumo-gui requires the user to provide T, the total number of time steps the simulation needs to be performed, and N_v, the number of vehicles used in each simulation. As per the architecture of sumo-gui, vehicles arrive uniformly from the interval $\{1, 2, \ldots, T\}$. Once a vehicle arrives, it has to be assigned a source and a destination node. We assign the source node

Table 1 Probability $p_{s,ap}$ for source node s and arrival pattern ap. $ap = 1$ assigns high probability to N_2, S_2 and E_1, W_1, whereas $ap = 2$ assigns high probability to all north and south nodes N_1, S_1, N_2, S_2 in Fig. 1

Source Node (s)	W_1	W_2	N_1	N_2	E_1	E_2	S_1	S_2
ap 1 probability	3/16	1/16	1/16	3/16	3/16	1/16	1/16	3/16
ap 2 probability	1/28	1/28	3/14	3/14	1/28	1/28	3/14	3/14

to each incoming vehicle according to various distributions. Different arrival patterns (ap) can be incorporated by considering different source-destination node assignment distributions. We first describe the assignment of the source node. Two different arrival patterns to capture high or low number of vehicles assigned to the source nodes in the network are taken. Let $p_{s,ap}$ be the probability that a vehicle is assigned a source node s if arrival pattern is ap. Table 1 gives probabilities, $p_{s,ap}$ for two arrival patterns ($ap \in \{1, 2\}$) that we consider. The destination node is sampled uniformly from the nodes except the source node. We assume that vehicles follow the shortest path from the source node to the destination node. However, if there are multiple paths with the same path length, then any one of them can be chosen with uniform probability.

For $ap = 1$, we have higher $p_{s,ap}$ for north–south nodes N_2, S_2, and east–west nodes E_1, W_1. Thus, we expect to see heavy congestion for traffic light T_2; almost same congestion for traffic lights T_1 and T_4; and the least congestion for traffic light T_3. For $ap = 2$, more vehicles are assigned to all the north–south nodes. So we expect that all the traffic lights will be equally congested. We now provide the distribution of the number of vehicles assigned to a source node s at time t for a given arrival pattern ap.

Let N_t be the number of vehicles arrived at time t, and N_t^s be the number of vehicles assigned to source node s at time t. Thus, $N_t = \sum_s N_t^s$. Note that the arrivals are uniform in $\{0, 1, \ldots, T\}$, so N_t is a binomial random variable with parameters $\left(N_v, \frac{1}{T}\right)$. Therefore, we have $\mathbb{P}(N_t = r) = \binom{N_v}{r}\left(\frac{1}{T}\right)^r\left(1 - \frac{1}{T}\right)^{N_v - r}$, $\forall\, r = 0, 1, \cdots, N_v$. Moreover, using the law of total probability, for all $ap \in \{1, 2\}$, we obtain

$$\mathbb{P}(N_t^s = k \mid ap) = \binom{N_v}{k}\left(\frac{p_{s,ap}}{T}\right)^k\left(1 - \frac{p_{s,ap}}{T}\right)^{N_v - k},\ \forall\, k = 0, 1, \cdots, N_v, \qquad (11)$$

i.e., the distribution of N_t^s for a given arrival pattern ap is also binomial with parameters $\left(N_v, \frac{p_{s,ap}}{T}\right)$. More details on above probability are available in Appendix B.2.3 of [49].

In our experiments, we take $T = 180000$ seconds which is divided into simulation cycle (called decision epoch) of $T_c = 120$ seconds each. Thus, there are 1500 decision epochs. The number of vehicles are taken as $N_v = 50000$.

5.1.1 Decentralized Framework for Traffic Network Control

In this section, we model the above traffic network control as a fully decentralized MARL problem with traffic lights as agents, $N = \{T_1, T_2, T_3, T_4\}$. Let $E_{in} = \{e_1, e_2, e_{12}, e_{16}, e_3, e_4, e_9, e_{15}, e_8, e_7, e_{11}, e_{13}, e_5, e_6, e_{10}, e_{14}\}$ be the set of edges directed toward the traffic lights. Let the maximum capacity of each lane in the network be $C = 50$. The state-space of the system consists of the number of vehicles in the lanes belonging to E_{in}. Hence, the size of the state space is 50^{16}. At every decision epoch, each traffic light follows one of the following traffic signal plans for the next $T_c = 120$ simulation steps.

1. Equal green time of $\frac{T_c}{2}$ for both north–south and east–west lanes

2. $\frac{3T_c}{4}$ green time for north–south and $\frac{T_c}{4}$ green time for east–west lanes
3. $\frac{T_c}{4}$ green time for north–south and $\frac{3T_c}{4}$ green time for east–west lanes.

Thus, the total number of actions available at each traffic light is $3^4 = 81$. The rewards given to each agent is equal to the negative of the average number of vehicles stopped at its corresponding traffic light. Note that the rewards are privately available to each traffic light only. We aim to maximize the expected time average of the globally averaged rewards, which is equivalent to minimize the (time average of) number of stopped vehicles in the system. Since the state space is huge (50^{16}), we use the linear function approximation for the state value function and the reward function. The approximate state value for state s is $V(s; v) = v^\top \varphi(s)$, where $\varphi(s) \in \mathbb{R}^L$, $L << |\mathcal{S}|$, is the feature vector for the state s. Moreover, the reward function is approximated as $R(s, a; \lambda) = \lambda^\top f(s, a)$ where $f(s, a) \in \mathbb{R}^M$, $M << |\mathcal{S}||\mathcal{A}|$ are the features associated with each state-action pair (s, a). Next, we describe these features [9].

Let x_t^i denote the number of vehicles in lane $e_i \in E_{in}$ at time t. We normalize x_t^i via maximum capacity of a lane C to obtain $z_t^i = x_t^i/C$. We define $\xi(s) = \left(z_t^1, z_t^2, \cdots, z_t^{16}, z_t^1 z_t^2, \cdots, z_t^6 z_t^5, z_t^1 z_t^2 z_t^{12}, \cdots, z_t^5 z_t^6 z_t^{10}\right)$ as a vector having components containing z_t^i, as well as components with products of two or three z_t^i's. The product terms are of the form $z_t^i z_t^j$ and $z_t^i z_t^j z_t^l$, where all terms in the product correspond to the same traffic light. The feature vector $\varphi(s)$ is defined as having all the components of $\xi(s)$ along with an additional bias component, 1. Thus, $\varphi(s) = \left(1, \xi(s)\right)^\top$. The length of the feature vector $\varphi(s)$ is $L = 1 + (16 + 4 \times (4^2 + 4^3)) = 337$.

For each agent $i \in N$, we parameterize the local policy $\pi^i(s, a^i)$ using the Boltzmann distribution as $\pi^i_{\theta^i}(s, a^i) = \frac{\exp(q_{s,a^i}^\top \cdot \theta^i)}{\sum_{b^i \in \mathcal{A}^i} \exp(q_{s,b^i}^\top \cdot \theta^i)}$, where $q_{s,b^i} \in \mathbb{R}^{m_i}$ is the feature vector of dimension same as θ^i, for any $s \in \mathcal{S}$ and $b^i \in \mathcal{A}^i$, for all $i \in N$. The feature vector is $q_{s,a^i} = \left(1, a^{i,1}\xi(s), a^{i,2}\xi(s), a^{i,3}\xi(s)\right)^\top$, $\forall\, i \in N$, where $\xi(s)$ is defined as earlier, and $a^{i,j}$ is 1 if signal plan j is selected in action a^i by agent $i \in N$, and 0 otherwise. The length of q_{s,a^i}, i.e., $m_i = 3 \times 336 + 1 = 1009$. For the Boltzmann policy function $\pi^i_{\theta^i}(s, a^i)$, we have $\nabla_{\theta^i} \log \pi^i_{\theta^i}(s, a^i) = q_{s,a^i} - \sum_{b^i \in \mathcal{A}^i} \pi^i_{\theta^i}(s, b^i) q_{s,b^i}$, where $\nabla_{\theta_i} \log \pi^i_{\theta_i}(s, a^i)$ are the compatible features as in the policy gradient Theorem [46]. Note that the compatible features are same as the features associated to policy, except normalized to be mean zero for each state. The features $f(s, a)$ for the rewards function are similar to q_{s,a^i} for each $i \in N$, thus $M = 4 \times 3 \times 336 + 1 = 4033$.

We implement all the three MAN algorithms and compared the average network congestion with the MAAC algorithm. For all $i \in N$, the initial value of parameters μ_0^i, $\tilde{\mu}_0^i$, v_0^i, $\tilde{v}_0^i$, λ_0^i, $\tilde{\lambda}_0^i$, θ_0^i, w_0^i are taken as zero vectors of appropriate dimensions. The Fisher information matrix inverse, $G_0^{i^{-1}}$, is initialized to I, $\forall\, i \in N$. The critic and actor step-sizes are taken as $\beta_{v,t} = \frac{1}{(t+1)^{0.65}}$, and $\beta_{\theta,t} = \frac{1}{(t+1)^{0.85}}$, respectively. These step-sizes satisfy the Robbins–Monro conditions. We assume that the communication graph $\mathcal{G}_t$ is a complete graph at all time instances t and $c_t(i, j) = \frac{1}{4}$ for all pairs i, j of agents. Although we do not use the eligibility traces in the convergence analysis, we use them ($\lambda = 0.25$ for TD(λ) [45]) to provide better performance in case of function approximations. We believe that the convergence of MAN algorithms while incorporating eligibility traces is easy to follow, so we avoid them here.

Table 2 Average network congestion, standard deviation and 95% confidence interval for arrival pattern 1. FIAP-MAN has ≈ 18%, and FI-MAN has ≈ 14% less congestion than MAAC algorithm. The congestion for the AP-MAN algorithm is almost the same as the MAAC algorithm with high confidence

Algorithms	Avg network congestion	Standard Deviation	Confidence Interval (95%)
MAAC	14.01687	0.08405	(13.96478, 14.06896)
FI-MAN	12.02819	1.48071	(11.11045, 12.94593)
AP-MAN	14.07899	0.08266	(14.02776, 14.13022)
FIAP-MAN	11.28657	1.04137	(10.64113, 11.93201)

5.1.2 Performance of Traffic Network for Arrival Pattern 1

Recall, for arrival pattern 1 we assign high probability $p_{s,ap}$ to the source nodes N_2, S_2 and E_1, E_2 and low probability to other source nodes. Table 2 provides the average network congestion (averaged over 10 runs, and round off to 5 decimal places), standard deviation and 95% confidence interval.

We observe an ≈ 18% reduction in average congestion for FIAP-MAN and ≈ 14% reduction for FI-MAN algorithms compared to the MAAC algorithm. These observations are theoretically justified in Sect. 3.4.

To show that these algorithms have attained the steady state, we provide average congestion, and the correction factor (CF), i.e., the 95% confidence value which is defined as $CF = 1.96 \times \frac{std\ dvn}{\sqrt{10}}$ for last 200 decision epochs in Table 8 of Appendix B.2.1 of [49]. The average network congestion for the MAN algorithms are almost (up to 1st decimal) on decreasing trend w.r.t. network congestion; however, this decay is prolonged (0.1 fall in congestion in 200 epochs), suggesting the convergence of these algorithms to local minimum. Thus, we see that algorithms involving the natural gradients dominate those involving standard gradients. Figure 2 shows the (time) average network congestion for single run (thus lower the better).

For almost 180 decision epochs, all the algorithms have the same (time) average network congestion. However, after 180 decision epochs, FI-MAN and FIAP-MAN follow different paths and hence find different local minima as shown in Theorem 3. We want to emphasize that the Theorem 3 is for maximization framework. As given in Sect. 5.1.1, we are also maximizing the globally average rewards, which is equivalent to minimizing the (time average of) number of stopped vehicles.

Actor Parameter Comparison for Arrival Pattern 1

Recall, in Theorem 3, we show that under some conditions $J(\tilde{\theta}^N_{t+1}) \geq J(\tilde{\theta}^M_{t+1})$, for all $t \geq t_0$, and hence at each iterate the average network congestion in FI-MAN, and FIAP-MAN algorithms are better than MAAC algorithm. To investigate this further, we plot the norm of difference of the actor parameter of all the three MAN algorithms with MAAC algorithm for each agent. For traffic light T_1 (or agent 1), these differences are shown in Fig. 3 (for other agents see Fig. 8 in Appendix B.2.1 of [49]).

We observe that all the three MAN algorithms pick up θ_2 (i.e., the actor parameter at decision epoch 2) that is different from that of the MAAC scheme at varying degrees, with FI-MAN being a bit more "farther." However, a significant difference is observed around

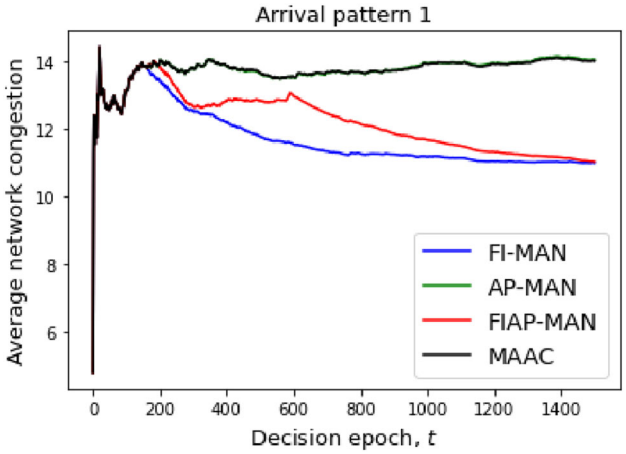
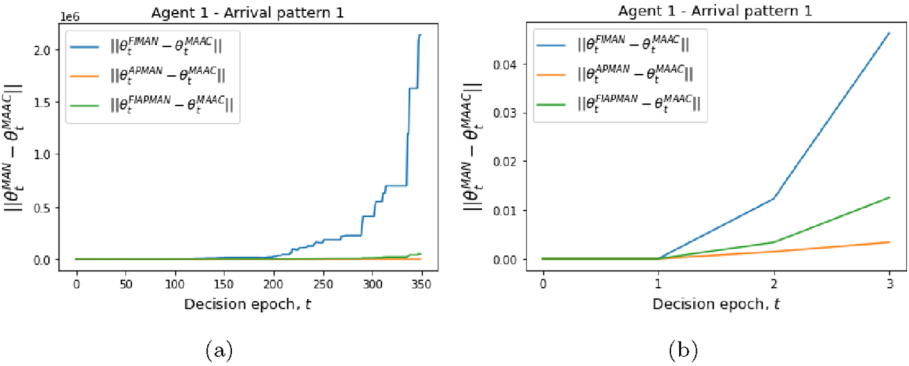


Fig. 2 (Time) average network congestion of all the algorithms with ap 1. The congestion is least for FIAP-MAN and FI-MAN algorithms. However, MAAC and AP-MAN algorithms have almost the same congestion. For a few initial decision epochs ≈ 180, all the algorithms have almost the same performance, but afterward, they find different directions and ends in different optima



(a) (b)

Fig. 3 Norm of difference in the actor parameter of agent 1 for all the 3 MAN algorithms with MAAC algorithm for arrival pattern 1. Figure (a) is shown for 350 decision epochs; to show that the differences in the actor parameter are from decision epoch 2 itself, we zoom it in figure (b) in the left panel. However, the significant differences are observed only after ≈ 180 epochs. This illustrates Theorem 3 and related discussions in Sect. 3.4

decision epoch $\approx$ 180. For better understanding, the same graphs are also shown in the logarithmic scale for agent 1 and agent 2 with arrival pattern 1 in Fig. 4.

We see that the norm difference is linearly increasing in FI-MAN and FIAP-MAN algorithms, whereas it is almost flat for the AP-MAN algorithm. So, the iterates of these 2 algorithms are exponentially separating from those of the MAAC algorithm. This again substantiates our analysis in Sect. 3.4.

Though we aim to minimize the network congestion, in Table 3, we also provide the average congestion and the correction factor (CF) to each traffic light for last decision epoch (Table 8 in Appendix B.2.1 shows these values for last 200 decision epochs). Expectedly, the average congestion for traffic light T_2 is highest; it is almost same for traffic lights T_1, T_4; and least for T_3.

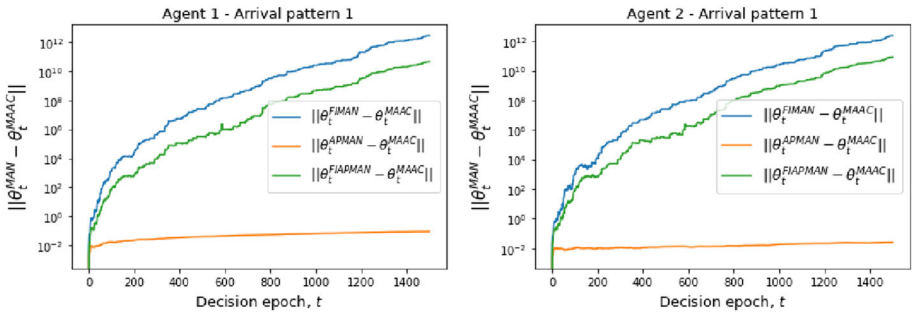


Fig. 4 Norm of differences of the actor parameters for agents 1 and 2 with traffic arrival pattern 1 in logarithmic scale illustrating Theorem 3 and related discussions in Sect. 3.4

Table 3 Average congestion and correction factor (CF) for each traffic light for arrival pattern 1 (described in Sect. 5.1.2). CF is defined as $1.96 \times \frac{std\ dvn}{\sqrt{10}}$

Algorithms	Congestion (Avg $\pm$ CF)			
	T_1	T_2	T_3	T_4
MAAC	3.733 ± 0.067	4.336 ± 0.050	2.249 ± 0.013	3.699 ± 0.040
FI-MAN	2.613 ± 0.110	4.492 ± 0.868	2.359 ± 0.161	2.564 ± 0.038
AP-MAN	3.748 ± 0.073	4.338 ± 0.046	2.247 ± 0.013	3.746 ± 0.039
FIAP-MAN	2.711 ± 0.214	3.785 ± 0.371	1.907 ± 0.148	2.883 ± 0.147

Table 4 Average network congestion, standard deviation and 95% confidence interval at last decision epoch for arrival pattern 2 (described in Sect. 5.1.3). FI-MAN and FIAP-MAN algorithms has $\approx 25\%$ less average network congestion than MAAC algorithm. The performance of AP-MAN algorithm is almost similar to MAAC algorithm as shown in Theorem 3 and in Sect. 3.4

Algorithms	Avg network congestion	Standard deviation	Confidence Interval (95%)
MAAC	13.64571	0.19755	(13.52327, 13.76815)
FI-MAN	10.16988	0.11877	(10.09627, 10.24349)
AP-MAN	13.77573	0.18925	(13.65843, 13.89303)
FIAP-MAN	10.19858	0.21248	(10.06689, 10.33027)

5.1.3 Performance of Traffic Network for Arrival Pattern 2

In arrival pattern 2, the traffic origins N_1, N_2, S_1 and S_2 have higher probabilities of being assigned a vehicle. We take $p_{s,ap}$ for these nodes as $\frac{3}{14}$, and for all other nodes, it is $\frac{1}{28}$. So, we expect almost the same average congestion to all the traffic lights. This observation is reported in Appendix B.2.2 of [49]. Table 4 provides the average network congestion (averaged over 10 runs, and round off to 5 decimal places), standard deviation and 95% confidence interval for arrival pattern 2.

We observe an $\approx 25\%$ reduction in the average congestion with FI-MAN and FIAP-MAN algorithms as compared to the MAAC algorithm. AP-MAN is on par with the MAAC algorithm. This again shows the usefulness of the natural gradient-based algorithms. As opposed to ap 1 where FIAP-MAN algorithm has slightly better performance than FI-MAN algo-

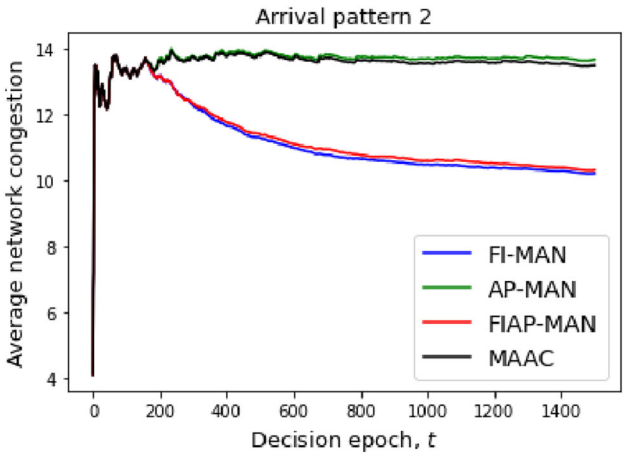


Fig. 5 (Time) average network congestion for arrival pattern 2. For few initial decision epochs ≈ 180 all the algorithms have almost same performance, but afterward they find different directions, and hence ends in different optima

rithm, in ap 2, both algorithms have almost similar average network congestion. Moreover, the standard deviation in ap 1 is much higher than in ap 2. Figure 5 shows the (time) average congestion for single simulation run of all the algorithms.

Actor Parameter Comparison for Arrival Pattern 2

Similar to arrival pattern 1, in Fig. 6, we plot the norm of the difference for traffic light T_1 (for other traffic lights (agents) see Fig. 9 of [49]). For better understanding, the same graphs are also shown for agent 1 and 2 in the logarithmic scale in Fig. 7 of [49]. Again, we see that the norm difference is linearly increasing in FI-MAN and FIAP-MAN algorithms, whereas it is almost flat for the AP-MAN algorithm. So, the iterates of these 2 algorithms are exponentially separating from the MAAC algorithm. This again substantiates our analysis in Sect. 3.4. Moreover, we also compute the average congestion and (simulation) correction factor for each traffic light. Table 5 of [49] shows these values for last decision epoch (See Table 9 for last 200 decision epochs). As expected, the average congestion to each traffic light is almost the same.

We now present another computational experiment where we consider an abstract MARL with $n = 15$ agents. The model, algorithm parameters, including transition probabilities, rewards, and features for state value function, and rewards are the same as given in [18, 53]

5.2 Performance of Algorithms in Abstract MARL Model

The abstract MARL model that we consider consists of $n = 15$ agents and $|\mathcal{S}| = 15$ states. Each agent $i \in N$ is endowed with the binary valued actions $\mathcal{A}^i \in \{0, 1\}$. Therefore, the total number of actions are 2^{15}. Each element of the transition probability is a random number uniformly generated from the interval [0, 1]. These values are normalized to incorporate the stochasticity. To ensure the ergodicity, we add a small constant 10^{-5} to each entry of the transition matrix. The mean reward $R^i(s, a)$ are sampled uniformly from the interval $[0, 4]$ for each agent $i \in N$, and for each state-action pair (s, a). The instantaneous rewards r_t^i are

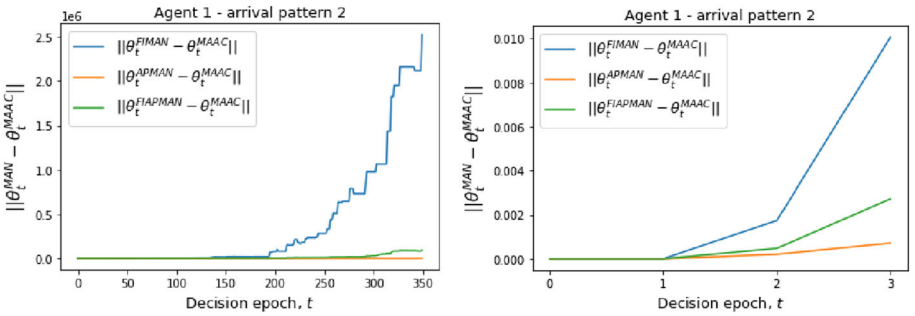


Fig. 6 Norm of difference in the actor parameter of agent 1 for all the 3 MAN algorithms with MAAC algorithm. Figure (a) is shown for 350 decision epochs; to show that the differences in the actor parameter are from decision epoch 2 itself, we zoom it in Figure (b). However, the significant differences are observed only after $\approx$ 180 epochs

Table 5 Globally averaged rewards, standard deviation and 95% confidence for all the algorithms for the abstract multi-agent RL problem. We observe that globally averaged rewards and standard deviation are almost same for all the algorithms with high confidence. All the values are averaged over 25 runs

Algorithm	Avg Rewards	Std Dvn	Confidence interval
MAAC	1.993280	0.066421	(1.967243, 2.019316)
FI-MAN	2.008412	0.055538	(1.986642, 2.030183)
AP-MAN	1.982451	0.079404	(1.951325, 2.013576)
FIAP-MAN	1.981089	0.093754	(1.944338, 2.017839)

sampled uniformly from the interval $[R^i(s,a) - 0.5, R^i(s,a) + 0.5]$. We parameterize the policy using the Boltzmann distribution with $m_i = 5, \ \forall\, i \in N$. All the feature vectors (for the state value and the reward functions) are sampled uniformly from the set [0, 1] of suitable dimensions. The following table compares the globally averaged return from all the three MAN actor-critic algorithms with the MAAC algorithm. The globally averaged rewards are almost close to each other. To provide more details we also compute the relative V values for each agent $i \in N$ that is defined as $V_\theta(s; v^i) = {v^i}^\top \varphi(s)$; this way of comparison of MARL algorithms was earlier used by [53]. Thus, the higher the value, the better is the algorithm. More details of model and computations are available in Appendix B.1 of [49].

6 Related Work

Reinforcement learning has been extensively studied and explored by researchers because of its varied applications and usefulness in many real-world applications [16, 17, 21]. Single-agent reinforcement learning models are well-explained in many works including [5, 6, 45]. The average reward reinforcement learning algorithms are available in [19, 30].

Various algorithms to compute the optimal policy for single-agent RL are available; these are mainly classified as off-policy and on-policy algorithms in the literature [45]. Moreover, because of the large state and action space, it is often helpful to consider the function approximations of the state value functions [46]. To this end, actor-critic algorithms with

function approximations are presented in [27]. In actor-critic algorithms, the actor updates the policy parameters, and the critic evaluates the policy's value for the actor parameters until convergence. The convergence of linear architecture in actor-critic methods is known. The algorithm in [27] uses the standard gradient while estimating the objective function. However, as mentioned in Sect. 2, we outlined some drawbacks of using standard gradients [31, 42].

To the best of our knowledge, the idea of natural gradients stems from the work of [3]. Afterward, it has been expanded to learning setup in [2]. For recent developments and work on natural gradients we refer to [31]. Some recent overviews of natural gradients are available in [13] and lecture slides by Roger Grosse.[2] The policy gradient theorem involving the natural gradients is explored in [26]. For the discounted reward [1, 33] recently showed that despite the non-concavity in the objective function, the policy gradient methods under tabular setting with softmax policy characterization find the global optima. However, such a result is not available for average reward criteria with actor-critic methods and the general class of policy. Moreover, we also see in our computations in Sect. 5.1 that MAN algorithms are stabilizing at different local optima. Actor-critic methods involving the natural gradients for single-agent are available in [7, 40]. On the contrary, we deal with the multi-agent setup where all agents have private rewards but have a common objective. For a comparative survey of the MARL algorithms, we refer to [15, 50, 52].

The MARL algorithms given in [52] are majorly centralized, and hence relatively slow. However, in many situations [16, 17] deploying a centralized agent is inefficient and costly. Recently, [53] gave two different actor-critic algorithms in a fully decentralized setup; one based on approximating the state-action value function and the other approximating the state value function. Another work in the same direction is available in [24, 32, 44, 51]. In particular, for distributed stochastic approximations, authors in [32] introduced and analyzed a non-linear gossip-based distributed stochastic approximation scheme. We use some proof techniques as part of consensus updates from it. We build on algorithm 2 of the [53] and incorporate the natural gradients into it. The algorithms that we propose use the natural gradients as in [7]. We propose three algorithms incorporating natural gradients into multi-agent RL based on Fisher's information matrix inverse, approximation of advantage parameters, or both. Using the ideas of stochastic approximation available in [11, 12, 29], we prove the convergence of all the proposed algorithms.

7 Discussion

This paper proposes three multi-agent natural actor-critic (MAN) reinforcement learning algorithms. Instead of using a central controller for taking action, our algorithms use the consensus matrix and are fully decentralized. These MAN algorithms majorly use the Fisher information matrix and the advantage function approximations. We show the convergence of all the three MAN algorithms, possibly to different local optima.

We prove that a deterministic equivalent of the natural gradient-based algorithm dominates that of the MAAC algorithm under some conditions. It follows by leveraging a fundamental property of the Fisher information matrix that we show: the minimum singular value is within the reciprocal of the dimension of the policy parameterization space.

The Fisher information matrix in the natural gradient-based algorithms captures the KL divergence curvature between the policies at consecutive iterates. Indeed, we show that the KL

[2] https://csc2541-f17.github.io/slides/lec05a.pdf.

divergence is proportional to the objective function's gradient. The use of natural gradients offered a *new representation* to the objective function's gradient in the prediction space of policy distributions which improved the search for better policies.

To demonstrate the usefulness of our algorithm, we empirically evaluate them on a bi-lane traffic network model. The goal is to minimize the overall congestion in the network in a fully decentralized fashion. Sometimes the MAN algorithms can reduce the network congestion by almost $\approx 25\%$ compared to the MAAC algorithm. One of our natural gradient-based algorithms, AP-MAN, is on par with the MAAC algorithm. Moreover, we also consider an abstract MARL with $n = 15$ agents; again, the MAN algorithms are at least as good as the MAAC algorithm with high confidence.

We now mention some of the further possibilities. Firstly, some assumptions on the communication network can be relaxed [48]. A careful study of the trade-off between extra per iterate computation versus the gain in the objective function value of the learned MARL policy obtained by these MAN algorithms would be useful. It is in the context of a similar phenomenon in optimization algorithms [13, 37] and other computational sciences.

Moreover, further understanding of the natural gradients and its key ingredient, the Fisher information matrix $G(\theta)$, is needed in their role as learning representations. Our uniform bound on the smallest singular value of $G(\theta)$ and its role in the dominance of deterministic MAN algorithms are initial results in these directions. More broadly, various learning representations for RL like natural gradients and others are further possibilities.

Acknowledgements We thank both the Reviewers for their comments that helped us to streamline the presentation. We thank the Editors of this Special Issue for their encouragement. While working on this paper, Prashant Trivedi is partially supported by a Teaching Assistantship offered by, GoI, Government of India. We also thank the IRCC IIT Bombay for partially funding the open access publication of this article.

Data Availability We declare that all the input parameters used in the computations for data generation are available within the article.

References

1. Alekh A, Kakade Sham M, Lee Jason D, Gaurav M (2021) On the theory of policy gradient methods: optimality, approximation, and distribution shift. J Mach Learn Res 22(98):1–76
2. Amari SI (1998) Natural gradient works efficiently in learning. Neural Comput 10(2):251–276
3. Amari SI, Douglas SC (1998) Why natural gradient? In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'98 (Cat. No. 98CH36181)*, 2:1213–1216. IEEE
4. Aoki M (2016) Optimization of stochastic systems: topics in discrete-time systems. Elsevier, Amsterdam
5. Bertsekas Dimitri P (1995) Dynamic programming and optimal control, vol 1. Athena scientific, Belmont
6. Bertsekas DP (2019) Reinforcement learning and optimal control. Athena scientific, Belmont
7. Bhatnagar S, Sutton RS, Ghavamzadeh M, Lee M (2009) Natural actor-critic algorithms. Automatica 45(11):2471–2482
8. Bhatnagar S, Sutton RiS, Ghavamzadeh M, Lee M (2009) Natural actor–critic algorithms. University of Alberta Department of Computing Science Technical Report TR 09-10

9. Bhatnagar S (2020) Single and multi-agent reinforcement learning in changing environments. Master's thesis, IE & OR, IIT Bombay
10. Bianchi P, Fort G, Hachem W (2013) Performance of a distributed stochastic approximation algorithm. IEEE Trans Inf Theory 59(11):7405–7418
11. Borkar VS (2009) Stochastic approximation: a dynamical systems viewpoint, vol 48. Springer, Cham
12. Borkar Vivek S, Meyn Sean P (2000) The ode method for convergence of stochastic approximation and reinforcement learning. SIAM J Control Optim 38(2):447–469
13. Bottou L, Curtis FE, Nocedal J (2018) Optimization methods for large-scale machine learning. SIAM Rev 60(2):223–311
14. Boyd S, Ghosh A, Prabhakar B, Shah D (2006) Randomized gossip algorithms. IEEE Trans Inf Theory 52(6):2508–2530
15. Busoniu L, Babuska R, De Schutter B (2008) A comprehensive survey of multiagent reinforcement learning. IEEE Trans Syst Man Cybern Part C (Appl Rev), 38(2):156–172
16. Corke P, Peterson R, Rus D (2005) Networked robots: flying robot navigation using a sensor net. In: Robotics Research. The eleventh international symposium pp. 234–243. Springer
17. Dall'Anese E, Zhu H, Giannakis GB (2013) Distributed optimal power flow for smart microgrids. IEEE Trans Smart Grid 4(3):1464–1475
18. Dann C, Neumann G, Peters J (2014) Policy evaluation with temporal differences: a survey and comparison. JMLR 15:809–883
19. Dewanto V, Dunn G, Eshragh A, Gallagher M, Roosta F (2020) Average-reward model-free reinforcement learning: a systematic review and literature mapping. arxiv:2010.08920
20. Doan TT, Maguluri ST, Romberg J (2021) Finite-time performance of distributed temporal-difference learning with linear function approximation. SIAM J Math Data Sci 3(1):298–320
21. Alexander Fax J, Murray RM (2004) Information flow and cooperative control of vehicle formations. IEEE Trans Autom Control 49(9):1465–1476
22. Folland GB (1999) Real analysis: modern techniques and their applications, vol 40. John Wiley & Sons, New York
23. Ivo G, Lucian B, Lopes Gabriel AD, Robert B (2012) A survey of actor-critic reinforcement learning: standard and natural policy gradients. IEEE Trans Syst Man Cybern Part C (Appl Rev) 42(6):1291–1307
24. Heredia PC, Mou S (2019) Distributed multi-agent reinforcement learning by actor-critic method. IFAC-PapersOnLine 52(20):363–368
25. Horn R, Johnson R (2012) Matrix analysis. Cambridge University Press, Cambridge
26. Kakade SM (2001) A natural policy gradient. Adv Neural Inf Process Syst 14
27. Konda VR, Tsitsiklis JN (2000) Actor-critic algorithms. Adv Neural Inf Process Syst pp. 1008–1014. Citeseer
28. Krajzewicz D, Erdmann J, Behrisch M, Bieker L (2012) Recent development and applications of sumo-simulation of urban mobility. Int J Adv Syst Measure, 5(3&4)
29. Kushner HJ, Clark DS (2012) Stochastic approximation methods for constrained and unconstrained systems, vol 26. Springer Science & Business Media, Cham
30. Mahadevan S (1996) Average reward reinforcement learning: foundations, algorithms, and empirical results. Mach Learn 22(1):159–195
31. Martens J (2020) New insights and perspectives on the natural gradient method. J Mach Learn Res 21(146):1–76
32. Mathkar AS, Borkar VS (2016) Nonlinear gossip. SIAM J Control Optim 54(3):1535–1557
33. Mei J, Xiao C, Szepesvari C, Schuurmans D (2020) On the global convergence rates of softmax policy gradient methods. In: International conference on machine learning, pp. 6820–6829. PMLR
34. Nedic A, Olshevsky A, Shi W (2017) Achieving geometric convergence for distributed optimization over time-varying graphs. SIAM J Optim 27(4):2597–2633
35. Nedic A, Ozdaglar A (2009) Distributed subgradient methods for multi-agent optimization. IEEE Trans Autom Control 54(1):48–61
36. Neveu TP (1975) Jacques translated by Speed. *Discrete-parameter martingales*, vol. 10. North-Holland, Amsterdam
37. Nocedal J, Wright S (2006) Numerical optimization. Springer Science & Business Media, Cham
38. Patchell JW, Jacobs OLR (1971) Separability, neutrality and certainty equivalence. Int J Control 13(2):337–342
39. Peters J, Vijayakumar S, Schaal S (2003) Reinforcement learning for humanoid robotics. In: 3rd International conference on humanoid robots, pp. 1–20
40. Peters J, Schaal S (2008) Natural actor-critic. Neurocomputing 71(7–9):1180–1190
41. Puterman Martin L (2014) Markov decision processes: discrete stochastic dynamic programming. John Wiley & Sons, New York

42. Ratliff N (2013) Information geometry and natural gradients. Available at https://ipvs.informatik.uni-stuttgart.de/mlr/wp-content/uploads/2015/01/mathematics_for_intelligent_systems_lecture12_notes_I.pdf. Last accessed August 20, 2021
43. Sherman J, Morrison WJ (1950) Adjustment of an inverse matrix corresponding to a change in one element of a given matrix. Ann Math Stat 21(1):124–127
44. Suttle W, Yang Z, Zhang K, Wang Z, Başar T, Liu J (2020) A multi-agent off-policy actor-critic algorithm for distributed reinforcement learning. IFAC-PapersOnLine 53(2):1549–1554
45. Sutton RS, Barto AG (2018) Reinforcement learning: an introduction. MIT press, Cambridge
46. Sutton RS, McAllester D, Singh S, Mansour Y (1999) Policy gradient methods for reinforcement learning with function approximation. Adv Neural Inf Process Syst 99:1057–1063
47. Thoppe G, Borkar V (2019) A concentration bound for stochastic approximation via Alekseev's formula. Stoch Syst 9(1):1–26
48. Thoppe GC, Kumar B (2021) A law of iterated logarithm for multi-agent reinforcement learning. Adv Neural Inf Process Syst, 34
49. Trivedi P, Hemachandra N (2021) Multi-agent natural actor-critic reinforcement learning algorithms. arxiv:2109.01654
50. Tuyls K, Weiss G (2012) Multiagent learning: basics, challenges, and prospects. AI Magaz 33(3):41–41
51. Zhang K, Yang Z, Başar T (2021) Decentralized multi-agent reinforcement learning with networked agents: recent advances. Front Inf Technol Electron Eng 22(6):802–814
52. Zhang K, Yang Z, Başar T (2021) Multi-agent reinforcement learning: a selective overview of theories and algorithms. Handbook of reinforcement learning and control, pp. 321–384
53. Zhang K, Yang Z, Liu H, Zhang T, Basar T (2018) Fully decentralized multi-agent reinforcement learning with networked agents. In: International conference on machine learning, pp. 5872–5881. PMLR