Intermittent Communications in Decentralized Shadow Reward Actor-Critic

Amrit Singh Bedi[1], Alec Koppel[1], Mengdi Wang[2], and Junyu Zhang[2]

***Abstract*— Broader decision-making goals such as risk-sensitivity, exploration, and incorporating prior experience motivates the study of cooperative multi-agent reinforcement learning (MARL) problems where the objective is any nonlinear function of the team's long-term state-action occupancy measure, i.e., a *general utility*, which subsumes the aforementioned goals. Existing decentralized actor-critic algorithms to solve this problem require extensive message passing per policy update, which may be impractical. Thus, we put forth Communication-Efficient Decentralized Shadow Reward Actor-Critic (CE-DSAC) that may operate with time-varying or event-triggered network connectivities. This scheme operates by having agents to alternate between policy evaluation (critic), weighted averaging with neighbors (information mixing), and local gradient updates for their policy parameters (actor). CE-DSAC is different from the usual critic update in its local occupancy measure estimation step which is needed to estimate the derivative of the local utility with respect to their occupancy measure, i.e., the "shadow reward," and the amount of local weighted averaging steps executed by agents. This scheme improves existing tradeoffs between communications and convergence: to obtain ϵ-stationarity, we require in $\mathcal{O}(1/\epsilon^{2.5})$ (Theorem IV.6) or faster $\mathcal{O}(1/\epsilon^{2})$ (Corollary IV.8) steps with high probability. Experiments demonstrate the merits of this approach for multiple RL agents solving cooperative navigation tasks with intermittent communications.**

I. Introduction

In reinforcement learning (RL), an autonomous agent starting from one state, repeatedly selects actions which trigger state transitions according to a Markov transition density, whereby instantaneous rewards are revealed by the environment. This setting, associated with a Markov Decision Process (MDP) [1], is different from optimal control [2] in that access to a system model that governs the state evolution is unavailable. Instead, one seeks to directly estimate parameters of a distribution over actions, i.e., policy [3]. The standard goal is to find the policy associated with maximizing the cumulative return in the long-run. This framework has gained traction in recent years in settings where first-principles models are unavailable or intractably complicated, as in vision-based robotic manipulation [4], web services [5], and logistics [6], and games [7].

This work focuses on team settings: a collection of RL agents interact to effect aggregate outcomes. We concentrate on the case where agents cooperate [8], as in autonomous vehicular networks [9], games [7], and various other settings. Cooperation in multi-agent RL (MARL) may be contrasted with competitive or mixed settings [10], and necessitates an incentive for teamwork. Conventionally, this incentive is encoded by defining the reward of the team as the global sum of agents' individual rewards [11]. Instead, we study a more general setting in which agents' utilities are any nonlinear function of the global state-action occupancy measure, i.e., a *general utility*. This specification is motivated by the fact that making modern RL effective in practice often requires reasoning about exploration [12], risk and safety [13], constraints [14], prior experience [15], all of which may be defined as nonlinear functions of the occupancy measure [16], whereas the standard cumulative return (value function) is necessarily linear.

To approach algorithms for MARL with general utilities, a critical observation is that the foundation of most centralized RL techniques, i.e., the classical Policy Gradient Theorem [17] or Bellman's equations, break down. Generalizations of the PG Theorem for general utilities [18] which express the gradient as product of the partial derivative of the utility respect to the occupancy measure, and the occupancy measure with respect to the policy, cannot overcome the fact that the later factor is a *global nonlinear function* of agents' policies, which does not permit decentralization. Therefore, in recent work, we define an agent's local occupancy measure as the joint occupancy measure of all agents' polices with all others' marginalized out, and its local general utility as any function of its marginal occupancy measure. The team objective, then, is the global aggregation of all local utilities.

Armed with this definition, we previously put forth **D**ecentralized **S**hadow Reward **A**ctor-**C**ritic (DSAC) for MARL with general utilities, and established its consistency and sample complexity [19]. This algorithm may be interpreted as a generalization of classical actor-critic algorithms for the multi-agent setting in [8], which are restricted to the cumulative return. It operates in four stages for each agent: (i) a marginalized occupancy measure estimation step used to evaluate the instantaneous gradient of the local utility with respect to the occupancy measure, which we dub the "shadow reward" (shadow reward computation); (ii) accumulate "shadow rewards" along a trajectory to estimate "shadow" critic parameters (critic); (iii) average critic parameters with those of its neighbors (information mixing); and (iv) a stochastic policy gradient ascent step along trajectories (actor).

Unfortunately, DSAC requires significant communications overhead between agents in order to operate: agents need to communicate shadow critic parameter estimates possibly multiple times for every policy update, depending on the desired convergence rate. This is a drawback common to classical techniques for multi-agent optimization: a weighted averaging step is conventionally employed in order to diffuse information between agents across time while optimizing

The author list is in alphabetical order.

[1]U.S. Army Research Laboratory, Adelphi, MD 20783, USA. E-mails: alec.e.koppel.civ@mail.mil, amrit0714@gmail.com

[2]Dept. of Electrical Engineering, Princeton University {mengdiw}@princeton.edu

local utilities [20], inspired by flocking [21] and gossip protocols [22]. We note that alternatives based upon Lagrange multipliers [23] may more effectively enforce consensus, but primal-only approaches are simpler and directly compatible with Perron-Frobenius theory [24], which set forth conditions on the network mixing matrices to ensure consensus.

Communications Efficiency: In this work, we enhance the communications efficiency of DSAC under two different models of message passing: a deterministic time-varying network model [20], and an event triggering scheme [25], where agents only communicate when their local shadow critic parameter estimates change more than a threshold. The result, Communications Efficient DSAC (CE-DSAC), broadens recent efforts toward communication-efficient MARL from the cumulative return (see [26]) to general utilities. In particular, we establish convergence to stationarity for both time-varying (Theorem IV.6) and event-triggered communications (Theorem IV.9) which, implies global optimality when the general utility is concave. Experimentally, we observe a favorable tradeoff between communications and policy learning for a cooperative multi-agent navigation problem.

II. Problem Formulation

In this work, we focus on Markov decision processes (MDP) over a finite state space $\mathcal{S}$ and finite action space $\mathcal{A}$. A transition to state $s' \in \mathcal{S}$ when starting from $s \in \mathcal{S}$ occurs upon selecting action $a \in \mathcal{A}$ according to a conditional probability distribution $s' \sim \mathcal{P}(\cdot|a,s)$, for which we define the short-hand notation $P_a(s,s')$. Denote as ξ the initial state distribution of the MDP, i.e., $s_0 \sim \xi$. We further denote $S := |\mathcal{S}|$ and $A := |\mathcal{A}|$ as the number of states and actions. Consider policy optimization for maximizing general objectives that are nonlinear function of the *cumulative discounted state-action occupancy measure* under policy π [18], [27]:

$$\max_{\pi} R(\pi) := F(\lambda^{\pi}) \tag{1}$$

where F is a general (not necessarily concave) functional and λ^{π} is aforementioned occupancy measure given by

$$\lambda^{\pi}(s,a) = \sum_{t=0}^{\infty} \gamma^t \mathbb{P}\Big(s^t = s, a^t = a \Big| \pi, s^0 \sim \xi\Big),\ \forall (s,a) \in \mathcal{S}\times\mathcal{A} \tag{2}$$

In this work, we consider a multi-agent extension of the problem in (1), where the state space $\mathcal{S}$, the action space $\mathcal{A}$, the policy π, and the general utility F are decentralized among $N = |\mathcal{V}|$ distinct agents. Agents are defined by a time varying undirected graph $\mathcal{G}_k = (\mathcal{V}, \mathcal{E}_k)$ with vertex set $\mathcal{V}$ and edge set $\mathcal{E}_k$ for each step k. In this case, the state space is the product of N local spaces $\mathcal{S}_i$, i.e., $\mathcal{S} = \mathcal{S}_1 \times \mathcal{S}_2 \times \cdots \times \mathcal{S}_N$ with $s = (s_{(1)}, s_{(2)}, \cdots, s_{(N)})$ and $s_{(i)} \in \mathcal{S}_i, i \in \mathcal{V}$. Similarly, the action space $\mathcal{A}$ is the product of N local spaces $\mathcal{A}_i$: $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2 \times \cdots \times \mathcal{A}_N$, meaning that for any $a \in \mathcal{A}$, we may write $a = (a_{(1)}, a_{(2)}, \cdots, a_{(N)})$ with $a_{(i)} \in \mathcal{A}_i, i \in \mathcal{V}$. Each agent has access to the global state s, as customary of joint-action learners training in a decentralized manner under full observability [8], [28]–[31]. Full observability means each agent i may access global actions a concatenating all local ones.

We assume the network of agents follows global policy $\pi(a|s)$ that maps global action a for a given global state s, which defined as the product of local policies $\prod_{i=1}^{N} \pi^{(i)}(a_{(i)}|s)$, which prescribes statistical independence among agents' policies. For the parameterized policy $\pi_\theta(a|s)$ where $\theta \in \Theta$, we denote $\theta = (\theta_1, \theta_2, \cdots, \theta_N)$ as the parameter, so we can write $\pi_\theta(a|s) = \prod_{i\in\mathcal{V}} \pi^{(i)}_{\theta_i}(a_{(i)}|s)$, where the local policy of agent i is parameterized by θ_i. Since the global state is visible to all agents, the *local policy* is based on the observation of the *global state*. The parameters θ_i are kept private by agent i.

Similar to the global occupancy measure $\lambda^{\pi}(s,a)$ [cf. (2)], we define the *local cumulative state-action*:

$$\lambda^{\pi}_{(i)}(s_{(i)},a_{(i)}) = \sum_{t=0}^{\infty} \gamma^t \cdot \mathbb{P}\Big(s^t_{(i)} = s_{(i)}, a^t_{(i)} = a_{(i)} \Big| \pi, s^0 \sim \xi\Big) \tag{3}$$

for $\forall a_{(i)} \in \mathcal{A}_i, s_{(i)} \in \mathcal{S}_i$, which is the marginalization of the global occupancy measure with respect to all others', whose indices are denoted as $\{-i\} \subset \mathcal{V}$. Marginalization allows us to write

$$\lambda^{\pi}_{(i)}(s_{(i)}, a_{(i)}) = \sum_{a\in\{a_{(i)}\}\times\mathcal{A}_{-i}} \sum_{s\in\{s_{(i)}\}\times\mathcal{S}_{-i}} \lambda^{\pi}(s,a) \tag{4}$$

with $\mathcal{A}_{-i}$=$\Pi_{j\neq i}\mathcal{A}_j$ and $\mathcal{S}_{-i}$=$\Pi_{j\neq i}\mathcal{S}_j$. The *local state-action* occupancy measure is a linear transform of λ^{π} in (2).

Let $S_i = |\mathcal{S}_i|$ denote the number of local states and $A_i := |\mathcal{A}_i|$ the number of local actions. For agent i, define the local utility function $F_i(\cdot) : \mathbb{R}^{S_i A_i} \mapsto \mathbb{R}$ as a function of $\lambda^{\pi}_{(i)}$, depends on θ_i when agent i follows policy π_{θ_i}. Then, define the global utility as the sum of local ones:

$$R(\pi_\theta) = F(\lambda^{\pi_\theta}) := \frac{1}{N} \sum_{i=1}^{N} F_i\big(\lambda^{\pi_\theta}_{(i)}\big). \tag{5}$$

Observe that the local utility of agent i is not node-separable with respect to policy parameters θ_i due to the dependence of the local occupancy measure (4) on the global policy π. This is a key point of departure from standard multi-agent optimization [20]: the global utility (5) is *not node-separable*. Next we shift to deriving a variant of actor-critic attuned to the multi-agent setting with general utilities (5) which operates upon the principle of partial linearization [32].

III. Elements of MARL with General Utilities

We develop an actor-critic type algorithm for MARL with general utilities (5). Technical challenges emerge as the occupancy measure, the policy parameters, and the utility are coupled. That is, the general utility, in contrast to the standard value function, is not additive across trajectories, which invalidates RL approaches based upon either the Policy Gradient Theorem [17] or Bellman's equations [1].

We chart a course through these issues based upon a combination of the chain rule, a density estimation step, and the construction of a "shadow reward." The result is a MARL scheme in which the critic step operates through an occupancy measure estimation step, a policy evaluation step with respect to the "shadow value function" (critic). Then the actor update is a typical stochastic ascent step. Further, in the critic update, agents compute local weighted averages

to diffuse information across the team. To proceed, we begin by defining the shadow reward and value function.

A. Shadow Rewards and Policy Evaluation

As previously mentioned, the general utility (1) cannot be written as cumulative sum of returns, which is critical to the original definition of the reward function and Q function in dynamic programming [1] or policy search [17]. To circumvent the need for additivity, we introduce auxiliary variables called shadow rewards and shadow Q functions.

Definition III.1 (Shadow Reward and Shadow Q Function)**.** *The shadow reward function $r_\pi : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ of a policy π corresponding to a general utility F is $r^\pi(s,a) := \frac{\partial F(\lambda^\pi)}{\partial \lambda(s,a)}$, and its associated shadow Q function is*

$$Q_F^\pi(s,a) := \mathbb{E}\Big[\sum_{t=0}^{+\infty} \gamma^t \cdot r^\pi(s^t,a^t) \,\Big|\, s^0 = s, a^0 = a, \pi\Big].$$

For the shadow Q function and the occupancy measure, we assume they are smooth respect to policy parameters.

Assumption III.2. *$\exists \ell_Q, \ell_\lambda > 0$ s.t. for $\forall (s,a) \in \mathcal{S} \times \mathcal{A}$, $\forall \theta, \theta'$, it holds that $|Q_F^{\pi_\theta}(s,a) - Q_F^{\pi_{\theta'}}(s,a)| \le \ell_Q \|\theta - \theta'\|$, and $|\lambda^{\pi_\theta}(s,a) - \lambda^{\pi_{\theta'}}(s,a)| \le \ell_\lambda \|\theta - \theta'\|$.*

These definitions may be understood by considering the linearization of general utility F with respect to λ^π, which, via the chain rule, is equivalent to a MDP with cumulative return, with the shadow reward and Q function in place of the usual the reward and Q functions:

$$\nabla_\theta F(\lambda^{\pi_\theta}) = \mathbb{E}\Big[\sum_{t=0}^{+\infty} \gamma^t \cdot Q_F^\pi(s^t,a^t) \cdot \nabla_\theta \log \pi_\theta(a^t|s^t) \,\Big|\, s_0 \sim \xi, \pi\Big]. \quad (6)$$

This expression for the policy gradient illuminates the centrality of the shadow reward/value function for nonlinear functions of the occupancy measure (2), which motivates the generalized policy evaluation scheme we present next.

B. Policy Evaluation Criterion for Shadow Q Function

We seek to compute the Shadow Q-function from trajectory information to form the target value for the parameters of a critic. We consider the case that the critic is parameterized. One simple choice is linear function approximation, i.e., with a set of feature vectors $\{\phi(s,a) \in \mathbb{R}^d : s \in \mathcal{S}, a \in \mathcal{A}\}$, we want to find some weight parameter $w \in \mathbb{R}^d$ so that

$$Q_w(s,a) := \langle \phi(s,a), w \rangle \quad \forall (s,a) \in \mathcal{S} \times \mathcal{A}. \quad (7)$$

In our algorithm, we will update a sequence of $\hat{w}$ to closely approximate the sequence of implicit shadow Q functions, as the policy is updated. In practice, the parametrization (7) need not be linear – in Section V, we define Q as a multi-layer neural network.

Thus, the critic objective at policy π may be defined as the mean-square-error of a regressor w.r.t. shadow Q-function:

$$\begin{aligned}\ell(w;\pi) &:= \mathbb{E}\Big[\sum_{t=0}^{\infty} \frac{\gamma^t}{2} \big(Q_w(s^t,a^t) - Q_F^\pi(s^t,a^t)\big)^2 \,\big|\, s^0 \sim \xi, \pi\Big] \\ &= \frac{1}{2} \sum_{s,a} \lambda^\pi(s,a) \big(\phi(s,a)^\top w - Q_F^\pi(s,a)\big)^2. \quad (8)\end{aligned}$$

Via the definition of the occupancy measure λ^π [cf. (2)], the expectation may be substituted by weighting factors in the summand on the second line. We assume the set of features $\{\phi(s,a)\}_{s\in\mathcal{S}, a\in\mathcal{A}}$ are always bounded, formalized next.

Assumption III.3. *$\exists C_\phi > 0$ s.t. $\|\phi(s,a)\| \le C_\phi$, $\forall (s,a)$.*

Through the boundedness of the features, which holds, e..g, for radial basis functions or auto-encoders with bounded range, we may establish critic objective function has Lipschitz continuous gradients.

Proposition III.4. *Regardless of policy π_θ, the critic objective $\ell(w;\pi_\theta)$ [cf. (8)] is L_w-smooth with $L_w = \frac{C_\phi^2}{1-\gamma}$.*

Proposition III.4 may be established by evaluating the Hessian of the critic objective function (8): $\nabla_w^2 \ell(w;\pi_\theta) = \sum_{s,a} \lambda^{\pi_\theta}(s,a) \cdot \phi(s,a)\phi(s,a)^\top$. Consequently, $L_w \le \|\nabla_w^2 \ell(w;\pi_\theta)\|_F \le \frac{C_\phi^2}{1-\gamma}$. With the shadow reward and associated Q-function (Definition III.1), the policy evaluation criterion (8), and its smoothness properties (Proposition III.4) with respect to the critic parameters w [cf. (7)] in place, we expand on their role in the multi-agent setting.

C. Multi-Agent Optimization for Critic Estimation

Setting aside the issue of policy parameter updates, we focus on estimating the global general utility. The shadow Q-function and shadow reward (Definition III.1) depend on global knowledge of all local utilities, which are unavailable as local incentives are local only. Therefore, introduce their localized components r_i^π for agent i, which together comprise the global shadow Q-function and reward:

$$r_i^\pi(s_{(i)}, a_{(i)}) := \frac{\partial F_i(\lambda_{(i)}^\pi)}{\partial \lambda_{(i)}(s_{(i)}, a_{(i)})}, \forall (s_{(i)}, a_{(i)}) \in \mathcal{S}_i \times \mathcal{A}_i \quad (9)$$

Observe that $r^\pi(s,a) = \frac{1}{N}\sum_{i=1}^N r_i^\pi(s_{(i)}, a_{(i)})$. Based on observing local shadow reward, agent i may access its local component of the global shadow Q-function $Q : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$:

$$Q_i^\pi(s,a) := \mathbb{E}\Big[\sum_{t=0}^{+\infty} \gamma^t \cdot r_i^\pi\big(s_{(i)}^t, a_{(i)}^t\big) \,\Big|\, s^0 = s, a^0 = a, \pi\Big] \quad (10)$$

for $\forall (s,a) \in \mathcal{S} \times \mathcal{A}$, which also implies $Q_F^\pi(s,a) = \frac{1}{N}\sum_{i=1}^N Q_i^\pi(s,a)$. Then, each agent i seeks to estimate common critic parameters w that well-represent its shadow Q function in the sense of minimizing the global mean-square error (8). The aforementioned node-separability, together wit introducing a localized critic parameter vector w_i associated to agent i, allows us to formulate a consensus problem:

$$\min_{\{w_i\}_{i=1}^N} \frac{1}{N}\sum_{i=1}^N \ell_i(w_i;\pi) \;\; \text{s.t.} \;\; w_i = w_j, \, (i,j) \in \mathcal{E} \quad (11)$$

with local policy evaluation criterion

$$\ell_i(w_i;\pi) := \mathbb{E}\Big[\sum_{t=0}^{\infty} \frac{\gamma^t}{2} \big(Q_{w_i}(s^t,a^t) - Q_{F_i}^\pi(s^t,a^t)\big)^2 \,\big|\, s^0 \sim \xi, \pi\Big]. \quad (12)$$

By identifying the shadow critic estimation problem as a consensus problem, we may develop solutions that allow

Algorithm 1: CE-DSAC Algorithm

Input: initial policy θ^0; actor & critic step-sizes $\{\eta_\theta^k, \eta_w^k\}$; Batch sizes $\{B_k\}$; Episode lengths $\{H_k\}$; initial critic $W^0 := [w_1^0, w_2^0, ..., w_N^0]$, $w_i^0 = w_j^0$; weight matrices $\{M_k\} \subseteq \mathbb{R}_+^{N\times N}$; mixing rounds $m \geq 1$.

for *iteration* $k = 0, 1, 2, ...$ **do**

  Perform B_k Monte Carlo rollouts to obtain trajectories $\tau = \{s^0, a^0, \cdots, s^{H_k}, a^{H_k}\}$ with initial dist. ξ, policy π_{θ^k} collected as batch $\mathcal{B}_k$.

  for *agent* $i = 1, 2, ..., N$ **do**

    Compute empirical local occupancy measure

$$\hat{\lambda}_i^k = \frac{1}{B_k}\sum_{\tau\in\mathcal{B}_k}\sum_{t=0}^{H_k}\gamma^t\cdot\mathbf{e}\Big(s_{(i)}^t, a_{(i)}^t\Big). \qquad (13)$$

    Estimate shadow reward $\hat{r}_i^k = \nabla_{\lambda_i} F_i(\hat{\lambda}_i^k)$.

  for *agent* $i = 1, 2, ..., N$ **do**

    With $G_{w_i}(\cdot)$ defined in (14), compute

$$\widehat{\Delta}_{w_i}^k = \frac{1}{B_k}\sum_{\tau\in\mathcal{B}_k} G_{w_i}(\tau, \hat{r}_i^k, w_i^k),\ w_i^{k+1} = w_i^k - \eta_w^k\,\widehat{\Delta}_{w_i}^k.$$

  for $iter = 1, ..., m$ **do**

    for *agent* $i = 1, 2, ..., N$ **do**

      Exchange information with neighbours: $w_i^{k+1} = \sum_{\{j:(j,i)\in\mathcal{E}\}} M_k(j,i)\cdot w_i^{k+1}$.

  With $G_{\theta_i}(\cdot)$ defined in (15), update the policy:

$$\widehat{\Delta}_{\theta_i}^k := \frac{1}{B_k}\sum_{\tau\in\mathcal{B}_k} G_{\theta_i}(\tau, w_i^{k+1})\ ,\ \theta_i^{k+1} = \theta_i^k + \eta_\theta^k\,\widehat{\Delta}_{\theta_i}^k.$$

Algorithm 2: Event Triggered communication

Input: Every agent i keeps a record of neighbour j's last communication: $\hat{w}(i,j)$. If $k = 0$, $\hat{w}(i,j) = w_j^0$.

for $iter = 1, 2, ...m$ **do**

  for *agent* $i = 1, 2, ..., N$ **do**

    if $\|w_i^{k+1} - \hat{w}(i,i)\| \geq \epsilon_{k+1}$ **then**

      Agent i update $\hat{w}(i,i) \leftarrow w_i^{k+1}$.

      Agent i sends $\hat{w}(i,i)$ to its neighbours. Namely, for all $j \in \mathcal{N}(i)$, update

$$\hat{w}(j,i) \leftarrow \hat{w}(i,i)$$

  for *agent* $i = 1, 2, ..., N$ **do**

    Exchange information with neighbours: $w_i^{k+1} = \sum_{\{j:(j,i)\in\mathcal{E}\}} M_k(j,i)\cdot\hat{w}(i,j)$.

agent i to evaluate its policy with respect to global utility (5) through the local criterion (12) under the consensus constraint on its local parameters w_i. Next, we incorporate solutions to (11) into the critic step with a policy update for parameters θ_i along stochastic ascent directions via (6) to assemble DSAC.

D. Decentralized Shadow Reward Actor-Critic

Next, we put together these pieces to present **C**ommunication-**E**fficient **D**ecentralized **S**hadow Reward **A**ctor-**C**ritic (CE-DSAC) as Algorithm 1. This scheme allows agents to keep their local utilities F_i, and policies π_{θ_i} with associated parameters θ_i private. The agents share a common function approximator for the shadow Q function. Further, they retain local copies w_i of the shadow critic parameters, which they communicate to neighbors according to the network structure defined by edge set $\mathcal{E}_k$ and mixing matrix M_k to be subsequently specified.

Algorithm 1 proceeds in four stages: (i) density estimation step for to obtain the shadow reward; (ii) shadow critic updates; (iii) information mixing via weighted averaging; and (iv) actor updates. These stages require access to trajectories through the MDP for all the agents. In a broad sense, assuming fixed policy and critic parameterizations for each agent, one may implement actor-critic. The major departure is that individual agents estimate shadow rewards based on the empirical *marginal occupancy measure*, which has significantly smaller dimension S_iA_i than that of the global state S and action spaces A. Additionally, a message passing step based upon weighted averaging is incorporated. Each step is detailed next.

(i) Occupancy Measure Estimation. Via trajectory τ, each agent i evaluates its current policy with respect to the general utility. This is accomplished by computing its shadow reward, by first executing a local empirical occupancy measure estimator$\hat{\lambda}_i^k$ by (13). Then, the shadow reward is compute as $\hat{r}_i^k = \nabla_{\lambda_i} F_i(\hat{\lambda}_i^k)$.

(ii) Shadow Policy Evaluation. The shadow reward (9) is then accumulated along the trajectory to form the local policy evaluation error ℓ_i [cf. (12)] with respect to the shadow Q function (10). Note that the shadow Q function $Q_{F_i}^\pi(s^t, a^t)$ is substituted by an empirical estimate along the current trajectory. Specifically, $\hat{Q}_i^t = \sum_{t'=t}^H \gamma^{t'-t}\cdot r_i\big(s_{(i)}^{t'}, a_{(i)}^{t'}\big)$ is the accumulation of rewards starting from (s^t, a^t). Then, differentiating the resulting expression with respect to local critic parameters w_i yields the critic gradient direction:

$$G_{w_i}(\tau, r_i, w_i) = \sum_{t=0}^H \gamma^t\cdot(Q_{w_i}(s^t, a^t) - \hat{Q}_i^t)\cdot\nabla_{w_i} Q_{w_i}(s^t, a^t), \quad (14)$$

where agent i then uses to update its local shadow critic as $\hat{w}_i^{k+1} = w_i^k - \eta_w^k\cdot\widehat{\Delta}_{w_i}^k$ at step $k+1$, under the initialization with $w_1^0 = \cdots w_N^0$ and step-size η_w^k specified as in Theorem IV.6. Moreover, $\widehat{\Delta}_{w_i}^k$ is a mini-batched version of the stochastic gradient in (14) specified in Algorithm 1.

(iii) Information Exchange. To ensure information effectively propagates across the network $\mathcal{G}_k$, agents perform a simple weighted averaging step using mixing matrix M_k, which is a symmetric doubly stochastic matrix that respects the edge connectivity of the graph, see Assumption IV.3 for details. When agents execute m-steps of averaging per step k, we compactly express it as $W^{k+1} \leftarrow M_k^m\cdot W^{k+1}$. We note that the time-varying connectivity of graph results in

less number of edges across the network as compared to a fully connected network as long as the edge connectivity is respected. This allows to reduce communication requirements at each step k in the network.

(iv) Policy Update. Given the Q-function approximation parameter w_i, the policy gradient is constructed as

$$G_{\theta_i}(\tau, w_i) = \sum_{t=0}^{H} \gamma^t Q_{w_i}(s^t, a^t) \nabla_{\theta_i} \log \pi_{\theta_i}^{(i)}(a_{(i)}^t | s^t). \quad (15)$$

which is a stochastic approximation of the gradient in (6). Notice that replacing $Q_{w_i}(s^t, a^t)$ with the exact shadow Q-function $Q_F(s^t, a^t)$ reduces (15) to the REINFORCE [33] estimator equipped with the newly defined shadow Q-function. Then, each agent executes a simple mini-batch stochastic gradient ascent step.

Event-Triggered Communications. Now we propose a variant of CE-DSAC where agents only transmit local critic parameters if they are statistically significant, which refines communication module of Algorithm 1. In particular, the event-triggered communication module summarized in Algorithm 2 dictates that agent i tracks an auxiliary variable $\hat{w}(i,i)$ which is the value of its critic parameters at the previous time instance, and sends its parameters w_i^{k+1} to neighbors $j \in \mathcal{N}(j)$ only if the change exceeds an ϵ-threshold in magnitude, i.e., $\|w_i^{k+1} - \hat{w}(i,i)\| \geq \epsilon_{k+1}$. Otherwise, its neighbors use the previous value $\hat{w}(j,i)$. The event-triggered variant employs Algorithm 2 in place of steps 8-10 in Algorithm 1.

IV. Consistency and Sample Complexity

In this section, we study the convergence rate of agents' policies θ_i when following Algorithm 1 to stationary points of global utility (5). Our key result builds upon the fact that weighted averaging (Sec. III-D(iii)) causes agents' critic parameter estimates to tend to the globally aggregated shadow value function [cf (11)] at rates common to multi-agent optimization [20]. Then, we analyze the evolution of the attenuation of the gradient norm of the general utility (5), which is bottlenecked by the trajectory subsampling error, a function approximation error term depending on the richness of the chosen features in (7), a decreasing function of the iteration index, and a consensus error. Overall, we obtain that to achieve ϵ-stationarity, $\tilde{\mathcal{O}}(1/\epsilon^{2.5})$ (Theorem IV.6) or $\mathcal{O}(1/\epsilon^2)$ (Corollary IV.8) samples are required, depending on the number of communications per step, akin to best known rates for non-concave expected maximization problems [34]. We also establish that for this setting, there are no spurious extrema, meaning that agents obtain a globally optimal policy (Corollary IV.7).

Before continuing, we introduce several technical conditions which are required for the analysis. Specifically, for the utility function F, the parameterization π_θ, and the shadow Q-function approximation Q_w, we assume the following.

Assumption IV.1. *For utility F [cf. (5)], we assume:*
(i) Local utility function $F_i(\cdot)$ is private to agent i.
(ii) For local utility F_i, $\exists C_F > 0$ s.t. $\|\nabla_{\lambda_{(i)}} F_i(\lambda_{(i)})\|_\infty \leq C_F$ in a neighbourhood of the occupancy measure set.
(iii) For $\forall i \in \mathcal{V}$, $\exists L_\lambda > 0$ s.t. $\|\nabla_{\lambda_{(i)}} F_i(\lambda_{(i)}) - \nabla_{\lambda_{(i)}} F_i(\lambda'_{(i)})\|_\infty \leq L_\lambda \|\lambda_{(i)} - \lambda'_{(i)}\|$.
(iv) $\exists L_\theta > 0$ s.t. $\|\nabla_\theta F(\lambda^{\pi_\theta}) - \nabla_\theta F(\lambda^{\pi_{\theta'}})\| \leq L_\theta \|\theta - \theta'\|$.

Assumption IV.2. *For the parameterization π_θ and the occupancy measure λ^{π_θ} we assume the following holds:*
(i) The local policy $\pi_{\theta_i}^{(i)}$ is private to the agent i.
(ii) $\exists\ C_\pi > 0$ s.t. for each agent i, its score function is bounded: $\|\nabla_{\theta_i} \log \pi_{\theta_i}^{(i)}(a_{(i)}|s)\| \leq C_\pi$, for $\forall\ \theta$ and $\forall (s,a)$.
(iii) $\exists\ \ell_\theta > 0$ s.t. $\|\lambda^{\pi_\theta} - \lambda^{\pi_{\theta'}}\| \leq \ell_\theta \|\theta - \theta'\|$.

For mixing matrices $\{M_k\}$, we require the following.

Assumption IV.3. *In Algorithm 1, the mixing matrix M_k is a doubly stochastic matrix satisfying the following properties:*
(i) $M_k \in \mathbb{R}_+^{N \times N}$ is symmetric with $M_k(i,j) > 0$ at edges $(i,j) \in \mathcal{E}_k$.
(ii) $M_k \cdot \mathbf{1}_N = \mathbf{1}_N$, where $\mathbf{1}_N \in \mathbb{R}^N$ is the all-ones vector.
(iii) The eigenvalues of M_k satisfy $1 = \sigma_1(M_k) > \sigma_2(M_k) \geq \cdots \geq \sigma_N(M_k)$, and $\sup_{k \geq 0} \max\{|\sigma_2(M_k)|, |\sigma_N(M_k)|\} < \rho < 1$. (iv) For the event triggered communication case, we assume the network is static, i.e., $M_t \equiv M, \forall t \geq 0$.

Throughout the iterations of Algorithm 1, we also make the following assumption on the critic objective function.

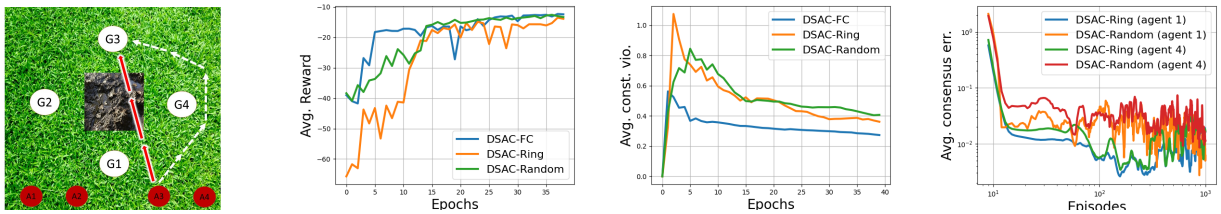
Assumption IV.4. *$\ell(w; \pi_{\theta^k})$ is μ_w-strongly convex for all k.*

Assumption IV.1 prescribes boundedness of the gradients of the general utility w.r.t the occupancy measure, as well as its Lipschitz continuity, together with the gradient of the general utility with respect to the policy parameters. Together, these conditions ensure smoothness of the general utility with respect to policy parameters. Assumption IV.2 ensures that the score function is bounded, and that the occupancy measure is smooth with respect to policy parameters. These conditions are common to reinforcement learning algorithms focusing on occupancy measures in recent years [12], [18], and are automatically satisfied by common policies such as the softmax. Assumption IV.3 holds for any undirected connected graph [24].

Lastly, Assumption IV.4 means that the minimum eigenvalue of the feature covariance matrix $\sum_{s,a} \lambda^{\pi_{\theta^k}}(s,a) \cdot \phi(s,a)\phi(s,a)^\top$ is uniformly lower bounded by some constant $\mu_w > 0$. Note that the shadow reward r^π is changing with iteration index k, and consequently we cannot assume that the fitted shadow Q-function perfectly tracks the true shadow Q-function. Motivated by the subtleties of the quality of a feature representation, we further place a condition on the shadow value function approximation error.

Assumption IV.5. *For any parameterized policy π_θ, we denote optimally fitted critic parameter as $w^*(\theta)$. Namely, $w^*(\theta) := \operatorname{argmin}_w \frac{1}{N} \sum_{i=1}^N \ell_i(w; \pi_\theta)$ [cf. (11)]. Then we assume the following feature mis-specification error is uniformly upper bounded by some constant $W > 0$:*

$$E_\theta^2 := \sum_{i=1}^N \|\nabla_{\theta_i} F(\lambda^{\pi_\theta}) - \Delta_{\theta_i}\|^2 \leq W, \quad \forall \theta \quad (16)$$



(a) Environment (b) Average return (c) Average cost (d) Consensus error

Fig. 1: Safe navigation in a multi-agent cooperative environment with 4 agents and 4 landmarks as an instantiation of a constrained MDP, where constraints are imposed to avoid the muddy region while reaching the goal. Note that the state space in this case would be 16 dimensional (location of agent and landmarks). We run this experiment for three different time-varying communication graphs; *fully connected (FC) (all the agents are connected to each other)*, *ring* (all the agents are connected using ring topology which changes for each iteration), and *random* (where agents are randomly using Erdős-Rényi random graph model at each iteration). **(a)** We plot running average of the reward return. **(b)** We plot the running average of the constraint violation. **(c)** We plot the running average of the consensus error for agent 1 and agent 4 for *ring* and *random* network.

where $\Delta_{\theta_i} := \mathbb{E}\big[\sum_{t=0}^{+\infty}\gamma^t\cdot Q_{w^*(\theta)}(s^t,a^t)\cdot\nabla_{\theta_i}\log\pi_{\theta_i}^{(i)}(a^t|s^t)\,\big|s^0\sim\xi,\pi_\theta\big]$ *is the PG estimate under* $w^*(\theta)$*, and* $Q_w(s,a)$ *is function approximation of the shadow Q function [cf (7)].*

The E_θ^2 defined above quantifies the degree of misspecification of the features used to define (7). Note that if they are chosen by a universal function approximator such radial basis functions centered at a dense grid across the state-action space, or a well-calibrated auto-encoder, then the shadow Q-function approximation may be near perfect, i.e., $E_\theta^2\approx 0$.

To analyze how Algorithm 1, we describe the high level idea of the proofs here. The proofs are deferred to a future journal version of this work. We begin by a standard stochastic gradient ascent analysis, illuminating its dependence on the gradient estimation error $\|\sum_{i=1}^N\widehat{\Delta}_{\theta_i}^k-\nabla_{\theta_i}F(\lambda^{\pi_{\theta^k}})\|^2$, which may be decomposed into attenuating error terms and a term associated with decentralization $\sum_{i=1}^N\|w_i^{k+1}-w_*^{k+1}\|^2$, where we denote $w_*^{k+1}=w^*(\theta^k)$ for the ease of notation. The later term is split two parts: a consensus error and the optimality gap of the critic fitting problem.

A subtlety emerges due to the presence of the persistent consensus error $\|\bar{w}^{k+1}-w_*^{k+1}\|^2$. That is, $F(\lambda^{\pi_{\theta^k}})$ may not be increasing (except for the $\mathcal{O}(B_k^{-1}+\gamma^{2H_k})$ stochastic estimation errors), which motivates us to carefully construct the following potential function: $R_k:=F(\lambda^{\pi_{\theta^k}})-\alpha\|\bar{w}^k-w_*^k\|^2$ with $\alpha=\frac{18C_\phi^2C_\pi^2}{(1-\gamma)^2\mu_w}\cdot\max_{k\geq 0}\{\eta_\theta^k/\eta_w^k\}$. Based on this potential function, we could characterizes the algorithm performance in terms of optimization error, the feature mis-specification error, the stochastic PG approximation error, and the multi-agent consensus error. By carefully specifying the algorithmic parameters, then, we have the following.

Theorem IV.6. *Suppose the Assumption III.3, IV.1, IV.2, IV.4 and IV.3 hold, and there is only one round of communication in each iteration, i.e.* $m=1$*. Then, Algorithm 1, under the following parameter selections:*

(i) For final iteration $T=\mathcal{O}(\epsilon^{-1.5})$*, trajectory lengths* $H_k\equiv\mathcal{O}\left(\frac{\log(1/\epsilon)}{1-\gamma}\right)$*,* $\delta_k\equiv\frac{\delta}{3N(T+1)}$*,* $\delta\in(0,1)$*, batch sizes* $B_k\equiv\log(1/\delta_k)\epsilon^{-1}$*, constant step-sizes* $\eta_w=\mathcal{O}(\sqrt{\epsilon})$*,* $\eta_\theta=\min\left\{\frac{(1-\gamma)\mu_w\eta_w}{C_wC_\phi C_\pi}\cdot\frac{1}{6\sqrt{10N}},\frac{1}{4L_\theta}\right\}=\mathcal{O}(\sqrt{\epsilon})$*,*

$$\frac{1}{T}\sum_{k=1}^T\|\nabla_\theta F(\lambda^{\pi_{\theta^k}})\|^2\leq\mathcal{O}\left(\epsilon+W\right).\quad w.p.\quad 1-\delta$$

(ii) For unspecified final iteration T*, we adaptively set:* $\delta_k=\frac{2\delta}{N\pi^2(k+1)^2}$*,* $\delta\in(0,1)$*, trajectory lengths* $H_k=\mathcal{O}((1-\gamma)^{-1}\log(k+1))$*, batchsizes* $B_k=\log(1/\delta_k)(k+1)^{\frac{2}{3}}$*, and step-sizes* $\eta_\theta^k=\min\left\{\frac{(1-\gamma)\mu_w\eta_w^{k+1}}{C_wC_\phi C_\pi}\cdot\frac{1}{6\sqrt{10N}},\frac{1}{4L_\theta}\right\}$*,* $\eta_w^k=\min\{(k+1)^{-\frac{1}{3}},L_w^{-1}\}$*, then*

$$\frac{\sum_{k=1}^T\eta_\theta^k\|\nabla_\theta F(\lambda^{\pi_{\theta^k}})\|^2}{\sum_{k=1}^T\eta_\theta^k}\leq\mathcal{O}\left(\frac{\log T}{T^{\frac{2}{3}}}+W\right),\quad w.p.\quad 1-\delta$$

In either case, Algorithm 1 requires $\tilde{\mathcal{O}}(\epsilon^{2.5})$ *samples to satisfy* $\frac{\sum_{k=1}^T\eta_\theta^k\|\nabla_\theta F(\lambda^{\pi_{\theta^k}})\|^2}{\sum_{k=1}^T\eta_\theta^k}\leq\mathcal{O}(\epsilon+W)$*.*

Next, we establish that for concave general (1) (and hence local (5)) utilities, there are no spurious stationary points.

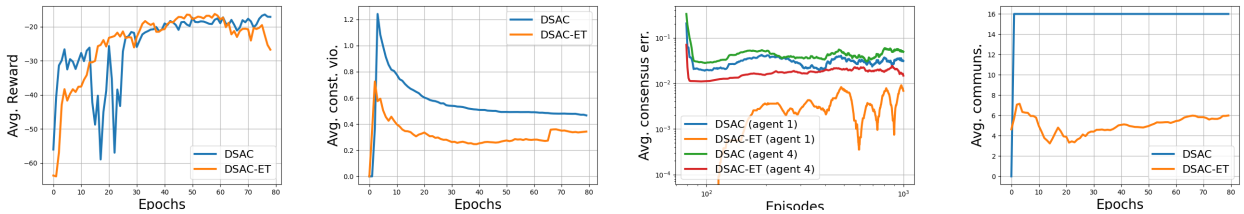
Corollary IV.7 (Convergence to global optimality)**.** *Suppose the general utility* F *is concave, and the shadow value function* Q_F *is realizable, i.e.,* $W=0$ *in* (16)*. Then for policies* π_θ *satisfying Assumption 1 of [18], every stationary point is a global optimizer. In Theorem IV.6(ii), if we further let* $\bar{\theta}_T$ *be the parameter randomly chosen from* $\{\theta^k\}_{k=1}^T$ *where* $\bar{\theta}_T=\theta^k$ *w.p.* $\eta_\theta^k/(\sum_{k'=1}^T\eta_\theta^{k'})$*, then* $\lim_{T\to\infty}\mathbb{E}[\|\nabla_\theta F(\lambda^{\pi_{\bar{\theta}_T}})\|^2|\mathcal{F}_T]=0$ *w.p.* $1-\delta$*. Thus, Algorithm 1 converges to the global optima.*

The preceding result is about the asymptotic performance of Algorithm 1. Next we spotlight the role of the number of communication steps in the convergence rate.

Corollary IV.8 (Multiple-round communication)**.** *Suppose we allow multiple steps of information exchange, i.e.,* $m>1$*. Then under the same parameter selections as Theorem IV.6(i), but additionally setting final iteration index* $T=\epsilon^{-1}$*, number of communication rounds* $m=\mathcal{O}((1-\rho)^{-1}\log(\epsilon^{-1}))$*, and the step-sizes* $\eta_\theta^k\equiv\min\left\{\frac{(1-\gamma)\mu_w/L_w}{C_wC_\phi C_\pi}\cdot\frac{1}{6\sqrt{10N}},\frac{1}{4L_\theta}\right\}$*,* $\eta_w^k\equiv L_w^{-1}$*, then the total sample complexity is* $\mathcal{O}(\epsilon^{-2})$*.*

This result specifies that with additional communication rounds $m=\mathcal{O}((1-\rho)^{-1}\log(\epsilon^{-1}))$ per actor update k, the convergence rate refines from $\mathcal{O}(\epsilon^{-2.5})$ to $\mathcal{O}(\epsilon^{-2})$. Following a similar strategy, we can show the sample complexity for Algorithm 1 when the event triggered communication scheme (Algorithm 2), is applied.

Theorem IV.9. *Suppose the Assumption III.3, IV.1, IV.2, IV.4 and IV.3 hold. Then, Algorithm 1, with the event-triggered*



(a) Average return (b) Average cost (c) Consensus error (d) No. of Communications

Fig. 2: Safe navigation in a multi-agent cooperative environment with 4 agents and 4 landmarks, an instance of a constrained MDP, where constraints are imposed to avoid the muddy region while reaching the goal – see Fig. 1 (a). Note that the state space in this case would be 16 dimensional (location of agent and landmarks). We run this experiment for fixed binomial graph with full communications (denoted by DSAC) and with event triggeting (denoted by DSAC-ET). **(a)** We plot running average of the reward return. **(b)** We plot the running average of the constraint violation. **(c)** We plot the running average of the consensus error for agent 1 and agent 4. **(d)** We plot the average number of communications happening per epochs in the network. We note that the algorithm is able to achieve the similar performance in terms of average return with less number of overall communications.

communication module (Algorithm 2), under the following parameter selections: $T = \epsilon^{-1}$, *trajectory lengths* $H_k \equiv \mathcal{O}\left(\frac{\log(1/\epsilon)}{1-\gamma}\right)$, $\delta_k \equiv \frac{\delta}{3N(T+1)}$, $\delta \in (0,1)$, *batch sizes* $B_k \equiv \log(1/\delta_k)\epsilon^{-1}$, *number of communication rounds* $m = \mathcal{O}((1-\rho)^{-1}\log(\epsilon^{-1}))$, *constant step-sizes* $\eta_\theta^k \equiv \min\left\{\frac{(1-\gamma)\mu_w/L_w}{C_w C_\phi C_\pi}, \frac{1}{6\sqrt{10N}}, \frac{1}{4L_\theta}\right\}$, $\eta_w^k \equiv L_w^{-1}$, *and* $\epsilon_k \equiv \mathcal{O}(\sqrt{\epsilon})$, *then*

$$\frac{1}{T}\sum_{k=1}^{T}\left\|\nabla_\theta F(\lambda^{\pi_{\theta^k}})\right\|^2 \leq \mathcal{O}\left(\epsilon + W\right). \quad w.p. \quad 1-\delta$$

The total sample complexity will be $\tilde{\mathcal{O}}(\epsilon^{-2})$.

Communication Efficiency: We remark that if α-fraction of the total Tm rounds of critic updates are smaller than the threshold ϵ_k for an agent i, and for some $\alpha \in (0,1)$, then potentially αTm rounds of communications are saved for agent i due to the event triggered mechanism. Next, we shift to investigating the experimental merit of the proposed approach.

V. Experimental Results

We experimentally investigate the merit of Algorithm 1 in the context of multi-agent problems. We experiment with $N = 4$ agents moving in a two-dimensional continuous space associated with the problem of *Safe Cooperative navigation* [35]. Since the state space is continuous, we use discretization of the state space to estimate the respective occupancy measure. Unless otherwise stated, we have used a 2 layer with 64 nodes per layer deep neural network (DNN) for the actor as well as critic in the experiments. We use a learning rate of 0.001 for all the experiments and a batch size of 10 for the count based density estimator. One epoch in the experiment consists of 1000 episodes unless otherwise stated and the maximum number of steps per episode are 50. We have reported running averages for all the results reported in this paper, such as, the general utility, the constraint violation, and the consensus error.

For the experiments, we consider a 4 agent cooperative environment from [35] where each agent needs to reach its assigned goal while traversing only through the safe region as visualized in Fig.2(a). In Fig.2(a), the red circle denotes the agents and white circle represents their corresponding goals. The aim here is to learn the trajectories such that it doesn't pass through the center of the region (mud in the middle of grassy region). The white arrows shows the preferred path in the figure. Agents receive a negative reward proportional to its distance from the landmark, and an additional negative reward of -1 if agents collide. Additionally, each agents receive a high cost of $c = 1$ if it passes through the unsafe region (middle of the state space) – see Fig. 2(a). Note that this behavior could be learned in a policy of a agent i via imposing a safety constraint for each agent $\langle \lambda_i^\pi, c\rangle \leq C$ where λ_i^π in the marginalized occupancy measure. This local constraint could be introduced into the global common objective as a quadratic penalty as $F(\lambda^\pi) = \frac{1}{N}\sum_{i=1}^N \langle \lambda_i^\pi, r_i\rangle - z\sum_{i=1}^N \left(\langle \lambda_i^\pi, c\rangle - C\right)^2$, where z is the penalty controlling parameter. Depending upon the connectivity among the agents, we divide the experiments into two parts: A) Time varying network, and B) Intermittent communications. Next, we provide further details for each set of experiment we presented in the paper.

A. Time-varying Network

We use the proposed DSAC algorithm to solve problem, and present the results for the average reward and constraint violation, respectively, in Fig. 1. For the experiments, we consider three different scenarios of network connectivity namely; *fully connected (FC)* (all the agents are connected to each other), *ring* (all the agents are connected using ring topology), and *random* (where agents are randomly connected using Erdős-Rényi random graph model with p being uniformly selected between 0 and 1). Note that even for the ring topology, the network graph changes at each iteration t. We plot the running average of the reward returns (Fig. 2(b)), running average of the constraint violations (Fig. 2(c)), and running average of the consensus error (Fig. 1(d)) for agent 1 and 4 in Fig. 1. Since the consensus error was converging to zero quickly, we have plotted it using log scale and episodes for the x axis.

B. Intermittent Communications

To show the effectiveness of the proposed scheme in Algorithm 2, we consider a 4 agent fully connected network to solve the *safe cooperative navigation* problem. For the experiment, we consider a constant threshold of $\epsilon = 0.03$

and compare the results to scenario when $\epsilon = 0$ (standard communication architecture) in Fig. 2, the performance with $\epsilon > 0$ is almost similar to the case when critic parameters are transmitted at each time. But we save a lot in terms of number of communications we need to perform as shown in Fig. 2(d). We remark that the proposed algorithm with event triggering is able to save a lot of communications and achieves the similar performance in terms of average return. This is really important from practical point of view because performing communication is more costly.

VI. Conclusions

We contributed a conceptual basis for defining agents' behavior in cooperative MARL beyond the cumulative return via nonlinear functions of their occupancy measure. This motivated defining "shadow rewards" and DSAC, whose critic employs shadow value functions and weighted averaging. We analyzed such a scheme on realistic communications models based upon fixed delay and event-triggered schemes, specifically establishing its consistency and sample complexity. Further, experiments illuminated the upsides of general utilities for multi-agent navigation problems amidst obstacles. Future work includes generalizations to continuous spaces, fusions of model-based and model-free approaches, and allowing partial observability.

References

[1] M. L. Puterman, *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[2] R. E. Kalman *et al.*, "Contributions to the theory of optimal control," *Bol. soc. mat. mexicana*, vol. 5, no. 2, pp. 102–119, 1960.

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

[4] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013.

[5] X. Zhao, L. Xia, L. Zhang, Z. Ding, D. Yin, and J. Tang, "Deep reinforcement learning for page-wise recommendations," in *Proceedings of the 12th ACM Conf. on Recommender Systems*, 2018, pp. 95–103.

[6] E. A. Feinberg, "Optimality conditions for inventory control," in *Optimization Challenges in Complex, Networked and Risky Systems*. INFORMS, 2016, pp. 14–45.

[7] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev *et al.*, "Grandmaster level in starcraft ii using multi-agent reinforcement learning," *Nature*, vol. 575, no. 7782, pp. 350–354, 2019.

[8] K. Zhang, Z. Yang, H. Liu, T. Zhang, and T. Basar, "Fully decentralized multi-agent reinforcement learning with networked agents," in *in ICML*, 2018, pp. 5872–5881.

[9] P. Wang, C.-Y. Chan, and A. de La Fortelle, "A reinforcement learning based approach for automated lane change maneuvers," in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 1379–1384.

[10] T. Başar and G. J. Olsder, *Dynamic noncooperative game theory*. SIAM, 1998.

[11] L. Busoniu, R. Babuska, and B. De Schutter, "A comprehensive survey of multiagent reinforcement learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 2, pp. 156–172, 2008.

[12] E. Hazan, S. Kakade, K. Singh, and A. Van Soest, "Provably efficient maximum entropy exploration," in *in ICML*, 2019, pp. 2681–2691.

[13] V. S. Borkar and S. P. Meyn, "Risk-sensitive optimal control for Markov decision processes with monotone cost," *Mathematics of Operations Research*, vol. 27, no. 1, pp. 192–209, 2002.

[14] E. Altman, *Constrained Markov decision processes*. CRC Press, 1999, vol. 7.

[15] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and autonomous systems*, vol. 57, no. 5, pp. 469–483, 2009.

[16] L. C. M. Kallenberg, "Survey of linear programming for standard and nonstandard Markovian control problems. Part I: Theory," *Zeitschrift für Operations Research*, vol. 40, no. 1, pp. 1–42, 1994.

[17] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *in NeurIPS*, 2000, pp. 1057–1063.

[18] J. Zhang, A. Koppel, A. S. Bedi, C. Szepesvari, and M. Wang, "Variational policy gradient method for reinforcement learning with general utilities," *in NeurIPS*, vol. 33, 2020.

[19] J. Zhang, A. S. Bedi, M. Wang, and A. Koppel, "Marl with general utilities via decentralized shadow reward actor-critic," *arXiv preprint arXiv:2106.00543*, 2021.

[20] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.

[21] H. G. Tanner, A. Jadbabaie, and G. J. Pappas, "Flocking in fixed and switching networks," *IEEE Transactions on Automatic control*, vol. 52, no. 5, pp. 863–868, 2007.

[22] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE transactions on information theory*, vol. 52, no. 6, pp. 2508–2530, 2006.

[23] A. Koppel, F. Y. Jakubiec, and A. Ribeiro, "A saddle point algorithm for networked online convex optimization," *IEEE Transactions on Signal Processing*, vol. 63, no. 19, pp. 5149–5164, 2015.

[24] F. R. Chung and F. C. Graham, *Spectral graph theory*. American Mathematical Soc., 1997, no. 92.

[25] C. Nowzari, E. Garcia, and J. Cortés, "Event-triggered communication and control of networked systems for multi-agent consensus," *Automatica*, vol. 105, pp. 1–27, 2019.

[26] T. Chen, K. Zhang, G. B. Giannakis, and T. Başar, "Communication-efficient distributed reinforcement learning," *arXiv preprint arXiv:1812.03239*, 2018.

[27] J. Zhang, A. S. Bedi, M. Wang, and A. Koppel, "Cautious reinforcement learning via distributional risk in the dual domain," *arXiv preprint arXiv:2002.12475*, 2020.

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

[29] H.-T. Wai, Z. Yang, Z. Wang, and M. Hong, "Multi-agent reinforcement learning via double averaging primal-dual optimization," in *in NeurIPS*, 2018, pp. 9649–9660.

[30] C. Qu, S. Mannor, H. Xu, Y. Qi, L. Song, and J. Xiong, "Value propagation for decentralized networked deep multi-agent reinforcement learning," in *in NeurIPS*, 2019, pp. 1184–1193.

[31] T. Doan, S. Maguluri, and J. Romberg, "Finite-time analysis of distributed td (0) with linear function approximation on multi-agent reinforcement learning," in *in ICML*, 2019, pp. 1626–1635.

[32] G. Scutari, F. Facchinei, P. Song, D. P. Palomar, and J.-S. Pang, "Decomposition by partial linearization: Parallel optimization of multi-agent systems," *IEEE TSP*, vol. 62, no. 3, pp. 641–656, 2013.

[33] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine learning*, vol. 8, no. 3-4, pp. 229–256, 1992.

[34] A. Shapiro, D. Dentcheva, and A. Ruszczyński, *Lectures on stochastic programming: modeling and theory*. SIAM, 2014.

[35] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," *Neural Information Processing Systems (NIPS)*, 2017.