Multi-Agent Reinforcement Learning with General Utilities via Decentralized Shadow Reward Actor-Critic

Junyu Zhang,1 Amrit Singh Bedi,2 Mengdi Wang,3 Alec Koppel4
1 Dept. of ISEM, National University of Singapore, Singapore
2 CISD, US Army Research Laboratory, USA
3 Dept. of EE, Center for Statistics and Machine Learning, Princeton University/Deepmind, Princeton, USA
4 Supply Chain Optimization Technologies, Amazon, USA
{junyz@nus.edu.sg, amrit0714@gmail.com, mengdiw@princeton.edu, aekoppel@amazon.com}

Abstract
We posit a new mechanism for cooperation in multi-agent reinforcement learning (MARL) based upon any nonlinear function of the team’s long-term state-action occupancy measure, i.e., a general utility. This subsumes the cumulative return but also allows one to incorporate risk-sensitivity, exploration, and priors. We derive the Decentralized Shadow Reward Actor-Critic (DSAC) in which agents alternate between policy evaluation (critic), weighted averaging with neighbors (information mixing), and local gradient updates for their policy parameters (actor). DSAC augments the classic critic step by requiring agents to (i) estimate their local occupancy measure in order to (ii) estimate the derivative of the local utility with respect to their occupancy measure, i.e., the “shadow reward”. DSAC converges to a stationary point in sublinear rate with high probability, depending on the amount of communications. Under proper conditions, we further establish the non-existence of spurious stationary points for this problem, that is, DSAC finds the globally optimal policy. Experiments demonstrate the merits of goals beyond the cumulative return in cooperative MARL.

Introduction
Reinforcement learning (RL) is a framework for directly estimating the parameters of a controller through repeated interaction with the environment (Sutton and Barto 2018), and has gained attention for its ability to alleviate the need for a physically exact model across a number of domains, such as robotic manipulation (Kober, Bagnell, and Peters 2013), web services (Zhao et al. 2018), and logistics (Feinberg 2016), and various games (Silver et al. 2016). In RL, an agent in a given state takes an action, and transitions to another according to a Markov transition density, whereby a reward informing the merit of the action is revealed by the environment. Mathematically, this setting may be encapsulated by a Markov Decision Process (MDP) (Puterman 2014), in which the one seeks to select actions to maximize the long-term accumulation of rewards.

In many domains, multiple agents interact in order to obtain favorable outcomes, as in finance (Lee, Zhang et al. 2002), social networks (Jaques et al. 2019), and games (Vinyals et al. 2019). In multi-agent RL (MARL) and more generally, stochastic games, a key question is the payoff structure (Shapley 1953; Basar and Olsder 1998). We focus on common payoffs among agents, i.e., the utility of the team is the sum of local utilities (Busoniu, Babuska, and De Schutter 2008), which contrasts with competitive settings where one agent’s gain is another’s loss, or combinations thereof (Littman 1994). Whereas typically cooperative MARL defines the global utility as the average over agents’ local reward accumulations, here we define a new mechanism for cooperation that permits agents to incorporate risk-sensitivity (Huang and Kallenberg 1994; Borkar and Meyn 2002; Prashanth and Ghavamzadeh 2016), prior experience (Argall et al. 2009), or exploration (Hazan et al. 2019; Touboul and Lazaric 2019). The usual common-payoff setting focuses on global cumulative return of rewards, which is a linear function of the the state-action occupancy measure. By contrast, the aforementioned decision-making goals define nonlinear functions of the state-action occupancy measure (Kallenberg 1994). Such functions, which we call general utilities, have recently yielded impressive performance in practice via prioritizing exploration (Mahajan et al. 2019; Gupta et al. 2020), risk-sensitivity (Mystery 2021), and prior experience (Le et al. 2017; Lee and Lee 2019). However, to the best of our knowledge, there exists few formal guarantees for algorithms designed to optimize general utilities in multi-agent settings.

This gap motivates us to put forth the first decentralized MARL scheme for general utilities, and establish its consistency and sample complexity. Our approach hinges upon first noting that the embarking point for most RL methodologies is the Policy Gradient Theorem (Williams 1992; Sutton et al. 2000) or Bellman’s equation, both of which break down for general utilities. One potential path forward is a recent generalization of the PG Theorem for general utilities (Zhang et al. 2020b), which expresses the gradient as product of the partial derivative of the utility with respect to the occupancy measure, and the occupancy measure with respect to the policy. However, in the team setting, this later factor is a global nonlinear function of agents’ policies, and hence does not permit decentralization. Thus, we define an agent’s local occupancy measure as the joint occupancy measure of all agents’ policies with all others’ marginalized out, and its local general utility as any (not-necessarily concave) function of its marginal occupancy measure. The team
objective, then, is the global aggregation of all local utilities.

From this definition, we derive a new variant of the Policy Gradient [cf. (6)] where each agent estimate its policy gradient based on local information and message passing with neighbors. This leads to a model-free algorithm, Decentralized Shadow Reward Actor-Critic (DSAC), that generalizes multi-agent actor-critic (see (Konda and Borkar 1999; Konda and Tsitsiklis 2000)) beyond cumulative reward as a special case (Zhang et al. 2020a,b): 

\[ R(s, a) = \sum_{t=0}^{\infty} \gamma^t \cdot P(s_t = s, a_t = a | \pi, s^0 \sim \xi) \]  

for \( \forall a \in A, \forall s \in S \). For instance, often in applications one has access to demonstrations which can be used to learn a prior on the policy for ensuring baseline performance. Suppose \( \lambda \) is a prior state-action distribution obtained from demonstrations. One may seek to maintain baseline performance with respect to this prior via minimizing the Kullback-Liebler (KL) divergence between the normalized distribution \( \lambda = (1 - \gamma)\lambda + \gamma \pi \) and the prior \( \lambda \) stated as \( \rho(\lambda) = KL((1 - \gamma)\lambda | \lambda) \). In behavioral cloning, action information is missing, in which case one may instead consider a variant with respect to only the state occupancy measure. Other forms for (1) are considered in Sec. 12.

In this work, we consider the decentralized version of the problem in (1), where the state space \( S \), the action space \( A \), the policy \( \pi \), and the general utility \( F \) are decentralized among \( |V| \) distinct agents associated with an undirected graph \( G = (V, E) \) with vertex set \( V \) and edge set \( E \). Each agent \( i \in V \) is associated with its own local incentives and actions, detailed as follows.

**Space Decomposition.** The global state space \( S \) is the product of \( |V| \) local spaces \( S_i \), i.e., \( S = S_1 \times S_2 \times \cdots \times S_N \), meaning that for any \( s \in S \), we may write \( s = (s_{(1)}, s_{(2)}, \cdots, s_{(N)}) \) with \( s_{(i)} \in S_i, i \in V \). Each agent has access to the global state \( s \), as customary of joint-action learners training in a decentralized manner under full observability (Kar, Moura, and Poor 2013; Zhang et al. 2018; Lee et al. 2018; Wai et al. 2018; Qu et al. 2019; Doan, Maguluri, and Romberg 2019). Similarly, the global action space \( A \) is the product of \( |V| \) local spaces \( A_i \): \( A = A_1 \times A_2 \times \cdots \times A_N \), meaning that for any \( a \in A \), we may write \( a = (a_{(1)}, a_{(2)}, \cdots, a_{(N)}) \) with \( a_{(i)} \in A_i, i \in V \). Full observability means each agent \( i \) has access to global actions \( a \) concatenating all local ones.

**Policy Factorization.** The global policy \( \pi(a|s) \) that maps global action \( a \) for a given global state \( s \) is defined as the product of local policies \( \prod_{i=1}^{N} \pi_{(i)}(a_{(i)}|s_{(i)}) \), which prescribes statistical independence among agents’ policies. For the parameterized policy \( \pi_{(i)}(a|s) \) where \( \theta \in \Theta \), we denote \( \theta = (\theta_{(1)}, \theta_{(2)}, \cdots, \theta_N) \) as the parameter, so we can write \( \pi_{(i)}(a|s) = \prod_{i \in V} \pi_{(i)}(a_{(i)}|s_{(i)}) \), where the local policy of agent \( i \) is parameterized by \( \theta_i \). Since the global state is visible to all agents, the local policy is based on the observation of the global state. The parameters \( \theta_i \) are kept private by agent \( i \), meaning that agents must pass messages to become informed about others’ incentives.

**Local Cumulative State-Action Occupancy Measure.** Similar to the global occupancy measure \( \lambda(\pi) \) [cf. (2)], we define the local cumulative state-action occupancy measure:

\[ \lambda_{(i)}(s_{(i)}, a_{(i)}) = \sum_{t=0}^{\infty} \gamma_t \cdot P(s_t = s, a_t = a | \pi, s^0 \sim \xi) \]  

for \( \forall a_{(i)} \in A_i, s_{(i)} \in S_i \). This local occupancy measure is the marginalization of the global occupancy measure with respect to all others’ measures than agent \( i \), whose indices are denoted as \( \{\bar{i}\} \subset V \). Via marginalization, we write

\[ \lambda(\pi) = \sum_{a \in \{a_{(i)}\} \times A_{\bar{i}}} \sum_{s \in \{s_{(i)}\} \times S_{\bar{i}}} \lambda(\pi) \]

**Problem Formulation**

Consider a Markov decision process (MDP) over the finite state space \( S \) and a finite action space \( A \). For each state \( s \in S \), a transition to state \( s' \in S \) occurs when selecting action \( a \in A \) according to a conditional probability distribution \( P_a(s, s') \), for which we define the short-hand notation \( P_a(s, s') \). Let \( \xi \) be the initial state distribution of the MDP, i.e., \( s_0 \sim \xi \). We let \( |S| \) denote the number of states and \( |A| \) the number of actions. Consider policy optimization for maximizing general objectives that are nonlinear function of the cumulative discounted state-action occupancy measure under policy \( \pi \), which contains the cumulative return as a special case (Zhang et al. 2020a,b):

\[ \max_{\pi} R(\pi) := F(\lambda(\pi)) \]  

(1)

where \( F \) is a general (not necessarily concave) functional and \( \lambda(\pi) \) is occupancy measure given by

\[ \lambda(\pi) = \sum_{t=0}^{\infty} \gamma_t \cdot P(s_t = s, a_t = a | \pi, s^0 \sim \xi) \]  

(2)

for \( \forall a \in A, \forall s \in S \). For instance, often in applications one has access to demonstrations which can be used to learn a prior on the policy for ensuring baseline performance. Suppose \( \lambda \) is a prior state-action distribution obtained from demonstrations. One may seek to maintain baseline performance with respect to this prior via minimizing the Kullback-Liebler (KL) divergence between the normalized distribution \( \lambda = (1 - \gamma)\lambda + \gamma \pi \) and the prior \( \lambda \) stated as \( \rho(\lambda) = KL((1 - \gamma)\lambda | \lambda) \). In behavioral cloning, action information is missing, in which case one may instead consider a variant with respect to only the state occupancy measure. Other forms for (1) are considered in Sec. 12.

In this work, we consider the decentralized version of the problem in (1), where the state space \( S \), the action space \( A \), the policy \( \pi \), and the general utility \( F \) are decentralized among \( |V| \) distinct agents associated with an undirected graph \( G = (V, E) \) with vertex set \( V \) and edge set \( E \). Each agent \( i \in V \) is associated with its own local incentives and actions, detailed as follows.

**Space Decomposition.** The global state space \( S \) is the product of \( |V| \) local spaces \( S_i \), i.e., \( S = S_1 \times S_2 \times \cdots \times S_N \), meaning that for any \( s \in S \), we may write \( s = (s_{(1)}, s_{(2)}, \cdots, s_{(N)}) \) with \( s_{(i)} \in S_i, i \in V \). Each agent has access to the global state \( s \), as customary of joint-action learners training in a decentralized manner under full observability (Kar, Moura, and Poor 2013; Zhang et al. 2018; Lee et al. 2018; Wai et al. 2018; Qu et al. 2019; Doan, Maguluri, and Romberg 2019). Similarly, the global action space \( A \) is the product of \( |V| \) local spaces \( A_i \): \( A = A_1 \times A_2 \times \cdots \times A_N \), meaning that for any \( a \in A \), we may write \( a = (a_{(1)}, a_{(2)}, \cdots, a_{(N)}) \) with \( a_{(i)} \in A_i, i \in V \). Full observability means each agent \( i \) has access to global actions \( a \) concatenating all local ones.

**Policy Factorization.** The global policy \( \pi(a|s) \) that maps global action \( a \) for a given global state \( s \) is defined as the product of local policies \( \prod_{i=1}^{N} \pi_{(i)}(a_{(i)}|s_{(i)}) \), which prescribes statistical independence among agents’ policies. For the parameterized policy \( \pi_{(i)}(a|s) \) where \( \theta \in \Theta \), we denote \( \theta = (\theta_{(1)}, \theta_{(2)}, \cdots, \theta_N) \) as the parameter, so we can write \( \pi_{(i)}(a|s) = \prod_{i \in V} \pi_{(i)}(a_{(i)}|s_{(i)}) \), where the local policy of agent \( i \) is parameterized by \( \theta_i \). Since the global state is visible to all agents, the local policy is based on the observation of the global state. The parameters \( \theta_i \) are kept private by agent \( i \), meaning that agents must pass messages to become informed about others’ incentives.

**Local Cumulative State-Action Occupancy Measure.** Similar to the global occupancy measure \( \lambda(\pi) \) [cf. (2)], we define the local cumulative state-action occupancy measure:

\[ \lambda_{(i)}(s_{(i)}, a_{(i)}) = \sum_{t=0}^{\infty} \gamma_t \cdot P(s_t = s, a_t = a | \pi, s^0 \sim \xi) \]  

(3)

for \( \forall a_{(i)} \in A_i, s_{(i)} \in S_i \). This local occupancy measure is the marginalization of the global occupancy measure with respect to all others’ measures than agent \( i \), whose indices are denoted as \( \{\bar{i}\} \subset V \). Via marginalization, we write

\[ \lambda(\pi) = \sum_{a \in \{a_{(i)}\} \times A_{\bar{i}}} \sum_{s \in \{s_{(i)}\} \times S_{\bar{i}}} \lambda(\pi) \]  

(4)
with $A_{i} = \Pi_{j \neq i} A_{j}$ and $S_{i} = \Pi_{j \neq i} S_{j}$. Note that (4) is a linear transform of $\lambda^\pi$ in (2).

**Local Utility.** Let $S_{i} = |S_{i}|$ denote the number of local states and $A_{i} := |A_{i}|$ the number of local actions. For agent $i$, define the local utility function $F_{i}(\cdot) : \mathbb{R}^{S_{i}A_{i}} \rightarrow \mathbb{R}$ as a function of $\lambda_{i}^{\pi}(i)$ depends on $\theta_{i}$ when agent $i$ follows policy $\pi_{\theta_{i}}$. Then, define the global utility as the sum of local ones:

$$R(\pi_{\theta}) = F(\lambda^{\pi_{\theta}}) := \frac{1}{N} \sum_{i=1}^{N} F_{i}(\lambda_{i}^{\pi_{\theta}}). \quad (5)$$

Note that (5) is not node-separable, and local occupancy measures depend on the global one through (4). This means that the policy parameters $\theta_{i}$ of agent $i$ depends on global policy $\pi$, and hence on global parameter $\theta = (\theta_{1}, \theta_{2}, \ldots, \theta_{N})$. This is a key point of departure from standard multi-agent optimization (Nedic and Ozdaglar 2009). Next we shift to deriving a variant of actor-critic that is tuned to the multi-agent setting with general utilities (5).

**Elements of MARL with General Utilities**

This section develops an actor-critic type algorithm for MARL with general utilities (5). One challenge is that the occupancy measure, the policy parameters, and the utility are coupled. Specifically, the value function is not additive across trajectories, and hence invalidates RL approaches tailored to maximizing cumulative returns based upon either the Policy Gradient Theorem (Williams 1992; Sutton et al. 2000) or Bellman’s equation (Puterman 2014). To address this issue, we employ a combination of the chain rule, an additional density estimation step, and the construction of a “shadow reward.” We first define the shadow reward and value function as follows and then will proceed towards the proposed algorithm.

**Shadow Rewards and Policy Evaluation**

The general utility objective cannot be written as cumulative sum of returns. The nonlinearity invalidates the additivity, which is the origination of the definition of the conventional reward function and Q function, quantities that are central to approaches for maximizing cumulative-returns, via either dynamic programming (Puterman 2014) or policy search (Williams 1992; Sutton et al. 2000). To circumvent the need for additivity, we will introduce auxiliary variables, which we call shadow rewards and shadow Q functions.

**Definition 1 (Shadow Reward and Shadow Q Function).** The shadow reward $r^{\pi} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ of policy $\pi$ w.r.t. general utility $F$ is $r^{\pi}(s,a) := \frac{\partial F(\lambda^{\pi_{\theta}})}{\partial \lambda_{\pi}(s,a)}$, with associated shadow Q function $Q_{\pi}^{\theta}(s,a) := \mathbb{E}\{\sum_{t=0}^{\infty} \gamma^{t} \cdot r^{\pi}(s^{t},a^{t})|s^{0} = s, a^{0} = a, \pi\}$. 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.

**Policy Evaluation Criterion.** We shift to how one may compute the Shadow Q-function from trajectory information, upon the basis of which we can estimate the parameters of a critic. To do so, we use function approximation to parameterize the high-dimensional shadow Q-function. One simple choice is linear function approximation. That is, given 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

$$Q_{w}(s,a) := \langle \phi(s,a), w \rangle \quad \forall (s,a) \in \mathcal{S} \times \mathcal{A}. \quad (8)$$

In our algorithm, we will update a sequence of $w$ to closely approximate the sequence of implicit shadow Q functions, as policy gets updated. In practice, the parametrization (8) needs not be linear. Indeed, experimentally, we consider $Q$ defined by a multi-layer neural network.

Thus, the critic objective of policy $\pi$ is defined as the mean-square-error w.r.t. shadow Q-function:

$$\ell(w; \pi) := \mathbb{E}\left[\sum_{t=0}^{\infty} \frac{1}{2}(Q_{w}(s^{t},a^{t}) - Q_{\pi}^{\theta}(s^{t},a^{t}))^{2}|s^{0} \sim \xi, \pi\right]$$

$$= \frac{1}{2} \sum_{s,a} \lambda^{\pi}(s,a) \langle \phi(s,a), w - Q_{\pi}^{\theta}(s,a) \rangle^{2} \quad (9)$$

<table>
<thead>
<tr>
<th>Objective</th>
<th>Approach</th>
<th>Convergence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cumulative Return</td>
<td>Value-Based [In caption]</td>
<td>✓</td>
</tr>
<tr>
<td>Risk</td>
<td>Policy-Based (Zhang et al. 2018; Chen et al. 2018)</td>
<td>✓</td>
</tr>
<tr>
<td>Exploration</td>
<td>(Mahajan et al. 2019; Gupta et al. 2020)</td>
<td>✓</td>
</tr>
<tr>
<td>Priors</td>
<td>(Le et al. 2017; Lee and Lee 2019)</td>
<td>✓</td>
</tr>
</tbody>
</table>
Algorithm 1: Decentralized Shadow Reward Actor-Critic (DSAC)

1. **Input**: initial policy \(\theta^0\); actor step-sizes \(\{\eta_k^a\}\); Batch sizes \(\{B_k\}\); Episode lengths \(\{H_k\}\); initial critic \(W^0 := [w^0_0, w^0_1, \ldots, w^0_N] \in \mathbb{R}^d\) with \(w^0_i = w^0\), for all \(i, j\); critic step-size \(\{\eta_k^c\}\); mixing matrix \(M \in \mathbb{R}^{N \times N}\); mixing round \(m \geq 1\).

2. **for** iteration \(k = 0, 1, 2, \ldots\) **do**
   3. Perform \(B_k\) Monte Carlo rollouts to obtain trajectories \(T = \{s^0, a^0, \ldots, s^H_k, a^H_k\}\) with initial dist. \(\xi\), policy \(\pi^k\) collected as batch \(B_k\).
   4. **for** agent \(i = 1, 2, \ldots, N\) **do**
      5. Compute empirical local occupancy measure
         \[
         \hat{\lambda}^k_i = \frac{1}{B_k} \sum_{T \in B_k} \sum_{t=0}^{H_k} \gamma^t \cdot \mathbf{e}(s^t_i, a^t_i),
         \]
         estimate shadow reward \(\hat{r}^k_i = \nabla_{\lambda_i} F_i(\hat{\lambda}^k_i)\).
   6. **for** agent \(i = 1, 2, \ldots, N\) **do**
      7. With localized policy gradient estimate
         \[
         G_{w_i}(T_i, \pi^k_i, w^k_i) = \frac{1}{B_k} \sum_{T \in B_k} \sum_{t=0}^{H_k} \gamma^t \cdot (Q_{w_i}(s^t_i, a^t_i) - \hat{Q}^k_i) \cdot \nabla_{w_i} Q_{w_i}(s^t_i, a^t_i),
         \]
         compute
         \[
         \hat{\Delta}^k_i = \frac{1}{B_k} \sum_{T \in B_k} G_{w_i}(T_i, \pi^k_i, w^k_i), \quad w^{k+1}_i = w^k_i - \eta_k^a \hat{\Delta}^k_i.
         \]
   8. **for** iter = 1, \ldots, \(m\) **do**
      9. **for** agent \(i = 1, 2, \ldots, N\) **do**
         10. Exchange information with neighbours:
             \[
             w^{k+1}_i = \sum_{(j, j_i) \in E} M(j, i) \cdot w^{k+1}_i.
             \]
      11. **for** agent \(i = 1, 2, \ldots, N\) **do**
         12. With \(G_{\theta_i}(T_i, w^k_i) = \sum_{t=0}^{H^i} \gamma^t \cdot Q_{w_i}(s^t_i, a^t_i) \nabla_{\theta_i} \log \pi^k_i(a^t_i|s^t_i)\), update the policy:
             \[
             \Delta_{\theta_i} := \frac{1}{B_k} \sum_{T \in B_k} G_{\theta_i}(T, w^{k+1}_i), \quad \theta^{k+1}_i = \theta^k_i + \eta_k^c \Delta_{\theta_i}.
             \]

Via the definition of the occupancy measure \(\lambda^\pi\) [cf. (2)], the expectation may be substituted by weighting factors in the summand on the second line. We assume features \(\{\phi(s, a)\}_{s \in \mathcal{S}, a \in \mathcal{A}}\) are bounded (see Zhang et al. (2021) for details). With the shadow reward and associated Q-function (Definition 1), the policy evaluation criterion (9), and its smoothness properties with respect to critic parameters \(w\) in place, we expand on their role in the multi-agent setting.

Multi-Agent Optimization for Critic Estimation

Setting aside the issue of policy parameter updates for now, we focus on estimating the global general utility. The shadow Q function and shadow reward (Definition 1) depend on global knowledge of all local utilities, which are unavailable as local incentives are local only. To mitigate this issue, we introduce their localized components, which together comprise the global shadow Q function and reward. Specifically, define the local shadow reward \(r^\pi_i\) for agent \(i\):

\[
r^\pi_i(s(i), a(i)) := \frac{\partial F_i(\lambda^\pi_i)}{\partial \lambda_i(s(i), a(i))), \forall(s(i), a(i)) \in \mathcal{S}_i \times \mathcal{A}_i.}
\]

Clearly, it holds that \(r^\pi(s, a) = \frac{1}{N} \sum_{i=1}^{N} r^\pi_i(s(i), a(i))\).

Based on the local observation of the its own shadow reward, agent \(i\) may access its local shadow Q function \(Q^\pi_i : S \times \mathcal{A} \rightarrow \mathbb{R}\):

\[
Q^\pi_i(s, a) := \mathbb{E} \left[ \sum_{t=0}^{\infty} \gamma^t \cdot r^\pi_i(s(t), a(t)) \mid s^0 = s, a^0 = a, \pi \right],
\]

for \(\forall(s, a) \in \mathcal{S} \times \mathcal{A}\). Therefore, we also have \(Q^\pi_F(s, a) = \frac{1}{N} \sum_{i=1}^{N} Q^\pi_i(s, a)\). 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 (9). By exploiting the aforementioned node-separability and introducing a localized critic parameter vector \(w_i\) associated to agent \(i\), this may equivalently be expressed as a consensus optimization problem (Nedic and Ozdaglar 2009):

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

where the local policy evaluation criterion is defined as \(f_i(w_i; \pi) := \mathbb{E} \left[ \sum_{t=0}^{\infty} \gamma^t \cdot (Q_{w_i}(s^t, a^t) - Q^\pi_F(s^t, a^t)) \right] \mathbb{E} \mid s^0 \sim \xi, \pi \).

This formulation allows agent \(i\) to evaluate its policy with respect to global utility (5) through the local criterion \(f_i(w_i; \pi)\) as a surrogate for that which aggregates global information (9), when consensus over local parameters \(w_i\) is imposed. Next, we incorporate solutions to (12) into the critic step together with a policy parameter \(\theta_i\) update along stochastic ascent directions via (6) for the actor to assemble DSAC.

Decentralized Shadow Reward Actor-Critic

Next, we put together these pieces to present Decentralized Shadow Reward Actor-Critic (DSAC) as Algorithm 1. This scheme allows agents to keep their local utilities \(F_i\), and policies \(\pi_{\theta_i}\) with associated parameters \(\theta_i\) 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 \(\mathcal{E}\) and mixing matrix \(M\) 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. Each step is detailed in Algorithm 1.

Consistency and Sample Complexity

In this section, we study the finite sample performance of Algorithm 1. We show \(\hat{O}(\varepsilon^{-2.5})\) (Theorem 2) or \(\hat{O}(\varepsilon^{-2})\)
(Corollary 2) sample complexities to obtain $\epsilon$-stationary points of global utility, depending on the number of communications per step, akin to best known rates for non-concave expected maximization problems (Shapiro, Dentcheva, and Ruszczyński 2014). We also establish the nonexistence of spurious extrema for this setting, indicating the convergence to global optimality (Corollary 1). Before continuing, we present a few key technical conditions for the utility $F$, the policy $\pi_\theta$, the mixing matrix $M$, and the critic approximation. The other assumptions are stated in Appendix B.2 of the supplementary material (see Zhang et al. 2021) for details.

**Assumption 1.** For utility $F$ [cf. (5)], we assume for $\forall i$ that:

(i). $F_i(\cdot)$ is private to agent $i$.

(ii). $\exists C_F > 0$ s.t. $\|\nabla_{\lambda_i(\cdot)} F_i(\lambda_i(\cdot))\|_\infty \leq C_F$ in a neighbourhood of the occupancy measure set.

(iii). $\exists L_\lambda > 0$ s.t. $\|\nabla_{\lambda_i(\cdot)} F_i(\lambda_i(\cdot)) - \nabla_{\lambda_i(\cdot)} F_i(\lambda_i(\cdot))\|_\infty \leq L_\lambda \|\lambda_i - \lambda_i(\cdot)\|_\infty$.

(iv). $\exists L_\theta > 0$ s.t. $F \circ \lambda(\cdot)$ is $L_\theta$-smooth.

**Assumption 2.** For $\pi_\theta$ and the occupancy measure $\lambda^{\pi_\theta}$, we assume:

(i). The local policy $\pi^{(i)}_\theta$ is private to each agent $i$.

(ii). $\exists C_\pi > 0$ s.t. for each agent $i$, its score function is bounded: $\|\nabla_{\theta_i} \log \pi^{(i)}_\theta((a_i | s))\|_\infty \leq C_\pi$, for $\forall \theta$ and $\forall (s,a)$.

(iii). $\exists \ell_\theta > 0$ s.t. $\|\lambda^{\pi_\theta} - \lambda^{\pi_{\theta'}}\|_\infty \leq \ell_\theta \|\theta - \theta'\|$.

**Assumption 3.** The mixing matrix $M$ is a doubly stochastic matrix satisfying:

(i). $M \in \mathbb{S}_+^{N \times N}$, $M(i,j) > 0$ iff. $(i,j) \in \mathcal{E}$.

(ii). $M \cdot 1_N = 1_N$, where $1_N \in \mathbb{R}^N$ is an all-ones vector.

(iii). Let the eigenvalues of $M$ be $1 = \sigma_1(M) > \sigma_2(M) \geq \cdots \geq \sigma_N(M)$. We define $\rho := \max\{|\sigma_2(M)|, |\sigma_N(M)|\} < 1$.

**Assumption 4.** For $\forall \theta$, define the optimal critic parameter $w^*(\theta) := \arg\min_{w} \frac{1}{N} \sum_{i=1}^N f_i(w; \pi_\theta)$. We assume that:

- $\exists W > 0$ s.t. $E_{\theta} = \sum_{i=1}^N \|\nabla_{\theta_i} F(\lambda^{\pi_\theta}) - \Delta_{\theta_i}\|^2 \leq W$, for $\forall \theta$, where $\Delta_{\theta_i} := \mathbb{E} \left[ \sum_{t=0}^{\infty} \gamma^t Q_{w^*(\theta)}(s^t, a^t) \nabla_{\theta_i} \log \pi^{(i)}_\theta(a^t | s^t) \right] s^0 \sim \xi, \pi_\theta$ is the PG estimate under $w^*(\theta)$. $w^*$.

Assumption 1 requires the boundedness and Lipschitz continuity of the gradient of the utility function. Assumption 2 ensures that the score function is bounded, and the occupancy measure is Lipschitz w.r.t. the policy parameters. These conditions are common to RL algorithms focusing on occupancy measures in recent years (Hanzan et al. 2019; Zhang et al. 2020b), and are automatically satisfied by common policies such as the softmax. Assumption 3 holds for any undirected connected loop-free static graph (Chung and Graham 1997). Assumption 4 states that the feature mis-specification error is uniformly upper bounded by $W$.

Next, we present a brief proof sketch with details provided in the appendices.

**Step 1.** We begin by a standard stochastic gradient ascent analysis (Lemma ??), which yields:

$$F(\lambda^{\pi_{\theta_k+1}}) - F(\lambda^{\pi_{\theta_k}}) \geq \frac{\gamma^k}{4} \|\nabla_{\theta} F(\lambda^{\pi_{\theta_k}})\|^2 - \frac{3\gamma^k}{4} \sum_{i=1}^N \|\nabla_{\theta_i} F(\lambda^{\pi_{\theta_k}}) - \Delta_{\theta_i}\|^2.$$  

**Step 2.** We provide high probability bounds for gradient estimation errors:

$$\sum_{i=1}^N \|\nabla_{\theta_i} f_i(w^*_k; \pi_\theta) - \Delta_{\theta_i}\|^2 \leq \mathcal{O}(B_k^{-1})$$  

and

$$\sum_{i=1}^N \|\nabla_{\theta_i} F(\lambda^{\pi_{\theta_k}}) - \Delta_{\theta_i}\|^2 \leq \mathcal{O}(B_k^{-1} + \sum_{i=1}^N \|w^{k+1}_i - w^{k+1}_0\|^2),$$  

where $w^{k+1}_i := \arg\min_{w} \frac{1}{N} \sum_{i=1}^N f_i(w; \pi_\theta)$ is the ideal critic variable at policy $\pi_\theta$ (Lemma ??).

**Step 3.** By analyzing the consensus error of the communication steps and the gradient descent step for the critic update, we bound critic fitting error term $\sum_{i=1}^N \|w^{k+1}_i - w^{k+1}_0\|^2$ with high probability:

$$\|w^{k+1}_i - w^{k+1}_0\|^2 \leq (1 - c) \|\bar{w}^k - w^{k+1}_0\|^2 + \mathcal{O}(B_k^{-1})$$  

$$+ \mathcal{O} \left( \sum_{i=1}^N \|w^{k+1}_i - \bar{w}^k\|^2 \right) + \text{other controllable noise},$$  

where $c \in (0, 1)$ is some constant, $\bar{w}^k := \sum_{i=1}^N w^{k+1}_i / N$ is the average critic variable in the $k$-th iteration, and $\sum_{i=1}^N \|w^k - \bar{w}^k\|^2 \leq \mathcal{O}((\rho m \sum_{k=0}^{k-1} \eta_k^2 m^2 (k-\delta)^2))$. See Lemmas ?? - ??.

**Step 4.** Next, we construct the following potential function with a carefully selected constant $\alpha$ as to enable the convergence analysis:

$$R_k := F(\lambda^{\pi_{\theta_k}}) - \alpha \|\bar{w}^k - w^k\|^2.$$  

Taking the advantage of the contraction property of $\|\bar{w}^k - w^k\|^2$ and this specific potential function, we characterize algorithm performance in terms of optimization error, the feature mis-specification error, the stochastic PG approximation error, and the multi-agent consensus error. See Lemma ??.

Combining the above steps and suitably specify the parameters, we have the final theorem.

**Theorem 2.** Under Assumption ??, 1, 2, ?? and 3, with one communication round per iteration, i.e. $m = 1$, Algorithm 1 satisfies, under the following parameter selections:

(i) For final iteration $T = \Theta(\epsilon^{-1})$, trajectory lengths $H_k = \mathcal{O}(\log(1/\epsilon) / (1 - \gamma))$, $\delta_k = \delta/\sqrt{3N(T+1)}$, $\delta \in (0, 1)$, batch sizes $B_k = \max\left\{ \frac{1}{\max(4\sqrt{m}, \sqrt{m} \eta_k)}, \frac{1}{4\epsilon} \right\}$, then

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

(ii) For unspecified final iteration $T$, we adaptively set: $\delta_k = \frac{1}{N^{5/2}(k+1)^5}$, $\delta \in (0, 1)$, trajectory lengths $H_k = \mathcal{O}(1 -$
 entropy comparison

state space coverage

\[ \gamma^{-1} \log(k + 1) \], batchsizes \( B_k = \log(1/\delta_k)(k + 1)^{3/2} \), and step-sizes \( \eta_i^k = \min \left\{ \frac{(1-\gamma)(1-B_k)}{\max\{4/3, N, 10\}} \cdot \frac{1}{\lambda_k} \right\} \),

\[ \eta_i^k = \min \left\{ (k + 1)^{-1} \cdot L_w^{-1} \right\} \], then

\[ \sum_{k=1}^{T} \eta_i^k \| \nabla_{\theta_i} F(\lambda_i^{x_k}) \|^2 \leq O \left( \frac{\log T}{T^{3/4}} + W \right), \quad \text{w.p. } 1 - \delta \]

In either case, Algorithm 1 requires \( O(\epsilon^{2.5}) \) samples to satisfy

\[ \sum_{k=1}^{T} \eta_i^k \| \nabla_{\theta_i} F(\lambda_i^{x_k}) \|^2 \leq O(\epsilon + W). \]

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

**Corollary 1** (Convergence to global optimality). Suppose \( F \) is concave, and the shadow \( Q \) function \( Q_F \) is realizable, i.e., \( W = 0 \) in Assumption 4. For \( \pi_0 \) satisfying Assumption 1 of (Zhang et al. 2020b). For this case, we consider the continuous MountainCar environment of OpenAI Gym (Brockman et al. 2016).

**Experimental Results**

We experimentally investigate the merit of Algorithm 1 in the context of both single and multi-agent problems. The single-node case \( (N = 1) \) bears investigation as the proposed scheme is a new way to solve RL problems with general utilities relative to (Zhang et al. 2020b). For this case, we consider the continuous MountainCar environment of OpenAI Gym (Brockman et al. 2016).

**Concept of Shadow Reward**

To understand the concept of shadow reward, we experiment with the single-agent setup. We consider the exploration maximization problem for the MountainCar environment in which the two dimensional continuous state space is divided into \([12, 11]\) grid size. We run the proposed algorithm for 40 epochs and then plot the count based occupancy measure estimate in the first row of Fig. 1(a). In the figure, light color denotes lower value and dark color represent the higher values as shown in the colorbar. We see that as we go from epoch 1 to epoch 39, the algorithm yields occupancy measures that better cover the state space, which is achieved by the special structure of the “shadow reward” we define as a by-product of the general utility.

**Multi-Agent Experiments**

For multi-agent problems, we experiment with \( N \geq 2 \) agents moving in a two-dimensional continuous space associated with the problem of Cooperative navigation (Lowe et al. 2017).

**Exploration Maximization.** We consider a variant of the cooperative navigation multi-agent environment provided in (Lowe et al. 2017) for \( N = 2 \) agents. The goal of maximum entropy exploration in the multi-agent setting is one in which all agents in the network seek to cover the unknown space, whereby their local utility is the entropy in (5) is given by

\[ F_i(\lambda_i^{s(i)})) = -\sum_{s(i)} \lambda_i^{s(i)} \log(\lambda_i^{s(i)}). \]
Figure 2: (a) Two agent safe navigation with green safe and brown unsafe state space. The goal is to reach G1 and G2 safely from the starting positions A1 and A2, respectively. (b) Undiscounted average reward comparison and (c) average constraint violation comparison for different values of penalty parameter $z$. Observe that imposing constraints allows agents to avoid collision and the unsafe region, while effectively reaching the goals more often.

Figure 3: Safe navigation in a multi-agent cooperative environment with 4 agents and 4 landmarks. Note that the state space in this case would be 16 dimensional (location of agent and landmarks). We run this experiment for three different communication graphs among agents; 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 using Erdős-Rényi random graph model). (a) Running average of the reward return, (b) running average of the constraint violation, and (c) running average of the consensus error for agent 1 and agent 4 for ring and random network connectivity.

We compare DSAC against its corresponding centralized implementations (Cen-AC) or a variant that uses Monte-Carlo rollouts (Cen-R, Dec-R), as well as existing Max-Ent (Hazan et al. 2019) in Fig.1(b)-1(c). Observe that Max-Ent does not achieve comparable performance, and DSAC achieves comparable performance to its variants that require centralization. Fig. 1(c) visualizes the heatmap of the marginalized measure at agent 1 for DSAC (red) at different epochs as compared to MaxEnt (purple) and random baseline (green) – note the superior space coverage of DSAC (red).

Safe Cooperative Navigation. We consider a two agent cooperative environment from (Lowe et al. 2017) where each agent needs to reach its assigned goal while traversing only through the safe region as visualized in Fig.2(a). Agents receive a negative reward proportional to its distance from the landmark, and an additional negative reward of $-1$ if agents collide. Additionally, each agent receives a high cost of $c = 1$ if it passes through the unsafe region (middle of the state space) – see Fig. 2(a). We impose safety via the constraint for each agent $(\lambda^+ \pi_i, c) \leq C$ where $\lambda^+ \pi_i$ in the marginalized occupancy measure, and including the constraint as a quadratic penalty in a manner similar to (??) (see (Zhang et al. 2021) for further details). To solve this problem, we compare the performance of DSAC for various values of its penalty parameter $z$ to its centralized variant, and a version of multi-agent actor-critic that only ignores the cost. Results for the average reward and constraint violation, respectively, are given in Fig. 2(b)-2(c). The decentralized DSAC achieves comparable performance to its centralized variant, and outperforms existing alternatives, yielding effective learned behaviors for navigation in team settings. Demonstrations for larger networks with different connectivities are in Figure 3.

1 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 motivates defining “shadow rewards” and DSAC, whose critic employs shadow value functions and weighted averaging. Its consistency and sample complexity was rigorously established. Further, experiments illuminated the upsides of general utilities for teams. Future work includes improving communications and sample efficiencies, connections to meta-learning, and allowing information asymmetry.
References


Başar, T.; and Olsder, G. J. 1998. Dynamic noncooperative game theory. SIAM.


Claus, C.; and Bottilier, C. 1998. The dynamics of reinforcement learning in cooperative multiagent systems.


Qu, G.; Wierman, A.; and Li, N. 2020. Scalable reinforcement learning of localized policies for multi-agent networked systems. In Learning for Dynamics and Control, 256–266. PMLR.


Shapiro, A.; Dentcheva, D.; and Ruszczyński, A. 2014. Lectures on stochastic programming: modeling and theory. SIAM.