D3C: Reducing the Price of Anarchy in Multi-Agent Learning

Ian Gemp
DeepMind
London, United Kingdom
imgemp@deepmind.com

Kevin R. McKee
DeepMind
London, United Kingdom
kevinrmckee@deepmind.com

Richard Everett
DeepMind
London, United Kingdom
reverett@deepmind.com

Edgar Duéñez-Guzmán
DeepMind
London, United Kingdom
duenez@deepmind.com

Yoram Bachrach
DeepMind
London, United Kingdom
yorambac@deepmind.com

David Balduzzi
XTX Markets
London, United Kingdom
dbalduzzi@gmail.com

Andrea Tacchetti
DeepMind
London, United Kingdom
atacchet@deepmind.com

ABSTRACT

In multiagent systems, the complex interaction of fixed incentives can lead agents to outcomes that are poor (*inefficient*) not only for the group, but also for each individual. Price of anarchy is a technical, game-theoretic definition that quantifies the inefficiency arising in these scenarios—it compares the welfare that can be achieved through perfect coordination against that achieved by self-interested agents at a Nash equilibrium. We derive a differentiable, upper bound on a price of anarchy that agents can cheaply estimate during learning. Equipped with this estimator, agents can adjust their incentives in a way that improves the efficiency incurred at a Nash equilibrium. Agents do so by learning to mix their reward (equiv. negative loss) with that of other agents by following the gradient of our derived upper bound. We refer to this approach as D3C. In the case where agent incentives are differentiable, D3C resembles the celebrated Win-Stay, Lose-Shift strategy from behavioral game theory, thereby establishing a connection between the global goal of maximum welfare and an established agent-centric learning rule. In the non-differentiable setting, as is common in multiagent reinforcement learning, we show the upper bound can be reduced via evolutionary strategies, until a compromise is reached in a distributed fashion. We demonstrate that D3C improves outcomes for each agent and the group as a whole on several social dilemmas including a traffic network exhibiting Braess's paradox, a prisoner's dilemma, and several multiagent domains.

KEYWORDS

Price of Anarchy; Nash; Reward Sharing; Win-Stay Lose-Shift; Collective Intelligence; Multiagent Reinforcement Learning

ACM Reference Format:
Ian Gemp, Kevin R. McKee, Richard Everett, Edgar Duéñez-Guzmán, Yoram Bachrach, David Balduzzi, and Andrea Tacchetti. 2022. D3C: Reducing the Price of Anarchy in Multi-Agent Learning. In *Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), Online, May 9–13, 2022*, IFAAMAS, 9 pages.

1 INTRODUCTION

We consider a setting consisting of many interacting artificially intelligent agents, each with specific individual incentives. It is well known that the interactions between individual agent goals can lead to inefficiencies at the group level, for example, in environments exhibiting social dilemmas [6, 15, 21]. In order to resolve these fundamental inefficiencies, agents must reach a compromise.

Any arbitration mechanism with a central coordinator[1] faces challenges when scaling to large populations. The coordinator's task becomes intractable as it must both query preferences from a larger population and make decisions accounting for the exponential growth of agent interactions. If agents are permitted to modify their incentives over time, the coordinator must collect all this information again, exacerbating the computational burden. In addition, a central coordinator represents a single point of failure for the system whereas successful multiagent systems identified in nature (e.g., market economies, ant colonies, etc.) are often robust to node failures [10]. Therefore, we focus on decentralized approaches.

Design Criteria: The celebrated Myerson-Satterthwaite theorem [3, 14, 27, 38] states that no mechanism can simultaneously achieve optimal efficiency (welfare-maximizing behavior), budget-balance (no taxing agents, burning side-payments, or hallucinating rewards), appeal to rational individuals (individuals want to opt-in to the mechanism), and be incentive compatible (resulting behavior is a Nash equilibrium). While this impossibility result precludes a mechanism that satisfies the above criteria perfectly, it says nothing about a mechanism that satisfies them *approximately*, which is our aim here. In addition, the mechanism should be decentralized, extensible to large populations, and adapt to learning agents with evolving incentives in possibly non-stationary environments.

Design: We formulate compromise as agents mixing their incentives (rewards or losses) with others. In other words, an agent may become incentivized to minimize a mixture of their loss and other agents' losses. We design a decentralized meta-algorithm that allows agents to search over the space of these possible mixtures.

[1] For example, the Vickrey-Clarke-Groves (VCG) mechanism [7].

We model the problem of efficiency using *price of anarchy*. The price of anarchy, $\rho \in [1, \infty)$, is a measure of inefficiency from algorithmic game theory with lower values indicating more efficient games [29]. Forcing agents to minimize a group (average) loss with a single local minimum results in a "game" with $\rho = 1$. Note that any optimal group loss solution is also Pareto-efficient. Computing the price of anarchy of a game is intractable in general. Instead, we derive a differentiable upper bound on the price of anarchy that agents can optimize incrementally over time. Differentiability of the bound makes it easy to pair the proposed mechanism with, for example, deep learning agents that optimize via gradient descent [22, 31]. Budget balance is achieved exactly by placing constraints on the allowable mixtures of losses. We appeal to individual rationality in three ways. One, we initialize all agents to optimize only their own losses. Two, we include penalties for agents that deviate from this state and mix their losses with others. Three, we show empirically on several domains that opting into the proposed mechanism results in better individual outcomes. We also provide specific, albeit narrow, conditions under which agents may achieve a Nash equilibrium, i.e. the mechanism is incentive compatible, and demonstrate the agents achieving a Nash equilibrium under our proposed mechanism in a traffic network problem. Note that budget-balance is the only property we guarantee is satisfied in absolute terms. All other properties are appealed to either indirectly via design choices (e.g., minimizing ρ) or post-hoc analysis.

Our Contribution: We propose a differentiable, local estimator of game inefficiency, as measured by price of anarchy. We then present two instantiations of a single decentralized meta-algorithm, one 1st order (gradient-feedback) and one 0th order (bandit-feedback), that reduce this inefficiency. This meta-algorithm is general and can be applied to any group of individual agent learning algorithms. In contrast to the centralized training, decentralized execution framework popular in multiagent reinforcement learning (MARL), we demonstrate the success of our meta-algorithm in a more challenging online setting (decentralized training, decentralized execution) on a range of games and MARL domains.

This paper focuses on how to enable a group of agents to respond to an unknown environment and minimize overall inefficiency. Agents with distinct losses may find their incentives well aligned to the given task, however, they may instead encounter a *social dilemma* (§3). We also show that our approach leads to sensible behavior in scenarios where agents may need to sacrifice team reward to *save an individual* (Appx. F.6) or need to form parties and *vote* on a new team direction (Appx. F.5). Ideally, one meta-algorithm would allow a multiagent system to perform sufficiently well in all these scenarios. The approach we propose, D3C (§2), represents a holistic effort to design such a meta-algorithm.[2] All appendices can be found in the longer version of this paper [13].

2 DYNAMICALLY CHANGING THE GAME

In our approach, agents may consider slight re-definitions of their original losses, thereby changing the definition of the original game. Critically, this is done in a way that conserves the original sum of losses (budget-balanced) so that the original group loss can still be measured. In this section, we derive our approach to minimizing the price of anarchy in several steps. First we formulate minimizing the price of anarchy via compromise as an optimization problem. Second we specifically consider compromise as the linear mixing of agent incentives. Next, we define a *local* price of anarchy and derive an upper bound that agents can differentiate. Then, we decompose this bound into a set of differentiable objectives, one for each agent. Finally, we develop a gradient estimator to minimize the agent objectives in settings with bandit feedback (e.g., RL) that enables scalable decentralization.

2.1 Notation and Transformed Losses

Let agent i's loss be $f_i(\boldsymbol{x}) : \boldsymbol{x} \in \mathcal{X} \to \mathbb{R}$ where $\boldsymbol{x}$ is the joint strategy of all agents. Let $f_i^A(\boldsymbol{x})$ denote agent i's transformed loss which mixes losses among agents. Let $\boldsymbol{f}(\boldsymbol{x}) = [f_1(\boldsymbol{x}), \dots, f_n(\boldsymbol{x})]^\top$ and $\boldsymbol{f}^A(\boldsymbol{x}) = [f_1^A(\boldsymbol{x}), \dots, f_n^A(\boldsymbol{x})]^\top$ where $n \in \mathbb{Z}$ denotes the number of agents. In general, we require $f_i^A(\boldsymbol{x}) > 0$ and $\sum_i f_i^A(\boldsymbol{x}) = \sum_i f_i(\boldsymbol{x})$ so that total loss is conserved[3]. Under these constraints, the agents will simply explore the space of possible non-negative group loss decompositions. We consider transformations of the form $\boldsymbol{f}^A(\boldsymbol{x}) = A^\top \boldsymbol{f}(\boldsymbol{x})$ (note the tranpose) where each agent i controls row i of A with each row constrained to the simplex, i.e. $A_i \in \Delta^{n-1}$. For example, agent 1's loss is mixed according to the first **column** of A which may not sum to 1, and not the first row, which it controls:

$$f_1^A(\boldsymbol{x}) = \langle \overbrace{[0.9, 0.3, 0.5]}^{[A_{11}, A_{21}, A_{31}]}, [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), f_3(\boldsymbol{x})] \rangle. \tag{1}$$

Lastly, $[a; b] = [a^\top, b^\top]^\top$ signifies row stacking of vectors, and $\mathcal{X}^*$ denotes the set of Nash equilibria.

2.2 Price of Anarchy

Nisan et al. [29] define price of anarchy as the worst value of an equilibrium divided by the best value in the game. Here, value means sum of player losses, best means lowest, and Nash is the chosen equilibrium concept. It is well known that Nash can be arbitrarily bad from both an individual agent and group perspective; Appx. B presents a simple example and demonstrates how opponent shaping [12, 23] is not a balm for these issues. With the above notation, the price of anarchy is defined as

$$\rho_{\mathcal{X}}(\boldsymbol{f}^A) \stackrel{\text{def}}{=} \frac{\max_{\mathcal{X}^*} \sum_i f_i^A(\boldsymbol{x}^*)}{\min_{\mathcal{X}} \sum_i f_i^A(\boldsymbol{x})} \geq 1. \tag{2}$$

Note that computing the price of anarchy precisely requires solving for both the optimal welfare and the worst case Nash equilibrium. We explain how we circumvent this issue in §2.4.

2.3 Compromise as an Optimization Problem

Given a game, we want to minimize the price of anarchy by perturbing the original agent losses:

$$\min_{\substack{\boldsymbol{f}' = \psi_A(\boldsymbol{f}) \\ \mathbf{1}^\top \boldsymbol{f}' = \mathbf{1}^\top \boldsymbol{f}}} \rho_{\mathcal{X}}(\boldsymbol{f}') + \nu \mathcal{D}(\boldsymbol{f}, \boldsymbol{f}') \tag{3}$$

[2] D3C is agnostic to any action or strategy semantics. We are interested in rich environments where high level actions with semantics such as "cooperation" and "defection" are not easily extracted or do not exist.

[3] The strict definition of price of anarchy assumes positive losses. This is relaxed in §2.5 to allow for losses in $\mathbb{R}$.

where $\boldsymbol{f}$ and $\boldsymbol{f'} = \psi_A(\boldsymbol{f})$ denote the vectors of original and perturbed losses respectively, $\psi_A : \mathbb{R}^n \to \mathbb{R}^n$ is parameterized by weights A, ν is a regularization hyperparameter, and $\mathcal{D}$ penalizes deviation of the perturbed losses from the originals or represents constraints through an indicator function. To ensure minimizing the price of anarchy of the perturbed game improves on the original, we incorporate the constraint that the sum of perturbed losses equals the sum of original losses, $\mathbf{1}^\top \boldsymbol{f'} = \mathbf{1}^\top \boldsymbol{f}$. We refer to this approach as ρ-minimization.

Our agents reconstruct their losses using the losses of all other agents as a basis. For simplicity, we consider linear transformations of their loss functions, although the theoretical bounds hereafter are independent of this simplification. We also restrict ourselves to convex combinations so that agents do not learn incentives that are directly adverse to other agents. The problem can now be reformulated. Let $\psi_A(\boldsymbol{f}) = A^\top \boldsymbol{f}$ and $\mathcal{D}(\boldsymbol{f}, \boldsymbol{f'}) = \sum_i \mathcal{D}_{KL}(\boldsymbol{e}_i \mid\mid A_i)$ where $A \in \mathbb{R}^{n \times n}$ is a right stochastic matrix (rows are non-negative and sum to 1), $\boldsymbol{e}_i \in \mathbb{R}^n$ is a unit vector with a 1 at index i, and $\mathcal{D}_{KL}$ denotes the Kullback-Liebler divergence.

2.4 A Local Price of Anarchy

The price of anarchy, $\rho \geq 1$, is defined over the joint strategy space of all players. Computing it is intractable for general games. However, many agents learn via gradient-based training, and so only observe the portion of the strategy space explored by their learning trajectory. Hence, we imbue our agents with the ability to locally estimate the price of anarchy along this trajectory.

Definition 1 (*Local* Price of Anarchy). *Define*

$$\rho_{\boldsymbol{x}}(\boldsymbol{f}^A, \Delta t) = \frac{\max_{\mathcal{X}^*_\tau} \sum_i f_i^A(\boldsymbol{x}^*)}{\min_{\tau \in [0, \Delta t]} \sum_i f_i^A(\boldsymbol{x} - \tau F(\boldsymbol{x}))} \geq 1 \tag{4}$$

where $F(\boldsymbol{x}) = [\nabla_{x_1} f_1^A(\boldsymbol{x}); \ldots; \nabla_{x_n} f_n^A(\boldsymbol{x})]$, Δt is a small step size, f_i^A is assumed positive $\forall\, i$, and $\mathcal{X}_\tau$ denotes the set of equilibria of the game when constrained to the line.

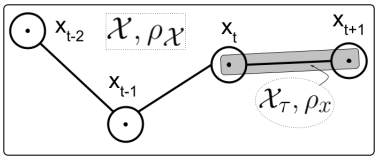


Figure 1: Agents estimate the price of anarchy assuming the joint strategy space, $\mathcal{X}$, of the game is restricted to a local linear region, $\mathcal{X}_\tau$, extending from the currently learned joint strategy, x_t, to the next, x_{t+1}. $\rho_{\mathcal{X}}$ and ρ_x denote the global and local price of anarchy.

To obtain bounds, we leverage theoretical results on *smooth games*, summarized as a class of games where "the externality imposed on any one player by the others is bounded" [36]. We assume a Lipschitz property on all $f_i^A(\boldsymbol{x})$ (details in Theorem 1), which allows us to appeal to this class of games. The bound in equation 7 is tight for some games. Proofs can be found in Appx. D.

For convenience, we repeat the core definition and lemma put forth by Roughgarden [36] here.

Definition 2 (Smooth Game). *A game is (λ, μ)-smooth [36] if:*

$$\sum_{i=1}^n f_i^A(x_i, x'_{-i}) \leq \lambda \sum_{i=1}^n f_i^A(x_i, x_{-i}) + \mu \sum_{i=1}^n f_i^A(x'_i, x'_{-i}) \tag{5}$$

for all $\boldsymbol{x}, \boldsymbol{x'} \in \mathcal{X}$ where $\lambda > 0$, $\mu < 1$. x_{-i} denotes all player $j \neq i$ strategies and $\sum_i f_i^A(\boldsymbol{x})$ is assumed to be non-negative for any $\boldsymbol{x} \in \mathcal{X}$.

The last condition is needed for the price of anarchy, a ratio of welfares, to be meaningful as a positive measure of inefficiency.

Lemma 1 (Smooth Games Imply a Bound on Price of Anarchy). *The price of anarchy is bounded above by a ratio of the coefficients that satisfy the smooth game definition [36]:*

$$1 \leq \rho_{\mathcal{X}}(\boldsymbol{f}^A) \leq \inf_{\lambda > 0, \mu < 1} \left[\frac{\lambda}{1 - \mu} \right]. \tag{6}$$

Theorem 1 (Local *Utilitarian* Price of Anarchy). *Assuming each agent's loss is positive and its loss gradient is Lipschitz, there exists a learning rate $\Delta t > 0$ sufficiently small such that, to $O(\Delta t^2)$, the local **utilitarian** price of anarchy of the game, $\rho_{\boldsymbol{x}}(\boldsymbol{f}^A, \Delta t)$, is upper bounded by*

$$\max_i \{1 + \Delta t\, \mathsf{ReLU}\Big(\frac{d}{dt} \log(f_i^A(\boldsymbol{x})) + \frac{||\nabla_{x_i} f_i^A(\boldsymbol{x})||^2}{\bar{\mu} f_i^A(\boldsymbol{x})} \Big)\} \tag{7}$$

where i indexes each agent, $\bar{\mu} \in \mathbb{R}_{\geq 0}$ is a user-defined upper bound on the true μ, $\mathsf{ReLU}(z) \stackrel{\text{def}}{=} \max(z, 0)$, and Lipschitz implies there exists a β_i such that $||\nabla_{x_i} f_i^A(\boldsymbol{x}) - \nabla_{y_i} f_i^A(\boldsymbol{y})|| \leq \beta_i ||\boldsymbol{x} - \boldsymbol{y}||\ \forall \boldsymbol{x}, \boldsymbol{y}, A$.

Proof Sketch: For a small enough region (grayed in Figure 1), we can approximate each agent's loss function with its Taylor series expansion. By rewriting all losses in the smoothness constraint (equation 5) in terms of expansions about $\boldsymbol{x}$ or $\boldsymbol{x'}$, i.e., quantities we can measure before and after a joint gradient step, we can proceed to define the smoothness constraint with μ and λ in terms of measurable quantities. The smoothness constraint is formulated as a sum over the n agents, but we can decompose this constraint into n individual constraints with their own μ_i's and λ_i's. If each agent can ensure local individual smoothness, which is possible for a small enough region, we show this is sufficient to satisfy the original local smoothness condition with $\mu = \max_i\{\mu_i\}$ and $\lambda = \max_i\{\lambda_i\}$. Each agent can further estimate their own individual price of anarchy, ρ_i, via equation 6 which reduces to a tractable two dimensional constrained optimization problem with a closed form solution. We further show that we can upper bound the local price of anarchy for the group (equation 4) with the max of these individual estimates. Finally, using another expansion along with the log-trick famous from the policy gradient theorem, we recover the final result presented in Theorem 1 below. The Lipschitz assumption exists simply to ensure the series approximations are sufficiently accurate for a small enough region. The full proof is in Appx. D.

Recall that this work focuses on price of anarchy defined using total loss as the value of the game. This is a *utilitarian* objective. We also derive an upper bound on the local *egalitarian* price of anarchy where value is defined as the max loss over all agents (replace $\sum_i$ with $\max_i$ in equation 4; see Appx. D.2), possibly of independent interest.

Theorem 2. *Given n positive losses, $f_i^A(\boldsymbol{x})$, $i \in \{1, \ldots, n\}$, with β_i-Lipschitz gradients there exists a $\Delta t > 0$ sufficiently small such*

*that, to $O(\Delta t^2)$, the local **egalitarian** price of anarchy of the game is upper bounded by*

$$\rho_e \leq 1 + \Delta t\, \mathsf{ReLU}\Big(\frac{d}{dt}\log(\max_i\{f_i^A(\boldsymbol{x})\}) + \frac{\sum_{i=1}^n ||\nabla_{x_i} f_i^A(\boldsymbol{x})||^2}{\bar{\mu}\max_i\{f_i^A(\boldsymbol{x})\}}\Big). \tag{8}$$

2.5 Decentralized Learning of the Loss Mixture Matrix A

Minimizing equation 3 w.r.t. A can become intractable if n is large. Moreover, if solving for A at each step is the responsibility of a central authority, the system is vulnerable to this authority failing. A distributed solution is therefore appealing, and the local price of anarchy bound admits a natural relaxation that decomposes over agents ($\max_i z_i \leq \sum_i z_i$ for $z_i \geq 0$). Equation 3 then factorizes as

$$\min_{A_i \in \Delta^{n-1}} \rho_i + \nu \mathcal{D}_{KL}(\boldsymbol{e}_i \,||\, A_i) \tag{9}$$

where $\rho_i = 1 + \Delta t\, \mathsf{ReLU}\Big(\frac{d}{dt}\log(f_i^A(\boldsymbol{x})) + \frac{||\nabla_{x_i} f_i^A(\boldsymbol{x})||^2}{f_i^A(\boldsymbol{x})\bar{\mu}}\Big)$. Local price of anarchy is subdifferentiable w.r.t. each A_i with gradient

$$\nabla_{A_i}\rho_i \propto \nabla_{A_i}\mathsf{ReLU}\Big(\frac{d}{dt}\log(f_i^A(\boldsymbol{x})) + \frac{||\nabla_{x_i} f_i^A(\boldsymbol{x})||^2}{f_i^A(\boldsymbol{x})\bar{\mu}}\Big). \tag{10}$$

The log appears due to price of anarchy being defined as the worst case Nash total loss *divided* by the minimal total loss. We propose the following modified learning rule for a hypothetical price of anarchy which is defined as a *difference* and accepts negative loss: $A_i \leftarrow A_i - \eta_A \tilde{\nabla}_{A_i}(\rho_i + \nu\mathcal{D}_{KL})$ where η_A is a learning rate and

$$\tilde{\nabla}_{A_i}\rho_i = \nabla_{A_i}\mathsf{ReLU}\Big(\frac{d}{dt} f_i^A(\boldsymbol{x}) + \epsilon\Big). \quad [\epsilon \text{ is a hyperparameter.}] \tag{11}$$

The update direction in (11) is proportional to $\nabla_{A_i}\rho_i$ asymptotically for large f_i^A; see Appx. D.1.1 for further discussion. Each agent i updates x_i and A_i simultaneously using $\nabla_{x_i} f_i^A(\boldsymbol{x})$ and $\tilde{\nabla}_{A_i}(\rho_i + \nu\mathcal{D}_{KL})$.

Improve-Stay, Suffer-Shift—Win-Stay, Lose-Shift (WSLS) [35] is a strategy shown to outperform Tit-for-Tat [33] in an iterated prisoner's dilemma [18, 30]. It was also shown to be psychologically plausible [42] in research on human play. The D3C update direction, $\nabla_{A_i}\rho_i$, encodes the rule: if the loss is decreasing, maintain the mixing weights, otherwise, change them. We can interpret this rule as a generalization of WSLS to learning (derivatives) rather than outcomes (losses). Therefore, we have shown that a sensible, agent-centric learning rule (WSLS) can be derived from minimization of the global, game theoretic concept *price of anarchy* by simply a) restricting agents' strategy spaces to be local to their learning trajectory, a form of bounded rationality, and b) having the agents consider improvements (derivatives) instead of direct outcomes. Furthermore, the fact that a lower price of anarchy entails a higher welfare at a Nash equilibrium means this style of WSLS is ultimately compatible with achieving high performance for the entire system.

Note that the trival solution of minimizing average group loss coincides with $A_{ij} = \frac{1}{n}$ for all i, j. If the agent strategies converge to a social optimum, this is a fixed point in the augmented strategy space $(\boldsymbol{x}, A)$. This can be seen by noting that 1) convergence to an optimum implies $\nabla_{x_i} f_i^A(\boldsymbol{x}) = 0$ and 2) convergence alone implies $\frac{df_i}{dt} = 0$ for all agents so $\nabla A_i = 0$ by equation 11 assuming $\epsilon = 0$.

2.6 Decentralized Learning & Extending to Reinforcement Learning

The time derivative of each agent's loss, $\frac{d}{dt} f_i^A(\boldsymbol{x})$, in equation 11 requires differentiating through potentially all other agent loss functions, which precludes scaling to large populations. In addition, this derivative is not always available as a differentiable function. In order to estimate $\tilde{\nabla}_{A_i}\rho_i$ when only scalar estimates of ρ_i are available as in, e.g., multiagent reinforcement learning (MARL), each agent perturbs their loss mixture and commits to this perturbation for a random number of training steps. If the loss increases over the trial, the agent updates their mixture in a direction *opposite* the perturbation. Otherwise, no update is performed.

This is formally accomplished with approximate one-shot gradient estimates [39] or *evolutionary strategies* [34]. A one-shot gradient of $\rho_i(A_i)$ is estimated by first perturbing A_i with entropic mirror ascent [4] as $\tilde{A}_i = \texttt{softmax}(\log(A_i) + \delta\tilde{\boldsymbol{a}}_i)$ where $\delta > 0$ and $\tilde{\boldsymbol{a}}_i \sim U_{sp}(n)$ is drawn uniformly from the unit sphere in $\mathbb{R}^n$. The perturbed weights are then evaluated $\tilde{\rho}_i = \rho_i(\tilde{A}_i)$. Finally, an unbiased gradient is given by $\frac{n}{\delta}\tilde{\rho}_i\tilde{\boldsymbol{a}}_i$. In practice, we cannot evaluate in one shot the $\frac{d}{dt} f_i^A(\boldsymbol{x})$ term that appears in the definition of ρ_i. Instead, Algorithm 1 uses finite differences and we assume the evaluation remains accurate enough across training steps.

Algorithm 1 D3C Update for RL Agent i

Input: η_A, δ, ν, $\tau_{\min}$, $\tau_{\max}$, A_i^0, ϵ, l, h, $\mathbb{L}$, iterations T
$A_i \leftarrow A_i^0$ {Initialize Mixing Weights}
$G = 0$ {Initialize Mean Return of Trial}
{Draw Initial Random Mixing Trial}
$\tilde{A}_i, \tilde{\boldsymbol{a}}_i, \tau, t_b, G_b = \texttt{trial}(\delta, \tau_{\min}, \tau_{\max}, A_i, 0, G)$
for $t = 1 : T$ **do**
  $g = \mathbb{L}_i(\tilde{A}_j \;\forall\; j)$ {Update Policy With Mixed Rewards}
  $\Delta t_b = t - t_b$ {Elapsed Trial Steps}
  $G = (G(\Delta t_b - 1) + g)/\Delta t_b$ {Update Mean Return}
  if $\Delta t_b == \tau$ {Trial Complete} **then**
    $\tilde{\rho}_i = \mathsf{ReLU}(\frac{G_b - G}{\tau} + \epsilon)$ {Approximate ρ}
    $\nabla_{A_i} = \tilde{\rho}_i\tilde{\boldsymbol{a}}_i - \nu\boldsymbol{e}_i \oslash A_i$ {Estimate Gradient —(11)}
    $A_i = \texttt{softmax}_l\lfloor\log(A_i) - \eta_A\nabla_{A_i}\rceil^h$ {Update}
    {Draw New Random Mixing Trial}
    $\tilde{A}_i, \tilde{\boldsymbol{a}}_i, \tau, t_b, G_b = \texttt{trial}(\delta, \tau_{\min}, \tau_{\max}, A_i, t, G)$
  end if
end for

Algorithm 1 requires several arguments: η_A is a global learning rate for each A_i, δ is a perturbation scalar for the one-shot gradient estimate, $\tau_{\min}$ and τ_{max} specify the lower and upper bounds for the duration of the mixing trial for estimating a finite difference of $\frac{d}{dt} f_i^A(\boldsymbol{x}) \approx -(G - G_b)/\tau$, l and h specify lower and upper bounds for clipping A in logit space ($_l\lfloor\cdot\rceil^h$), and $\mathbb{L}_i$ (Algorithm 3) represents any generic reinforcement learning algorithm augmented to take A as input (in order to mix rewards) and outputs *discounted return*. $\oslash$ indicates elementwise division.

Algorithm 2 `trial`—helper function

Input: δ, $\tau_{\min}$, $\tau_{\max}$, A_i, t, G
$\tilde{\boldsymbol{a}}_i \sim U_{sp}(n)$ {Sample Perturbation Direction}
$\tilde{A}_i = \texttt{softmax}(\log(A_i) + \delta\tilde{\boldsymbol{a}}_i)$ {Perturb Mixture}
$\tau \sim \texttt{Uniform}\{\tau_{\min}, \tau_{\max}\}$ {Draw Random Trial Length}
Output: $\tilde{A}_i, \tilde{\boldsymbol{a}}_i, \tau, t, G$

Algorithm 3 $\mathbb{L}_i$—example learner

Input: $\tilde{A} = [\tilde{A}_1; \ldots; \tilde{A}_n]$
while episode not terminal **do**
  draw action from agent policy
  play action and observe reward r_i
  broadcast r_i to all agents
  update policy with $\tilde{r}_i = \sum_j \tilde{A}_{ji} r_j$
end while
Output: return over episode g

2.7 Assessment

We assess Algorithm 1 with respect to our original design criteria. As described, agents perform gradient descent on a decentralized and local upper bound on the price of anarchy. Recall that a minimal global price of anarchy ($\rho = 1$) implies that even the worst case Nash equilibrium of the game is socially optimal; similarly, Algorithm 1 searches for a locally socially optimal equilibrium. By design, $A_i \in \Delta^{n-1}$ ensures the approach is budget-balancing. We justify the agents learning weight vectors A_i by initializing them to attend primarily to their own losses as in the original game. If they can minimize their original loss, then they never shift attention according to equation 11 because $\frac{df_i}{dt} \leq 0$ for all t. They only shift A_i if their loss increases. We also include a KL term to encourage the weights to return to their initial values. In addition, in our experiments with symmetric games, learning A helps the agents' outcomes in the long run. We also consider experiments in Appx. E.2.1 where only a subset of agents opt into the mechanism. If each agent's original loss is convex with diagonally dominant Hessian and the strategy space is unconstrained, the unique, globally stable fixed point of the game defined with mixed losses is a Nash (see Appx. H.4). Exact gradients $\nabla_{A_i}\rho_i$ require each agent differentiates through all other agents' losses precluding a fully decentralized and scalable algorithm. We circumvent this issue with noisy oneshot gradients. All that is needed in terms of centralization is to share the mixed scalar rewards; this is cheap compared to sharing $x_i \in \mathbb{R}^{d \gg 1}$. As mentioned in the introduction, the cost of communicating rewards can be mitigated by learning A_i via sparse optimization or sampling but is outside the scope of this paper.

2.8 Related Work

Collective Intelligence or COIN, surveyed in [46], examines the problem of how to design reward functions for individual agents such that a decentralized multiagent system maximizes a global world utility function. Wolpert and Tumer [46] describe several approaches taken by an array of diverse fields and motivate the creation of a collective intelligence as an important challenge. Follow-up works focus on aiding researchers in deriving static agent reward functions that are consistent with optimizing the desired world utility via, for instance, useful visualizations [1, 2]. Unlike conventional COIN approaches, D3C learns agent reward functions dynamically through online interaction with the environment. On the other hand, like D3C, studies in COIN find that agents optimizing modified versions of their original reward functions not only achieve high global utility, but also perform better individually [41].

In recent MARL work, Lupu and Precup [25] augment the agents' action space with a "gifting" action where agents can send a +1 reward to another agent. They evaluate this approach on a variant of *Harvest* we explore in Appx. F.4. They look at three different reward budget settings; ours is most similar to their zero-sum setting in which gifts are *budget-balanced* by matching −1 penalties. In contrast to [25], we consider a continuum of "gifting" amounts automatically grounded in the scale of the original rewards via mixing on the simplex.

Similarly, Hostallero et al. [16] introduce PED-DQN where agents gift their peers by a reciprocal amount proportional to the positive externality they perceive (as measured by their *td-error)* receiving from the group. Although they make no direct reference to price of anarchy, the stated goal is to shift the system's equilibrium towards an outcome that maximizes social welfare. In contrast to [16], D3C agents learn to share varying rewards with individual agents rather than sharing an average gift with everyone in their predefined peer group. This is important as the latter prevents the possible discovery of teams as demonstrated by D3C in Appx. F.5.

Yang et al. [47] propose an algorithm LIO (Learning to Incentivize Others) that equips agents with "gifting" policies represented as neural networks. At each time step, each agent observes the environment and actions of all other agents to determine how much reward to gift to the other agents. The parameters of these networks are adjusted to maximize the original environment reward (without gifts) minus some penalty regularizer for gifting meant to approximately maintain *budget-balance*. In order to perform this maximization, each agent requires access to every other agent's action-policy, gifting-policy, and return making this approach difficult to scale and decentralize. Yang et al. [47] demonstrate LIO's ability to maximize welfare and achieve division of labor on a very restricted version of the Cleanup game we evaluate in §3.5. We also evaluate D3C on this restricted variant in the Appx. F.3.

Inspired by social psychology, McKee et al. [26] explored imbuing agents with a predisposed *social value orientation* that modifies their rewards. Populations with heterogeneous populations achieved higher fitness scores than homogeneous ones in an evolutionary training approach (i.e., learning occurs outside the agent's lifetime).

One key innovation of D3C beyond the above works is its budget-balance guarantee. In [16, 26, 47], agents manifest extra reward to gift to peers, but no explanation is given for where this extra reward might come from. Also, none of these works tie their proposed approaches to the fundamental game theoretic concept price of anarchy. The derivation of D3C from first principles provides an explicit link, showing an agent-centric learning rule can be approximately consistent with the global objective of maximal social welfare.

Like D3C, OpenAI Five [31] also linearly mixed agents rewards which each other, but where the single "team spirit" mixture parameter (τ) is **manually** annealed throughout training from 0.3 to 1.0 (i.e., $A_{ii} = 1 - 0.8\tau, A_{ij} = 0.2\tau, j \neq i$).

Finally, we point out that loss transformation is consistent with human behavior. Within social psychology, *interdependence theory* [19] holds that humans make decisions based on self interest *and* social preferences, allowing them to avoid poor Nash equilibria.

3 EXPERIMENTS

Here, we show that agents minimizing local estimates of price of anarchy achieve lower loss on average than selfish, rational agents in five domains. In the first two domains, a traffic network (4 players) and a generalized prisoner's dilemma (10 players), players optimize using exact gradients (see equation 11). Then in three RL domains—Trust-Your-Brother, Coins and Cleanup—players optimize with approximate gradients as handled by Algorithm 1. Agents train with deep networks and A2C [11]. We refer to both algorithms as D3C (decentralized, differentiable, dynamic compromise).

For D3C, we initialize $A_{ii} = 0.99$ and $A_{ij} = \frac{0.01}{n-1}$, $j \neq i$. We initialize away from a onehot because we use entropic mirror descent [4] to update A_i, and this method requires iterates to be initialized to the interior of the simplex. In the RL domains, updates to A_i are clipped in logit-space to be within $l = -5$ and $h = 5$ (see Algorithm 1). We set the $\mathcal{D}_{KL}$ coefficient to 0 except for in Coins, where $\nu = 10^{-5}$. Additional hyperparameters are specified in Appx. G. In experiments where we cannot compute price of anarchy (equation 2) exactly, we either report the total loss of the learning algorithm (e.g., D3C) along with the loss achieved by fully cooperative agents ($A_{ij} = \frac{1}{n}$) or the ratio of these losses referred to as "ratio to optimal".

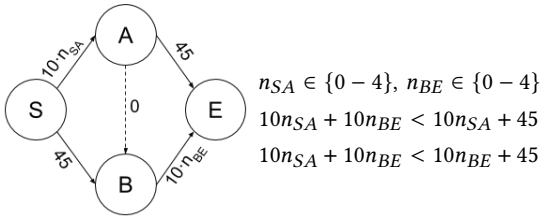
3.1 Traffic Networks and Braess's Paradox

In 2009, New York city's mayor closed Broadway near Times Square to alleviate traffic congestion [28]. This counter-intuitive phenomenon, where restricting commuter choices improves outcomes, is called Braess's paradox [5, 6, 43], and has been observed in real traffic networks [40, 48]. Braess's paradox is also found in physics [48], decentralized energy grids [45], and can cause extinction cascades in ecosystems [37]. Knowing when a network may exhibit this paradox is difficult, which means knowing when network dynamics may result in poor outcomes is difficult.

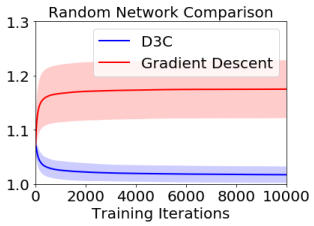
Figure 2a presents a theoretical traffic network. Without edge AB, drivers commute according to the Nash equilibrium, either learned by gradient descent or D3C. Figure 3a shows the price of anarchy approaching 1 for both algorithms. If edge AB is added, the network now exhibits Braess's paradox. Figure 3b shows that while gradient descent converges to Nash ($\rho = \frac{80}{65}$), D3C achieves an average "ratio to optimal" near 1. Figure 2b shows that when faced with a randomly drawn network, D3C agents achieve shorter commutes on average than agents without the ability to compromise.

3.2 Prisoner's Dilemma

In an n-player prisoner's dilemma, each player must decide to defect or cooperate with each of the other players creating a combinatorial action space of size 2^{n-1}. This requires a payoff tensor with $2^{n(n-1)}$ entries. Instead of generalizing prisoner's dilemma [33] to n players using nth order tensors, we translate it to a game with convex loss functions. Figure 4a shows how we can accomplish this. Generalizing this to n players, we say that for all i, j, k distinct, 1) player i wants to defect against player j, 2) player i wants player j to defect against player k, and 3) player i wants player j to cooperate with itself. In other words, each player desires a free-for-all with the exception that no one attacks it. See Appx. E.2 for more details.



(a) Traffic Network



(b) Random Network Results

Figure 2: (a) Four drivers aim to minimize commute time from S to E. Commute time on each edge depends on the number of commuters, n_{ij}. Without edge AB, drivers distribute evenly across SAE and SBE for a 65 min commute. After edge AB is added, switching to the shortcut, SABE, always decreases commute time given the other drivers maintain their routes, however, all drivers are incentivized to take the shortcut resulting in an 80 min commute. (b) The mean "ratio to optimal" over training for 1000 randomly generated networks exhibiting Braess's paradox with ±1 stdev shaded.

For three players, we can define the vector of loss functions with

$$f(x) = \sum_{columns} \left[\left(\begin{bmatrix} x^\top \\ x^\top \\ x^\top \end{bmatrix} - C \right)^2 \right] \tag{12}$$

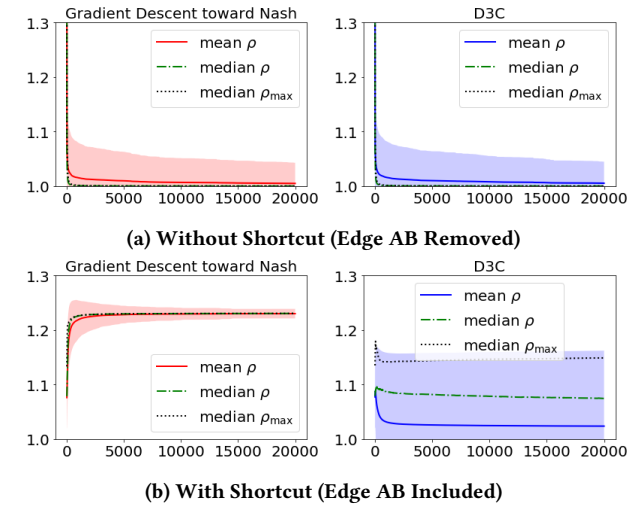
where $x = [x_{ij}]$ is a column vector ($i \in [1, n], j \in [1, n-1]$) containing the players' (randomly initialized) strategies and C is an $n \times n(n-1)$ matrix with entries that either equal 0 or $c \in \mathbb{R}^+$.

Figure 5 shows that D3C with a randomly initialized strategy successfully minimizes the price of anarchy. In contrast, gradient descent learners provably converge to Nash at the origin with $\rho = \frac{n}{c(n-1)}$. The price of anarchy grows unbounded as $c \to 0^+$. We set $n = 10$ and $c = 1$ ($\rho = \frac{10}{9}$) in this experiment with additional settings explored in Appx. F.1.

Figure 6 highlights a single training run. Both agents are initialized to minimize their original loss, but then learn over training to minimize the mean of the two player losses.

3.3 Trust-Your-Brother

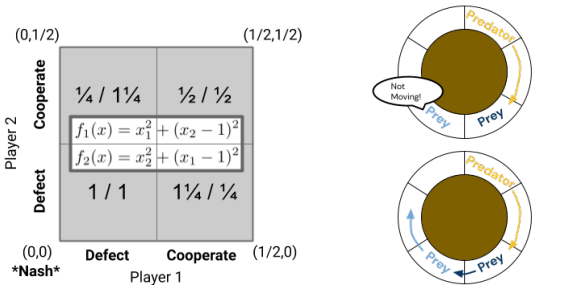
In this game, a predator chases two prey around a table. The predator uses a hard-coded policy to move towards the nearest prey unless it is already adjacent to a prey, in which case it stays put. If



(a) Without Shortcut (Edge AB Removed)

(b) With Shortcut (Edge AB Included)

Figure 3: Traffic Network—(a) Without edge AB, agents are initialized with random strategies and train with either gradient descent (left) or D3C (right)—similar performance is expected. Statistics of 1000 runs are plotted over training. Median $\rho_{\max}$ tracks the median over trials of the longest-commute among the four drivers. The shaded region captures ± 1 stdev around the mean. (b) After edge AB is added, agents are initialized with random strategies and trained with either gradient descent (left) or D3C (right).



(a) Prisoner's Dilemma **(b) Trust-Your-Brother**

Figure 4: (a) A reformulation of the prisoner's dilemma using convex loss functions instead of a normal form payoff table. (b) A bot chases two agents around a table. The predator's prey can only escape if the other prey simultaneously moves out of the way. Selfish (top), cooperative (bottom).

the prey are equidistant to the predator, the predator selects its prey at random. The prey receive 0 reward if they chose not to move and −.01 if they attempted to move. They additionally receive −1 if the predator is adjacent to them after moving.

The prey employ linear softmax policies (no bias term) and train via REINFORCE [44]. Both prey receive the same 2-d observation vector. The first feature specifies the counter-clockwise distance to the predator minus the clockwise distance for the blue prey. The second feature specifies the same for the green prey.

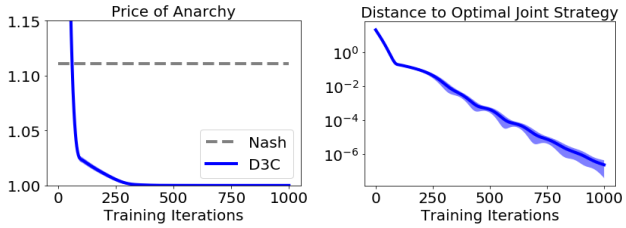


Figure 5: Prisoner's Dilemma—Convergence to $\rho = 1$ (left) and the unique optimal joint strategy (right) over 1000 runs. The shaded region captures ± 1 standard deviation around the mean (too small to see on left). Gradient descent (not shown) provably converges to Nash.

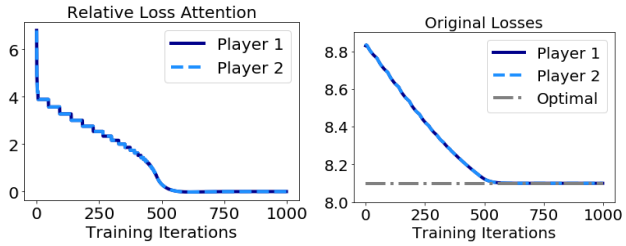


Figure 6: Prisoner's Dilemma—Single run: relative loss attention measured as $\ln\left(\frac{A_{ii}}{A_{j\neq i}}\right)$ (left) and player losses, f_i, (right).

Figure 7 shows D3C approaches maximal total return over training; this is achieved by the agents compromising on their original reward incentives and attending to those of the other agent instead.

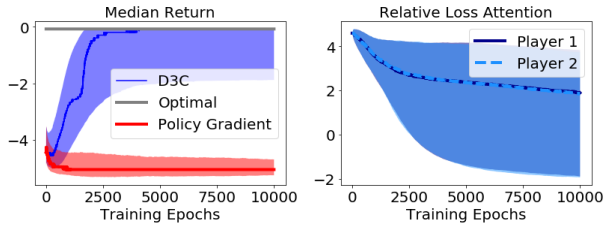
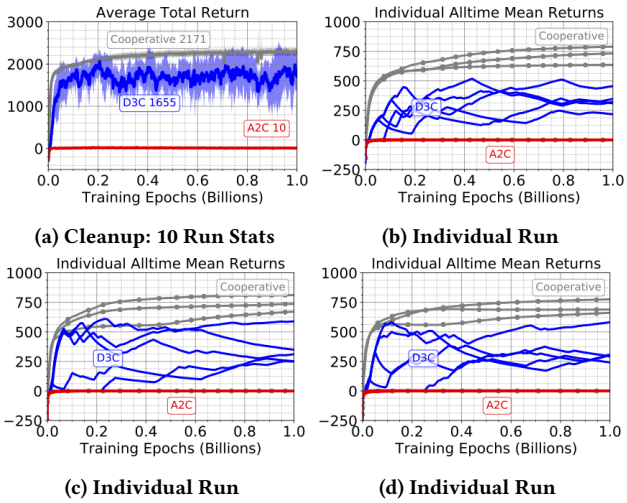


Figure 7: Trust-Your-Brother—Median return achieved during training for agents trained with policy gradient vs policy gradient augmented with D3C (left); relative reward attention is measured as $\ln\left(\frac{A_{ii}}{A_{j\neq i}}\right)$ where a positive value corresponds to selfish attention and a negative value to other-regarding (right). The ± 1 standard deviation shading about the mean for both players overlaps (1000 runs).

3.4 Coin Dilemma

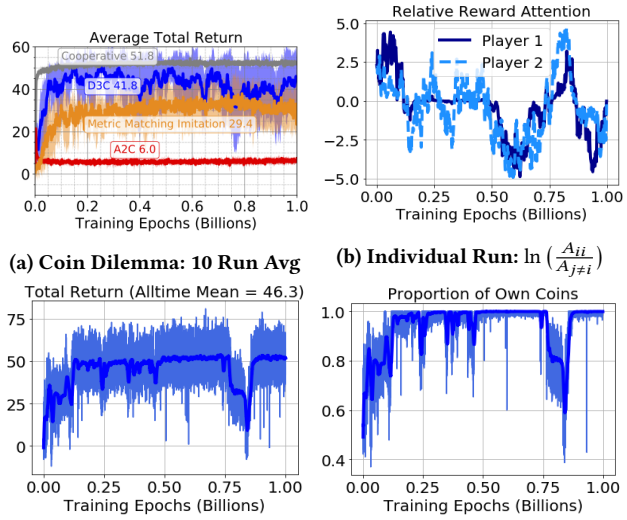
In the Coins game [8, 22], two agents move on a fully-observed 5×5 gridworld, on which coins of two types corresponding to each agent randomly spawn at each time step with probability 0.005. When an agent moves into a square with a coin of either type, they get a



(a) Cleanup: 10 Run Stats **(b) Individual Run**

(c) Individual Run **(d) Individual Run**

Figure 9: Cleanup (a) Mean total returns over ten training runs. D3C hyperparameters were selected using five independent validation runs. Cooperative agents trained to maximize total return represent the best possible baseline. Shaded region captures ± 1 standard deviation around the mean. (b-d) Three randomly selected runs. Each curve shows the mean return up to the current epoch for 1 of 5 agents.



(a) Coin Dilemma: 10 Run Avg **(b) Individual Run:** $\ln\left(\frac{A_{ii}}{A_{j\neq i}}\right)$

(c) Individual Run: Return **(d) Individual Run: Coins**

Figure 8: Coin Dilemma—(a) Mean total return over ten training runs for agents. Mean return over all epochs is reported in the legend. D3C hyperparameters were selected using five independent validation runs. Cooperative agents trained to maximize total return represent the best possible baseline. Shaded region captures ± 1 standard deviation around the mean. (b-d) One training run ($A^0_{ii} = 0.9$): relative reward attention measured as $\ln\left(\frac{A_{ii}}{A_{j\neq i}}\right)$ (b); sum of agent returns (c); % of coins picked up that were the agent's type (d).

reward of 1. When an agent picks up a coin of the other player's type, the other agent receives −2. The episode lasts 500 steps. Total reward is maximized when each agent picks up only coins of their own type, but players are tempted to pick up all coins.

D3C agents approach optimal cooperative returns (see Figure 8a). We compare against Metric Matching Imitation [9], which was previously tested on Coins and designed to exhibit reciprocal behavior towards co-players. Figure 8b shows D3C agents learning to cooperate, then temporarily defecting before rediscovering cooperation. Note that the relative reward attention of both players spikes towards selfish during this small defection window; agents collect more of their opponent's coins during this time. Oscillating between cooperation and defection occurred across various hyperparameter settings. Relative reward attention trajectories between agents appear to be reciprocal (see Appx. H.2 for analysis).

3.5 Cleanup

We provide additional results on Cleanup, a five-player gridworld game [17]. Agents are rewarded for eating apples, but must keep a river clean to ensure apples receive sufficient nutrients. The option to freeload and only eat apples presents a social dilemma. D3C increases both welfare and individual reward over A2C (no loss mixing). We also observe that direct welfare maximization (Cooperation) always results in three agents collecting rewards from apples while two agents sacrifice themselves and clean the river. In contrast, D3C avoids this stark division of labor. Agents take turns on each task and all achieve some positive cumulative return.

4 CONCLUSION

We formulate learning incentives as a price of anarchy minimization problem and propose a decentralized, gradient-based approach (D3C) that incrementally adapts agent incentives to the environment at hand. We demonstrate its effectiveness on achieving near-optimal agent outcomes in socially adversarial environments.

It is conceptually possible to scale our approach to very large populations through randomly sharing incentives according to the learned mixture weights or sparse optimization over the simplex [20, 24, 32], but we leave this challenge to future work.

ACKNOWLEDGMENTS

We are grateful to Jan Balaguer for fruitful discussions and advice on revising parts of the manuscript.

REFERENCES

[1] Adrian Agogino and Kagan Turner. 2005. Multi-agent reward analysis for learning in noisy domains. In *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems*. 81–88.
[2] Adrian K Agogino and Kagan Tumer. 2008. Analyzing and visualizing multiagent rewards in dynamic and stochastic domains. *Autonomous Agents and Multiagent Systems* 17, 2 (2008), 320–338.
[3] Kenneth J Arrow. 1970. *Social choice and individual values*. Vol. 12. Yale university press.
[4] Amir Beck and Marc Teboulle. 2003. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters* 31, 3 (2003), 167–175.
[5] Martin Beckmann, Charles B McGuire, and Christopher B Winsten. 1956. *Studies in the Economics of Transportation*. Technical Report.
[6] Dietrich Braess. 1968. Über ein Paradoxon aus der Verkehrsplanung. *Unternehmensforschung* 12, 1 (1968), 258–268.
[7] Edward H Clarke. 1971. Multipart pricing of public goods. *Public choice* (1971), 17–33.

[8] Tom Eccles, Edward Hughes, János Kramár, Steven Wheelwright, and Joel Z Leibo. 2019. The Imitation Game: Learned Reciprocity in Markov games. In *Proceedings of the 18th International Conference on Autonomous Agents and Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 1934–1936.

[9] Tom Eccles, Edward Hughes, János Kramár, Steven Wheelwright, and Joel Z Leibo. 2019. Learning Reciprocity in Complex Sequential Social Dilemmas. *arXiv preprint arXiv:1903.08082* (2019).

[10] Gerald M Edelman and Joseph A Gally. 2001. Degeneracy and complexity in biological systems. *Proceedings of the National Academy of Sciences* 98, 24 (2001), 13763–13768.

[11] Lasse Espeholt, Hubert Soyer, Remi Munos, Karen Simonyan, Volodymir Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, et al. 2018. Impala: Scalable distributed deep-RL with importance weighted actor-learner architectures. *arXiv preprint arXiv:1802.01561* (2018).

[12] Jakob Foerster, Richard Y Chen, Maruan Al-Shedivat, Shimon Whiteson, Pieter Abbeel, and Igor Mordatch. 2018. Learning with opponent-learning awareness. In *Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 122–130.

[13] Ian Gemp, Kevin R McKee, Richard Everett, Edgar A Duéñez-Guzmán, Yoram Bachrach, David Balduzzi, and Andrea Tacchetti. 2022. D3C: Reducing the Price of Anarchy in Multi-Agent Learning. *arXiv preprint arXiv:2010.00575* (2022).

[14] Jerry Green and Jean-Jacques Laffont. 1977. Characterization of satisfactory mechanisms for the revelation of preferences for public goods. *Econometrica: Journal of the Econometric Society* (1977), 427–438.

[15] Garrett Hardin. 1968. The tragedy of the commons. *Science* 162, 3859 (1968), 1243–1248.

[16] David Earl Hostallero, Daewoo Kim, Sangwoo Moon, Kyunghwan Son, Wan Ju Kang, and Yung Yi. 2020. Inducing cooperation through reward reshaping based on peer evaluations in deep multi-agent reinforcement learning. In *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems*. 520–528.

[17] Edward Hughes, Joel Z Leibo, Matthew Phillips, Karl Tuyls, Edgar Dueñez-Guzman, Antonio García Castañeda, Iain Dunning, Tina Zhu, Kevin McKee, Raphael Koster, et al. 2018. Inequity aversion improves cooperation in intertemporal social dilemmas. In *Advances in Neural Information Processing Systems*. 3326–3336.

[18] Lorens A Imhof, Drew Fudenberg, and Martin A Nowak. 2007. Tit-for-tat or win-stay, lose-shift? *Journal of Theoretical Biology* 247, 3 (2007), 574–580.

[19] Harold H. Kelley and John W. Thibaut. 1978. *Interpersonal Relations: A Theory of Interdependence*. John Wiley & Sons.

[20] Anastasios Kyrillidis, Stephen Becker, Volkan Cevher, and Christoph Koch. 2013. Sparse projections onto the simplex. In *International Conference on Machine Learning*. 235–243.

[21] Joel Z Leibo, Vinicius Zambaldi, Marc Lanctot, Janusz Marecki, and Thore Graepel. 2017. Multi-agent reinforcement learning in sequential social dilemmas. *arXiv preprint arXiv:1702.03037* (2017).

[22] Adam Lerer and Alexander Peysakhovich. 2017. Maintaining cooperation in complex social dilemmas using deep reinforcement learning. *arXiv preprint arXiv:1707.01068* (2017).

[23] Alistair Letcher, Jakob Foerster, David Balduzzi, Tim Rocktäschel, and Shimon Whiteson. 2018. Stable opponent shaping in differentiable games. *arXiv preprint arXiv:1811.08469* (2018).

[24] Ping Li, Syama Sundar Rangapuram, and Martin Slawski. 2016. Methods for sparse and low-rank recovery under simplex constraints. *arXiv preprint arXiv:1605.00507* (2016).

[25] Andrei Lupu and Doina Precup. 2020. Gifting in multi-agent reinforcement learning. In *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems*. 789–797.

[26] Kevin R McKee, Ian Gemp, Brian McWilliams, Edgar A Duéñez-Guzmán, Edward Hughes, and Joel Z Leibo. 2020. Social Diversity and Social Preferences in Mixed-Motive Reinforcement Learning. In *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems*. 869–877.

[27] Roger B Myerson and Mark A Satterthwaite. 1983. Efficient mechanisms for bilateral trading. *Journal of Economic Theory* 29, 2 (1983), 265–281.

[28] William Neuman and Michael Barbaro. 2009. Mayor Plans to Close Parts of Broadway to Traffic. https://www.nytimes.com/2009/02/26/nyregion/26broadway.html. *NYTimes.com* (Feb 2009).

[29] Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V Vazirani. 2007. *Algorithmic game theory*. Cambridge university press.

[30] Martin Nowak and Karl Sigmund. 1993. A strategy of win-stay, lose-shift that outperforms tit-for-tat in the Prisoner's Dilemma game. *Nature* 364, 6432 (1993), 56.

[31] OpenAI, Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, Rafal Józefowicz, Scott Gray, Catherine Olsson, Jakub Pachocki, Michael Petrov, Henrique Pondé de Oliveira Pinto, Jonathan Raiman, Tim Salimans, Jeremy Schlatter, Jonas Schneider, Szymon Sidor, Ilya Sutskever, Jie Tang, Filip Wolski, and Susan Zhang. 2019. Dota 2 with Large Scale Deep Reinforcement Learning. (2019). arXiv:1912.06680 https://arxiv.org/abs/1912.06680

[32] Mert Pilanci, Laurent E Ghaoui, and Venkat Chandrasekaran. 2012. Recovery of sparse probability measures via convex programming. In *Advances in Neural Information Processing Systems*. 2420–2428.

[33] Anatol Rapoport, Albert M Chammah, and Carol J Orwant. 1965. *Prisoner's dilemma: A study in conflict and cooperation*. Vol. 165. University of Michigan press.

[34] Ingo Rechenberg. 1978. Evolutionsstrategien. In *Simulationsmethoden in der Medizin und Biologie*. Springer, 83–114.

[35] Herbert Robbins. 1952. Some aspects of the sequential design of experiments. *Bull. Amer. Math. Soc.* 58, 5 (1952), 527–535.

[36] Tim Roughgarden. 2015. Intrinsic robustness of the price of anarchy. *Journal of the ACM (JACM)* 62, 5 (2015), 32.

[37] Sagar Sahasrabudhe and Adilson E Motter. 2011. Rescuing ecosystems from extinction cascades through compensatory perturbations. *Nature Communications* 2 (2011), 170.

[38] Mark Allen Satterthwaite. 1975. Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory* 10, 2 (1975), 187–217.

[39] Shai Shalev-Shwartz et al. 2012. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning* 4, 2 (2012), 107–194.

[40] Richard Steinberg and Willard I Zangwill. 1983. The prevalence of Braess' paradox. *Transportation Science* 17, 3 (1983), 301–318.

[41] Kagan Tumer and Scott Proper. 2013. Coordinating actions in congestion games: impact of top–down and bottom–up utilities. *Autonomous Agents and Multiagent Systems* 27, 3 (2013), 419–443.

[42] Zhijian Wang, Bin Xu, and Hai-Jun Zhou. 2014. Social cycling and conditional responses in the Rock-Paper-Scissors game. *Scientific Reports* 4, 1 (2014), 1–7.

[43] John Glen Wardrop. 1952. Some theoretical aspects of road traffic research. *Proceedings of the Institution of Civil Engineers* 1, 3 (1952), 325–362.

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

[45] Dirk Witthaut and Marc Timme. 2012. Braess's paradox in oscillator networks, desynchronization and power outage. *New Journal of Physics* 14, 8 (2012), 083036.

[46] David H Wolpert and Kagan Tumer. 1999. An introduction to collective intelligence. *arXiv preprint cs/9908014* (1999).

[47] Jiachen Yang, Ang Li, Mehrdad Farajtabar, Peter Sunehag, Edward Hughes, and Hongyuan Zha. 2020. Learning to Incentivize Other Learning Agents. *Advances in Neural Information Processing Systems* 33 (2020).

[48] Hyejin Youn, Michael T Gastner, and Hawoong Jeong. 2008. Price of anarchy in transportation networks: efficiency and optimality control. *Physical Review Letters* 101, 12 (2008), 128701.