An Analysis of Connections Between Regret Minimization and Actor Critic Methods in Cooperative Settings

Chirag Chhablani
University of Illinois at Chicago
Chicago, Illinois, USA
cchhab2@uic.edu

Ian A. Kash
University of Illinois at Chicago
Chicago, Illinois, USA
iankash@uic.edu

ABSTRACT

Counterfactual Multi-agent Policy Gradients (COMA) is a popular algorithm for learning in cooperative multi-agent reinforcement learning settings. COMA computes difference rewards to solve the multiagent credit assignment problem by providing a local learning signal for each agent. Similar to other popular Cooperative multiagent RL (MARL) algorithms, there is a lack of theoretical justification for COMA's empirical success and specific way of doing credit assignment using difference rewards. We provide such a justification by connecting COMA's update rule to regret minimization in identical interest games. This leads to two further theoretical insights. First, COMA's update rule may lead to slow policy updates even in very simple environments. This can be ameliorated by a slight modification of the policy gradient update for COMA as was observed in the Neural Replicator Dynamics algorithm. Second, this provides a justification for the use of a bounded softmax policy in terms of a guarantee of favorable convergence rates in identical interest games. Experimental results show the relevance of these theoretical insights for the performance of COMA in practice. Our work compliments existing works on theoretically understanding the best practices and assumptions made in cooperative MARL.

ACM Reference Format:
Chirag Chhablani and Ian A. Kash. 2023. An Analysis of Connections Between Regret Minimization and Actor Critic Methods in Cooperative Settings . In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), London, United Kingdom, May 29 – June 2, 2023*, IFAAMAS, 11 pages.

1 INTRODUCTION

Cooperative multi-agent reinforcement learning (Coop-MARL) is a framework for many complex real-world reinforcement learning problems such as the coordination of autonomous vehicles [5], network packet delivery [47], and distributed logistics [48]. *Counterfactual Multi-Agent Policy Gradients (COMA)* is a recent technique for learning cooperation among agents [12] which showed early empirical success in popular cooperative multiagent learning benchmarks like StarCraft II where each agent has to cooperate with the other agents to maximize the single shared reward when each agent only has partial access to the state of game. Key to COMA's success is efficient multiagent credit assignment through the implementation of *difference rewards* which were proposed by Wolpert and Tumer [43] and Tumer and Agogino [40]. COMA uses difference rewards to evaluate the contribution of each agent's actions by comparing with the expected value of actions based on its current policy. For each agent, the difference reward signal represents the advantage of including the agent in the system compared to the counterfactual case when it is excluded from the system. This individual advantage signal is called the *counterfactual advantage baseline*. Despite good performance, there is lack of theoretical justification for *this particular* choice of baseline. We argue that this baseline works well because it is similar to minimizing regret in a cooperative setting.

Similar to very recent work on explaining popular value decomposition methods [15] and DDPG based methods [18] for Cooperative MARL, our theoretical justifications for COMA consider a identical interest game (also known as fully cooperative game or global/shared reward game), which is a single-state version of shared reward Coop-MARL. This allows us to connect a variant of COMA's update rule to the classic regret minimization algorithm Hedge. Regret minimization algorithms like Hedge in identical interest games have stronger properties than in competitive settings. For example in some cases it is possible to show last iterate convergence to a pure Nash equilibrium [27], while in competitive settings it is typically only possible to show average convergence to the Nash equilibrium.

Beyond the conceptual contribution of this high-level justification for COMA, we use this connection to make two further observations. First, we show that the current version of gradient update rule of COMA can lead to slow learning even in simple environments. This problem, and a solution to it known as Neural Replicator Dynamics (NeuRD) update. The NeuRD update has previously been empirically examined in stateful competitive settings [17] and has good convergence properties in stateless settings. In our experimental results, we show that such an update rule, when applied with decreasing learning rates (whose important is emphasized by our analysis), leads to accelerated learning and higher performance in game environments like StarCraft and N-step Matrix without adding any extra cost. Our results extend the applicability of the NeuRD update to algorithms for cooperative settings.

Second, we give a theoretical justification for COMA's use of a bounded softmax policy update instead of the more common softmax update. This update rule has been shown in recent work on potential games (a superset of identical interest games) to give faster convergence to pure Nash Equilibrium when updates are only applied to the action taken [6]. As this is the case for COMA, this connection provides an intuition for the good performance of this specific choice of policy representation.

In summary, our contribution, like parallel works for value decomposition techniques [8, 15], is conceptual and theoretical: connecting COMA's particular choice of baseline to regret minimization and exploring the implications. Our empirical results demonstrate

Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), A. Ricci, W. Yeoh, N. Agmon, B. An (eds.), May 29 – June 2, 2023, London, United Kingdom.

the relevance of the theory *for COMA*. Although COMA-N significantly improves the performance of COMA in many environments, we make no claims about state-of-the-art performance or comparison to other algorithms which have their own limitations. See the related work for additional discussion regarding the relevance of this approach.

2 RELATED WORK

Explaining success and failure modes of current RL methods is an active area of research. Prior work has explained the the role of the target network [50], discount factor[1, 32], and experience replay[49] in Deep RL methods. For cooperative MARL methods, although recent methods achieve strong empirical performance, there is lack theoretical justification of the assumptions and engineering practices employed to achieve the state-of-art performance on popular benchmarks. Recent work has therefore examined and modified the popular assumptions employed in such methods and tried to justify the engineering practices to improve the current understanding of cooperative MARL methods. Yang et al. [45, 46] explained the limitations of Individual Global Maximum condition of value decomposition methods and proposed alternate way of achieving value decomposition. Recent work has derived insights from coordination games to explain the cooperative MARL methods and find the limitations. For example, [8] illustrated the limitation of popular IGM condition using cooperative Markov Games. [18] illustrated the importance of credit assignment and fair local reward signals in cooperative MARL through identical interest games. More recently, Fu et al. [15] explained the importance and limitation of parameter sharing and value decomposition in identical interest XOR game and give practical recommendations for implementing policy gradient methods in multi-modal reward landscape. Li et al. [25] explained the trade off between policy bias and credit assignment through an N-step identical interest game. Complementing this line of work, we analyze COMA's design by giving theoretical justification for its particular counterfactual baseline and use of bounded softmax using Regret Minimization in identical interest games. Further, we use this connection to show the slow nature of standard updates in MARL settings and implement NeuRD update, which had prior known success in Actor-Critic methods for competitive games like poker.

For Coop-MARL settings where agents receive a single global reward, decentralized learning leads to the problem of multiagent credit assignment which is addressed by many techniques, including COMA, by generating a local reward for each agent. Such an approach typically performs better than directly using the global reward for each agent [18]. Various ways to assign local rewards from a single global reward have been proposed in previous work. Apart from COMA, Nguyen et al. [29] employ count-based variance reduction. Jianhong Wang et al. [18] model such cooperative multiagent problems as extended convex game and propose the Shapley Q-value. Yang et al. [44] perform credit assignment for more general case where individual agents have their own individual goals as well. Apart from using COMA-like advantages as a learning signal, for each agent they also use another learning signal which evaluates the alignment of each its action to individual goals of the other agents. Other recent approaches include those taken by Wang et al. [42], Zhou et al. [52] and Son et al. [35].

Apart from credit-assignment, joint value decomposition approaches have also shown good performance especially for the StarCraft micromanagement testbed. In particular, Value decomposition [39], in its original form, represents joint action-values as a summation of local action-value conditioned on individual agents' local observation history using Value Decomposition Networks (VDN) and then uses the local values for the learning of the agents. Moreover, recent approaches [33, 36] have used different factorization functions for joint action values of the players and, despite their convoluted architecture, [2] give good empirical results for many cooperative environments like the StarCraft micromanagment testbed. However, recent work by Dou et al. [8] has shown that these factorization approaches can restrict the way total value is factorized and so their application is mostly limited to a class of games, called decomposable games, where the reward, transition and Q function can be decomposed amongst agents. Further, as pointed out by Su et al. [38], actor critic approaches are more sample efficient than value decomposition approaches and can be combined with value decomposition approaches using value factorization functions [31] and maximum entropy loss functions [51] to guarantee optimal global convergence. Finally, COMA and credit assignment methods still lack the theoretical understanding which is prevalent for Value decomposition methods [8, 41, 45]. *Hence, we believe there is value in improving the performance and our understanding of COMA-style algorithms even if COMA does not currently achieve state-of-the-art performance on the current popular benchmarks such as StarCraft.*

Regret minimization algorithms have been extensively studied in stateless multiagent settings and provide guarantees about convergence to Nash equilibrium in zero-sum games and (coarse) correlated equilibrium in general-sum games [14, 16]. Their primary extension to stateful settings is the Counterfactual Regret Minimization (CFR) algoritm [53], which provides similar rigorous guarantees under restrictive assumptions of perfect recall and terminal states. More recently a line of work has begun to explore connections between policy gradient approaches and regret minimization. Advantage Regret Minimization [19] drew an analogy between advantages and CFR's update rule and used it to demonstrate strong performance in single agent POMDP settings. Srinivasan et al. [37] explored different variants of actor critic algorithms that exploit this insight in stochastic games. In an approach we build on, Hennes et al. [17] show how to draw an exact equivalence between regret minimization and policy gradient approaches by restricting to stateless settings. Recently, Li et al. [24] also use the idea of advantages for regret minimization in a cooperative setting, but they define team regret for group of agents and, similar to [33], they explore ways to divide the team regret for decentralized learning rather than directly doing regret minimization on the global reward.

3 BACKGROUND

3.1 Cooperative Stochastic Games

COMA learns in a fully cooperative, global reward, partially observable, multi-agent environment that can be modelled as a stochastic game G, also known in this special case as a DEC-POMDP

[4], defined by a tuple $G = (S, U, P, r, Z, O, n, \gamma)$. The number of agents is denoted by n, and we denote an arbitrary individual agent by $a \in A = \{1, ..., n\}$. The true state of the game is denoted by $s \in S$. At each time step t, every agent simultaneously chooses an action $u_a \in U$, forming a joint action $\mathbf{u} \in \mathbf{U} = U^n$. All the agents receive same global reward $r(s, \mathbf{u}) : S \times \mathbf{U} \rightarrow \mathbb{R}$ and the game transitions to a new state according to the transition function $P(s' \mid s, \mathbf{u})$. The goal of G is to maximize the expected discounted reward $R_t = \sum_{t=0}^{\infty} \gamma^t r_{t+1}$ where $\gamma \in [0, 1)$ is a discount factor. (In general quantities in bold will be used to denote joint variables and the notation $-a$ will used to denote variables for agents excluding agent a.) Further, we assume that from an agent's perspective, the environment is partially observable where the agents draw observations $z \in Z$ according to the observation function $O(s, a) : S \times A \rightarrow Z$. The agent maintains action-observation history $\tau_a \in T = (Z \times U)^* \times Z$, on which it conditions a stochastic policy $\pi^a(u_a|\tau_a) : T \times U \rightarrow [0, 1]$.

3.2 COMA

COMA uses centralized training and decentralized execution (CTDE) learning framework to guide the training for each agent. In CTDE, the agents have access to the true state information and actions during the training phase while they only observe the local action-observation history during the actual execution in the environment. COMA uses an actor-critic algorithm to maximize the discounted global reward for G and learn a policy for each agent. It uses a centralized critic with decentralized actor networks which implements the CTDE framework to update policy for each agent as shown in Algorithm 3 in the appendix. The input to the critic is the current state s_t and joint action $\mathbf{u}_t$ while the input to the actor network is the history h^a_t. The goal of the centralized critic is to estimate a value functions using sampled trajectories trajectories according to some current policy to learn the value functions $Q_\pi(s, \mathbf{u}) = \mathbb{E}_\pi[\sum_{k=t}^{\infty}[\gamma^{k-t} r_k \mid s_t = s, \mathbf{u}_t = \mathbf{u}]$ and $V_\pi(s) = \mathbb{E}_\pi[\sum_{k=t}^{\infty}[\gamma^{k-t} r_k \mid s_t = s]$.

Each agent plays an action u^a_t conditioned on agent's history h^a_t and sampling from current policy $\pi^a(u|h^a_t)$. The agents get the global reward r_t and reach the next state next state s_{t+1}. It uses this information to update the critic parameters θ^c by minimizing the mean square loss between the critic's target values y_t and critic values $Q(s, \mathbf{u})$. The critic's target network parameters are updated after every C steps. For updating the actor network, COMA uses difference rewards [40, 43] which measures the contribution of each agent to the global reward. COMA's equation for difference rewards is:

$$A^a(s, u^a, \mathbf{u}^{-a}) = Q(s, \mathbf{u}) - \sum_{u'^a} \pi^a(u'^a|\tau_a) Q(s, (u'^a, \mathbf{u}^{-a})) \quad (1)$$

The $A^a(s, u^a)$ is referred to as the *counterfactual baseline.* The first term on right hand side measures expected reward $Q(s, \mathbf{u})$ when agent a takes action u while second term is counterfactual scenario when a is removed from the system by subtracting its expected value based on current policy. This counterfactual baseline is used to update π^a (parameterized by θ^π_a) according to the standard policy gradient equation for actor-critic setting. For a given state-action pair $(s, \mathbf{u})$) this is given by:

$$\theta_t = \theta_{t-1} + \eta_t \sum \nabla \log \pi^a_\theta(u^a|\tau^a; \theta_{t-1}) A^a(s, \mathbf{u}) \quad (2)$$

The centralized value network enables faster calculation of difference rewards as it reduces number of parameters for training and the number of forward passes required to estimate counterfactual advantage values while decentralized actor networks are useful in representing broad range of policies for different agents and allows the decentralized execution.

3.3 Identical Interest Game

For our analysis of COMA, we use the special case where there is a single state, also known as an identical interest game in the game theory literature [18, 26]. All identical interest games have at least one joint action $\mathbf{u}$ which is a maximizer of the shared reward $r(\mathbf{u})$. With such a joint action, indeed for any local optimum of r, for all agents a and alternate actions u'_a,

$$r(u_a, \mathbf{u}_{-a}) \geq r(u'_a, \mathbf{u}_{-a}). \quad (3)$$

Such a point is known as a pure Nash equilibrium.

3.4 Regret

Regret is a tool for for measuring the relative performance of a single action to a policy in an online setting. For an identical interest game, regret measures how much the agent's particular action led to a better global reward for the system while keeping the actions of other agents $\mathbf{u}^{-a}$ fixed.

$$Regret(a, u^a) = r(u^a, \mathbf{u}^{-a}) - \sum_u \pi^a(u) r(u, \mathbf{u}^{-a}) \quad (4)$$

A key observation, made in prior works, is the structural similarity between Equations (1) and (4). We use this observation to show in Theorem 1 that in special case of cooperative stochastic games, the contribution of each agent in the team is equivalent to regret.
The average or the external regret is the average of instantaneous regret from time $t = 1$ to $t = T$.

$$R^T_{avg}(u, a) = 1/T \sum_{t=1}^{T} Regret(a, u) \quad (5)$$

and is minimized by an class of algorithms known as regret minimization, or no-regret, algorithms. The objective of no-regret algorithms is to reduce the average regret given by equation (5) to zero. One such algorithm is Hedge, which maintains a policy based on the sum of regrets[1], weighted by a learning rate [13].

$$R^T_{sum}(u^a, a) = \sum_{t=1}^{T} \eta_t Regret(a, u^a) \quad (6)$$

The policy update equation of agent a at time t is given by:

$$\pi^a_T = \exp(R^T_{sum}(u^a, a)) / \sum_u \exp(R^T_{sum}(u, a)) \quad (7)$$

Hedge is no-regret when η_t is chosen carefully, e.g.,$\eta_t \in \Theta(1/\sqrt{t})$ [28].

[1] Typically Hedge is defined to track the sum of rewards rather than regrets, but both versions are equivalent in that they output the same sequence of policies on the same sequence of inputs.

4 NEW INTERPRETATION AND ANALYSIS OF COMA

In this section we make three contributions. First, we give a new interpretation of COMA's difference rewards in terms of regret minimization in an identical interest setting. Second, inspired by approaches based on regret minimization in general settings [17], we point out that COMA's update can be slow even in many simpler settings and that a simple modification can lead to improved performance. Third, we show that this connection provides a principled rationale for COMA's use of a bounded softmax.

4.1 Justifying COMA's Difference Rewards

To calculate its difference rewards, COMA uses a centralized critic which calculates the counterfactual advantage baseline A^a. A^a compares the value of $Q(s, \mathbf{u})$ and $\sum \pi(u'^a|\tau_a)Q(s, (u'^a, \mathbf{u}^{-a}))$ for agent a and it's action u^a. However, the correctness proof for COMA is independent of the choice of baseline, *leaving no theoretical justification* for this particular choice other than by analogy to prior successes of difference reward approaches.

The literature on regret-like policy gradient methods in single agent [19] and competitive settings [37] has also emphasized the similarities between advantage calculations and the update rule used by counterfactual regret minimization (CFR) [53]. However, we show that in the special case of cooperative stochastic games, the credit assigned to each agent through the advantage value is equivalent to the regret of the agent. [2] Based on this connection, we provide an interpretation of and justification for COMA's difference reward implementation in terms of regret minimization in cooperative settings as Theorem 1. To be able to state the theorem, we begin by introducing a variant of COMA for this setting which uses all-actions updates and a tabular representation. We call this version Tabular COMA, and it is given in Algorithm 1.

Algorithm 1: Tabular COMA-N update for an agent **a** having **K** actions

```
Result: Update policy for agent $a$ given by $\pi^a$ at state $s$
Sample joint action $\mathbf{u}^{-a}$ from other agents' policy $\pi(\mathbf{u}^{-a})$;
for $k = 1$ to $K$ do
    $A^a(s, u_k) \leftarrow Q(s, u^a_k, \mathbf{u}^{-a}) - \sum_{u'} \pi^a(u')Q(s, u', \mathbf{u}^{-a})$
    $A^a_{sum}(s, u_k) \leftarrow A^a_{sum}(s, u_k) + \eta A^a(s, u_k)$
    $\pi^a(s) \propto \exp(A^a_{sum}(s))$
end
```

Tabular COMA has two changes from Algorithm 3. First, to aid in making the connections to regret minimization explicit we use an all-actions update rule rather than solely updating the action taken as COMA does. Second, we assume a softmax policy parameterized by a tabular representation where each parameter gives the logit of the policy for that action and agent. This leads to the given update rule for policies [17, equation (6)].

Tabular COMA is the natural all-actions implementation of COMA in the setting of stateful identical interest game. Hennes et al. [17] derive essentially the same algorithm in stateless general-sum games from Softmax Policy Gradient. As they point out, algorithms like Tabular COMA do not quite match up with regret minimization. The issue is the inclusion of the $\pi^a(u_k)$ term in the update for $A^a_{sum}(u_k)$ in Algorithm 1. To exactly match up with the Hedge algorithm (i.e. Equation (6)) it should instead be omitted yielding

$$A^a_{sum}(s, u_k) \leftarrow A^a_{sum}(s, u_k) + \eta A^a(s, u_k) \tag{8}$$

We refer to Algorithm 1 with the update according to Equation (8) as Tabular COMA-N, with the "N" representing the inclusion of this "NeuRD fix."

With this variant, we can make a precise connection between COMA's difference rewards and regret minimization.

Theorem 1. *Given a joint action $(u, \boldsymbol{u}^{-a})$, Tabular COMA-N is equivalent of running to a copy of Hedge at every state s with $Q(s, u, \boldsymbol{u}^{-a})$ as the reward for agent a.*

Proof.

$$\begin{aligned}
\pi^a_t(s) &\propto exp(R^t_{sum}(u, a)) \\
&= exp\left(\sum_{\tau=1}^{t} \eta_\tau Regret^t(a, u)\right) \\
&= exp\left(\sum_{\tau=1}^{t} \eta_\tau \left(Q(s, u, \mathbf{u}^{-a}) - \sum_{u'} \pi^{a,\tau}(u')Q(s, u', \mathbf{u}^{-a})\right)\right) \\
&= exp\left(\sum_{\tau=1}^{t} \eta_\tau A^a(s, u)\right) \\
&= exp(A^a_{sum}(s, u)) \\
&\propto \pi^a(s)
\end{aligned}$$

□

Theorem 1 makes a conceptual contribution by establishing the equivalence of two concepts which have previously been explored separately in games with identical interests: difference rewards and regret minimization. In doing so it also connects to the rapidly growing literature on algorithms that learn in stateful settings via a collection of regret minimizers [3, 9–11, 20]. In particular, Tabular COMA-N can be viewed as a variant of LONR [21], so Theorem 1 combined with the general convergence guarantees of COMA provides a novel extension of convergence guarantees for LONR-style algorithms from MDPs to a stateful multi-agent setting.

Similar to prior works [15, 41], we now turn to the more restricted setting of identical interest games to derive intuition for aspects of COMA's design. To begin, the literature on regret minimization in identical interest games shows that they enjoy nice convergence properties.

Corollary 2. *In an identical interest game, Tabular COMA-N is equivalent to all agents independently using the Hedge algorithm with immediate reward, in that all agents choose the same policy after each update under both.*

[2] There is a terminology conflict between these two literatures. The literature on difference rewards views the counterfactual advantage as representing *"How much does the agent currently contributes through the actual action u^a to the overall goal of the system, compared to the counterfactual case when the agent is not present in the system?"* while the literature on regret minimization[19, 37] interprets it as *"How much benefit do we do we get by taking a counterfactual action u^a instead of adhering to the actual policy $\pi^a(u^a|\tau_a)$ while keeping the policies of other fixed?"*, resulting in a difference in which part of the update is considered the counterfactual.

Proof. For an identical interest game, $Q(s, \mathbf{u})$ is the immediate reward $r(\mathbf{u})$. By Theorem 1, $\pi^a \propto exp(\sum_{\tau=1}^{T} \eta_\tau (A^\tau(u^a, \mathbf{u}))$. Since $\sum_{\tau=1}^{\tau} r^\tau(u^a, \mathbf{u}) - \sum_{u'} \pi^a(u') r^\tau(u', \mathbf{u}^{-a})) = exp(R_{sum}^{T}(u^a, a))$ this is equivalent to Hedge algorithm in equation 7. □

Corollary 2 provides a satisfying justification for COMA's particular choice of baseline for its difference rewards in terms of an approximation of regret minimization.[3] While regret minimization algorithms like Hedge have a long history and many appealing properties in normal-form games in general, a line of work has shown that they have particularly appealing properties in potential games (also known as congestion games), a generalization of identical interest games. Kleinberg et al. [22] show that players who use Hedge-like updates end up playing a pure equilibrium for a fraction of time that is arbitrarily close to 1 with probability also arbitrarily close to 1 after a polynomially small transient stage. Mehta et al. [27] showed that the multiplicative weights algorithm (MW) converges to a pure Nash equilibrium for all but a measure 0 of initial conditions, and hence obtained a stronger guarantee, in identical interest games. Palaiopanos et al. [30] showed that a version of the MW update rule converges to equilibrium in potential games. For the bandit setting, Coucheney et al. [7] showed that a "penalty-regulated" variant of the MW algorithm converges to ϵ-approximate Nash equilibria in congestion games with bandit feedback.

4.2 COMA vs COMA-N

While our use of Tabular COMA-N rather than Tabular COMA was necessary to make a precise connection to Hedge, Hennes et al. [17] argue that the NeuRD fix is also important to performance. In particular, if $\pi^a(u_k)$ is small then weighting by it results in slow updates even if u_k is a substantially superior action. They demonstrate that this leads to both slow adaptation or even a lack of convergence in some normal form games. It is not immediate that the same issues arise in cooperative settings. Identical interest games are known to be a particularly well-behaved class of games. In particular, they are special cases of potential games (and even more generally weakly acyclic games) which are relatively forgiving in the sorts of learning dynamics which are guaranteed to converge [26]. We argue that while a lack of the NeuRD fix will not prevent convergence[4], it can substantially harm the rate of convergence.

Consider the three agent identical interest game given in Table 1. In this game Agent 1 can take actions (L, R), Agent 2 can take actions (U, D), and Agent 3 can take actions (M_1, M_2, M_3). This game has a unique Nash Equilibrium (U, L, M_2), but M_2 is a poor choice unless the other two agents are playing their part of the equilibrium. Thus, agent 3 typically needs to "unlearn" playing M_0.

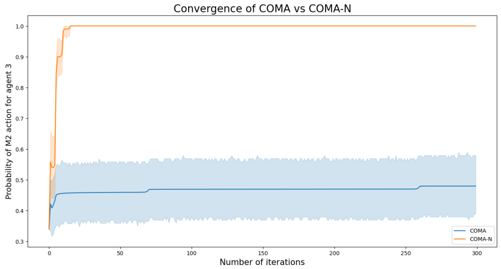

Figure 1: COMA with regular softmax update takes longer time to converge to Nash Equilibrium policy whereas Tabular COMA-N quickly reaches the Nash Equilibrium policy.

	L	R
U	4	3
D	2	1

M_0

	L	R
U	-4	-5
D	-5	-6

M_1

	L	R
U	6	-10
D	-20	-20

M_2

Table 1: *A 3-player Identical Interest Game*

Figure 1 shows the results of 100 runs of Tabular COMA and Tabular COMA-N on this game. Despite the tiny size, with 300 iterations of policy update Tabular COMA couldn't reach the Nash Equilibrium in the time allowed in about half of the runs. This is because, while finding the policy update for agent 3, Tabular COMA uses a sample of the policies of agents 1 and 2. This sampling will often lead to bad early rewards for M_2 and thus small values of $\pi^3(M_2)$. In contrast,Tabular COMA-N learns rapidly in all runs.

In the experimental section, we show that the benefits of applying the NeuRD fix to COMA hold in richer, stateful settings and not merely in identical interest games.

4.3 Justifying Bounded Softmax

The theoretical analysis of COMA by Foerster et al. [12] is agnostic about the choice of policy representation for each actor. For Tabular COMA we chose a softmax policy to make a precise connection to Hedge possible. The experimental analysis of COMA instead used a bounded softmax policy, where a parameter ϵ is used to put a weight of $1 - \epsilon$ on the softmax policy and a weight of ϵ on the uniform policy. COMA starts with a 0.5 value of ϵ value and anneals it to a minimum value of 0.01 over the first $10K$ iterations and keeps it constant for rest of $190K$ iterations. To explain this uncommon choice of a bounded softmax update with minimum exploration, we make another connection to prior work showing that this choice of bounded softmax with minimum exploration *is necessary* for strong convergence rate guarantees in identical interest games especially when players do COMA-like update for their own policies where they don't observe the payoff for all their actions but only for the actions that they take.

The use of bounded softmax is linked to COMA's updating of actions taken rather than all actions. Thus, rather than Tabular COMA, we use another variant which does not perform all actions

[3] In addition to the lack of the NeuRD fix, COMA is also missing the necessary weights, in the form of reach probabilities, used by algorithms like CFR [53] to ensure correct regret minimization in stateful settings. [17].

[4] That Tabular COMA is guaranteed to converge in identical interest games, unlike Softmax Policy Gradient in normal form games, is a consequence of the standard analysis of actor-critic algorithms. In their correctness proof for COMA, Foerster et al. [12] prove that COMA's use of a centralized critic effectively results in an actor-critic update that corresponds to single agent actor-critic updates with a policy parameterized by the parameters of independent actors representing the policy of each individual agent. Thus it is guaranteed to converge to a local optimum of the objective function under standard assumptions for the convergence of actor-critic methods. As the true rewards provide a perfect critic, the same applies to Tabular COMA. Since a local optimum of the shared utility function is a Nash equilibrium, this shows that failure to include the NeuRD fix does not prevent convergence in identical interest games, unlike in normal form games.

updates. In online learning this type of feedback is often referred to as bandit feedback, so we call this variant Bandit COMA-N. It is given in Algorithm 2.

Algorithm 2: Bandit COMA-N update for agent **a**

```
Result: Update policy for agent $a$ given by $\pi^a$ for a state $s$
Sample joint action $\mathbf{u}^{-a}$ from other agents policy $\pi(\mathbf{u}^{-a})$;
Sample action $u$ from current policy of an agent $\pi^a$;
$A^a(s,u) \leftarrow A(s,u,\mathbf{u}^{-a})/\pi^a(u)$
$A^a_{sum}(s,u) \leftarrow A^a_{sum}(s,u) + \eta A^a(s,u)$
$\pi'^a(s) \propto \exp(A^a_{sum}(s))$
$\pi^a(s) \propto (1-\epsilon)\pi'^a(s) + \epsilon/|U|$
```

Since we are working with bandit feedback we do not have access to the term $\sum_{u'} \pi^a(u')r(u', \mathbf{u}^{-a})$ of the Tabular COMA update. However, this term is independent of the choice of u for agent a so simply omitting it has no effect on the policy (this fact is why the way we specify Hedge is equivalent to the standard version).

Just as Tabular COMA-N is equivalent to Hedge, Bandit COMA-N is equivalent to ϵ-Hedge with bandit feedback. This algorithm was analyzed by Cohen et al. [6] in the more general setting of potential games.

Corollary 3. *In a generic identical interest game Bandit COMA-N with exploration parameter $\epsilon > 0$ and a suitable choice of learning rate converges almost surely to a $\delta(\epsilon)$ Nash equilibrium, where $\delta(\epsilon) \longrightarrow 0$ as $\epsilon \longrightarrow 0$. Furthermore, if the approximate equilibrium Bandit COMA-N converges to puts weight ϵ on the uniform strategy and weight $1-\epsilon$ on a pure strategy for each agent then almost surely this pure strategy profile is a (strict) Nash equilibrium and the convergence to it occurs at a quasi-exponential rate.*

Proof. For the bandit setting in a generic identical interest games, the advantage estimate in Algorithm 2 becomes $A(u, \mathbf{u}^{-a}) = r(u, \mathbf{u}^{-a})/\pi^a(u)$. Thus Bandit COMA-N is equivalent to ϵ-Hedge. The convergence of Tabular COMA-N now directly follows from the Theorem 3 by Cohen et al. [6] □

In the theorem, the requirement that the game be generic means that a sufficiently small perturbation of the payoffs does not change the set of Nash equilibria.[5] The quasi exponential rate is much faster than the typical bound of $O(\sqrt{T})$ for the growth of regret. See their paper for the exact bound and requirements on the learning rate. This analysis provides a principled rationale for the use of bounded softmax by COMA with a minimum exploration rate. Also, key to the convergence of ϵ–Hedge algorithm is annealing learning rate which helps in controlling the variance of reward estimate of agent's actions and helps the agent achieve last iterate convergence. The learning rate is an important hyperparameter for stability of NeuRD update in the bandit version of COMA. In our experiments, we therefore anneal the learning rate to achieve stable performance of COMA-N algorithm. Our experiments in stateful settings show that in some cases using bounded softmax significantly improves the performances over the corresponding regular softmax versions of COMA while in few cases it can lead to slower learning and lower performance because of the unnecessary forced exploration. An ablation study also confirms the importance of annealing the learning rate, which the corollary requires.

5 EVALUATION

In this section, we analyze the importance of the two key features we analyzed theoretically—the NeuRD fix and bounded softmax—in stateful settings. This gives us four algorithms: COMA, COMA-N (Algorithm 4 in the appendix with the one line fix highlighted in red), and their variants where the bounded softmax is replaced by a standard softmax Soft-COMA and Soft-COMA-N. As a reminder, our goal is to demonstrate the relevance of our theoretical analysis to COMA, not attain state-of-the-art results and so similar to other works on improving COMA[23, 25], our experiments only include COMA and its variants mentioned above. To implement the NeuRD fix, we use the following update rule for actors.

$$\Delta\theta^a = \Delta\theta^a + [1/\pi^a(u|h^a_t)]\hat{\nabla}_{\theta^a} v^a(\theta^a) A^a(s_t, \mathbf{u}) \tag{9}$$

Equation (9) is the version of the NeuRD policy gradient update (8) for the bandit case in stateful settings. For the implementation of COMA, we use the repository provided by Foerster et al. [12].[6] For COMA-N, we threshold the range of allowable logits using the same implementation as the OpenSpiel implementation of NeuRD[7]. This thresholding is described by Hennes et al. [17] as a way to prevent infinite gradients and our testing confirms that performance without it is poor. For all hyperpameters, we use the values from the repository unless otherwise noted. We present results for Switch, Blocker, and a Multistep Matrix Game from that repository. We also present the results on maps from the StarCraft Multi-Agent Challenge (SMAC)[34].

5.1 Switch

Switch is a small grid world navigation game having two or four agents where each agent wants to reach its own home location with either local or shared observations. The challenge is to coordinate with the other agent(s) to navigate through the narrow corridor which can be used only by one agent at a time. The agents need to coordinate to not block the pathway for the others. A team reward of +5 is given to the agents whenever one of agents reaches its home cell. The episode ends when all agents have reached their home cell or after 100 steps. Our experiments in Figures 2, 3, 4 and 5 show that using NeurRD fix with COMA-N and Soft-COMA-N never leads to worse performance in this game than COMA and Soft-COMA respectively. Further COMA-N leads to better performance on all four versions of the game which shows that this one line fix can lead to adaptive policies without incurring additional costs, consistent with our theoretical results. The story for softmax updates is more mixed. We observe opposite trends in performance of Soft-COMA vs COMA in Figures 3 and 4. Recall that the theoretical advantage of bounded softmax was more stable updates, but this does come at the cost of forced exploration due to ϵ. This also explains the performance of Soft-COMA-N vs COMA-N in Figure 5.

[5] A game can be made generic almost surely by an arbitrarily small perturbation to the payoffs, so this assumption is mild.

[6] https://github.com/oxwhirl/pymarl

[7] open_spiel/open_spiel/python/algorithms/neurd.py

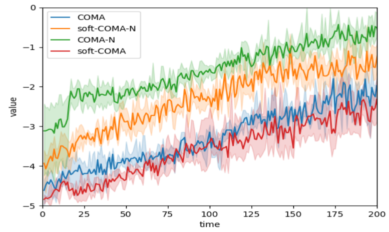

Figure 2: Switch4-v1

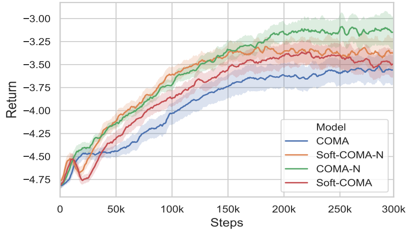

Figure 3: Switch4-v0

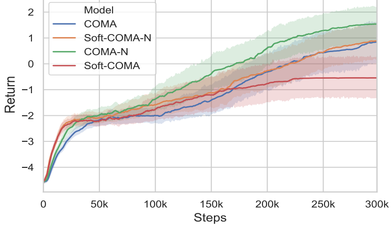

Figure 4: Switch2-v1

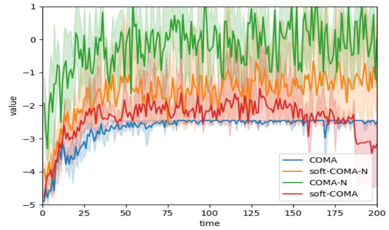

Figure 5: Switch2-v0

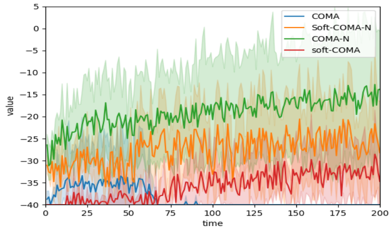

Figure 6: Blocker-v0

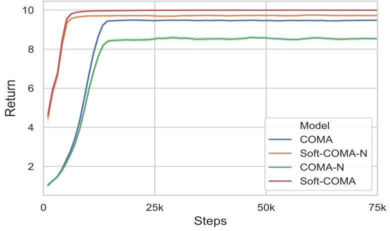

Figure 7: Multistep Matrix Game (v1)

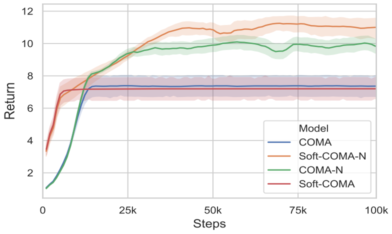

Figure 8: Multistep Matrix Game (v2)

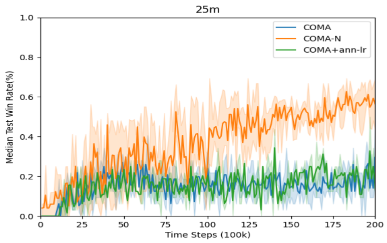

Figure 9: 25m

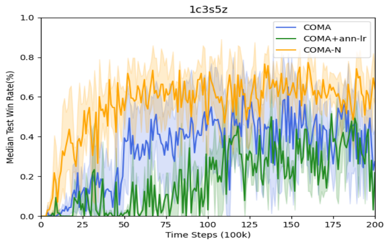

Figure 10: 1c3s5z

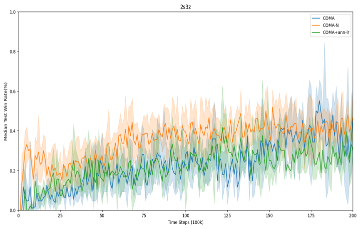

Figure 11: 2s3z

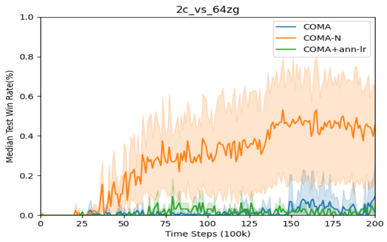

Figure 12: 2c-vs-64zg

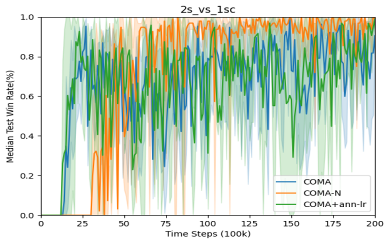

Figure 13: 2s-vs-1sc

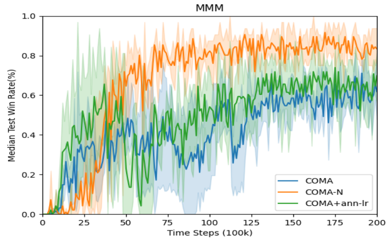

Figure 14: MMM

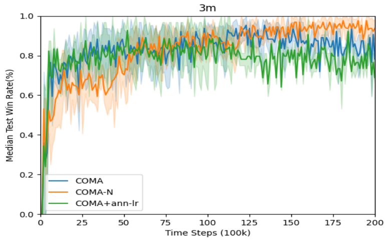

Figure 15: 3m

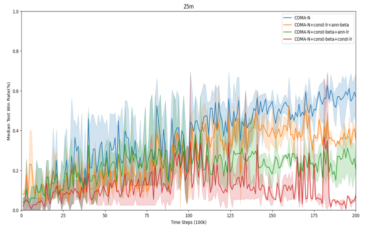

Figure 16: Ablations with respect to the learning rate and clipping coefficient β

5.2 Multistep Matrix Game

We use two versions of the Multistep Matrix game introduced by Yang et al. [46]: (v1) is their original version and (v2) is our variant, shown in Figure 17 in the appendix. The only difference is that the -4s in the upper left matrix were +1s in the original version. This environment is unique and designed to be somewhat pathological since it involves many intermediate terminal states (shown in red) and fewer paths leading to later stages. In (v1), Figure 7 shows a perhaps surprising order of performance. This occurs for two reasons. Both players coordinating on either (U,L) or (D,R) yields a score of 10, and all versions essentially converge to this policy. However, the bounded softmax policies anneal their ϵ parameter down to 0.02 over the first 10K steps[8] After that, each agent can still take its intended action at most 99% of the time, so performance degrades due to this forced exploration. Similarly, our clipping of the logits using $\beta = 2$ also puts a limit on the probability assigned to the desired action. These limits entirely explain the performance differences; in a sample run after 25K steps Soft-COMA took its desired action in the start state 100% of the time, Soft-COMA-N 99.45%, COMA 98.62%, and COMA-N 97.98%. In (v2), agents who coordinate on (U,L) now have a stronger incentive to learn to play (D,R) in the final stage, which only the more adaptive COMA-N variants do. Again, the lack of forced exploration makes softmax policies slightly better.

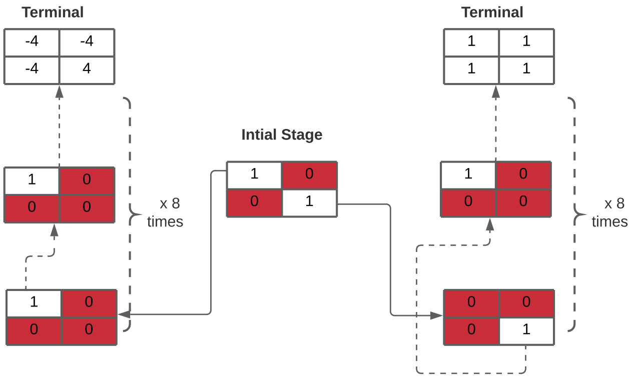

Figure 17: Illustration of Multistep Matrix Game (v2)

5.3 Blocker

This game requires one of a team of three agents to reach the last row by setting up a situation where moving blockers can only stop two of them. The agents receive -1 reward per time-step before they all reach the destination. The highest reward of the game varies from -6 and -3 depending on the starting points. The agents only have access to decentralized policies and local observations. Figure 6 shows that soft-COMA and COMA eventually lead to the worst performance, but Soft-COMA-N and COMA-N continue to improve. Thus in this setting, the use of bounded softmax value with the NeuRD fix leads to overall best performance while just using NeuRD with softmax outperforms the ones without the NeuRD fix.

5.4 StarCraft Multiagent Challenge

SMAC is built on the popular real-time strategy game StarCraft II. SMAC is meant for training the agents on micromanagement tasks in decentralized fashion. This introduces challenges like partial observability, decentralized execution, credit assignment and value assignment for joint actions. COMA-N outperforms COMA when both algorithms are run for 5 independent runs and improves COMA's performance on hard RL environments. We consider the following maps in our experiments: 2s_vs_1sc, 2c_vs_64zg, 2s3z, 1c3s5z, 3m, 2s3z, 25m and MMM. (COMA is known to not learn on many other maps and the COMA-N improvements are not sufficient to fix this.) To stabilize the NeuRD policy gradient, we tuned the initial logit clipping parameter β for each map from the set $\{2.5, 5, 7.5, 10, 15, 20, 25, 30,$
$35, 40, 45, 50\}$. Additionally, we linearly decay β to a value in $\{\beta/2 - 2.5, \beta/2, \beta/2 + 2.5\}$ over $150K$ iterations. We also linearly annealed the value of the actor's learning rate to $1/5$ or $1/10$ of its initial value over the first $150K$ iterations to stabilize training[9]. The training curves in Figures 9-15 show that the resulting algorithm generally improves performance, most notably on the harder map 2c-vs-64zg where COMA is known to perform poorly. To confirm that our results are due to the COMA-N and not the annealing of the learning rate, each plot also includes a version of COMA with this feature added; we found it had no significant effect.

Finally, to analyze the effect of annealing the clipping coefficient and the learning rate on COMA-N's performance, we performed an ablation study with $\beta = 5$ and $lr_s/lr_e = 5$ (annealing the learning rate to 1/5th of its original value) on the more challenging SMAC $25m$ map. We found that clipping coefficient has higher impact on the overall performance than the learning rate and that the lack of both has a substantial negative impact.

6 CONCLUSION

We provided a new justification for COMA's update rule by connecting it to regret minimization in identical interest games. Based on this we showed that COMA should apply the NeuRD fix and provided a justification for COMA's use of a bounded softmax policy. We demonstrated the efficacy of COMA-N on variety of environments including StarCraft where it consistently outperformed COMA and was able to learn on the harder map 2c-vs-64zg where COMA fails. Complementary to prior work on understanding theoretical foundation of Value Decomposition methods, our works add to the current understanding of COMA-like Actor-Critic Methods in cooperative settings.

REFERENCES

[1] Ron Amit, Ron Meir, and Kamil Ciosek. 2020. Discount factor as a regularizer in reinforcement learning. In *International conference on machine learning*. PMLR, 269–278.

[2] Raphaël Avalos, Mathieu Reymond, Ann Nowé, and Diederik M Roijers. 2021. Local Advantage Networks for Cooperative Multi-Agent Reinforcement Learning. *arXiv preprint arXiv:2112.12458* (2021).

[3] Yu Bai, Chi Jin, Song Mei, Ziang Song, and Tiancheng Yu. 2022. Efficient Φ-Regret Minimization in Extensive-Form Games via Online Mirror Descent. *arXiv preprint arXiv:2205.15294* (2022).

[8] Yang et al. [46] anneal over a longer period and so report slower learning for COMA than with our tuning.

[9] This method stabilizing learning by annealing the learning rate and clipping coefficient is much simpler than the one used by original NeuRD paper Hennes et al. [17] to where an entropy term was added to the Q-values for the critic and then the equilibrium point of the game recentered after every iteration of the game.

[4] Daniel S Bernstein, Robert Givan, Neil Immerman, and Shlomo Zilberstein. 2002. The complexity of decentralized control of Markov decision processes. *Mathematics of operations research* 27, 4 (2002), 819–840.

[5] Yongcan Cao, Wenwu Yu, Wei Ren, and Guanrong Chen. 2012. An overview of recent progress in the study of distributed multi-agent coordination. *IEEE Transactions on Industrial informatics* 9, 1 (2012), 427–438.

[6] Johanne Cohen, Amélie Héliou, and Panayotis Mertikopoulos. 2017. Learning with bandit feedback in potential games. In *Proceedings of the 31th International Conference on Neural Information Processing Systems*.

[7] Pierre Coucheney, Bruno Gaujal, and Panayotis Mertikopoulos. 2015. Penalty-regulated dynamics and robust learning procedures in games. *Mathematics of Operations Research* 40, 3 (2015), 611–633.

[8] Zehao Dou, Jakub Grudzien Kuba, and Yaodong Yang. 2022. Understanding Value Decomposition Algorithms in Deep Cooperative Multi-Agent Reinforcement Learning. *arXiv preprint arXiv:2202.04868* (2022).

[9] Eyal Even-Dar, Sham M Kakade, and Yishay Mansour. 2009. Online Markov decision processes. *Mathematics of Operations Research* 34, 3 (2009), 726–736.

[10] Gabriele Farina, Christian Kroer, and Tuomas Sandholm. 2019. Regret circuits: Composability of regret minimizers. In *International conference on machine learning*. PMLR, 1863–1872.

[11] Gabriele Farina, Chung-Wei Lee, Haipeng Luo, and Christian Kroer. 2022. Kernelized Multiplicative Weights for 0/1-Polyhedral Games: Bridging the Gap Between Learning in Extensive-Form and Normal-Form Games. *arXiv preprint arXiv:2202.00237* (2022).

[12] Jakob N Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. 2018. Counterfactual multi-agent policy gradients. In *Thirty-second AAAI conference on artificial intelligence*.

[13] Yoav Freund and Robert E Schapire. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences* 55, 1 (1997), 119–139.

[14] Yoav Freund and Robert E Schapire. 1999. Adaptive game playing using multiplicative weights. *Games and Economic Behavior* 29, 1-2 (1999), 79–103.

[15] Wei Fu, Chao Yu, Zelai Xu, Jiaqi Yang, and Yi Wu. 2022. Revisiting some common practices in cooperative multi-agent reinforcement learning. *arXiv preprint arXiv:2206.07505* (2022).

[16] Sergiu Hart and Andreu Mas-Colell. 2000. A simple adaptive procedure leading to correlated equilibrium. *Econometrica* 68, 5 (2000), 1127–1150.

[17] Daniel Hennes, Dustin Morrill, Shayegan Omidshafiei, Rémi Munos, Julien Perolat, Marc Lanctot, Audrunas Gruslys, Jean-Baptiste Lespiau, Paavo Parmas, Edgar Duéñez-Guzmán, et al. 2020. Neural Replicator Dynamics: Multiagent Learning via Hedging Policy Gradients. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*. 492–501.

[18] Jianhong Jianhong Wang, Yuan Zhang, Tae-Kyun Kim, and Yunjie Gu. 2020. Shapley Q-value: A Local Reward Approach to Solve Global Reward Games. (2020).

[19] Peter Jin, Kurt Keutzer, and Sergey Levine. 2018. Regret minimization for partially observable deep reinforcement learning. In *International Conference on Machine Learning*. 2342–2351.

[20] Ian A Kash, Lev Reyzin, and Zishun Yu. 2022. Slowly Changing Adversarial Bandit Algorithms are Provably Efficient for Discounted MDPs. *arXiv preprint arXiv:2205.09056* (2022).

[21] Ian A Kash, Michael Sullins, and Katja Hofmann. 2019. Combining No-regret and Q-learning. *arXiv preprint arXiv:1910.03094* (2019).

[22] Robert Kleinberg, Georgios Piliouras, and Eva Tardos. 2009. Multiplicative updates outperform generic no-regret learning in congestion games. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*. 533–542.

[23] Jakub Grudzien Kuba, Muning Wen, Linghui Meng, Haifeng Zhang, David Mguni, Jun Wang, Yaodong Yang, et al. 2021. Settling the variance of multi-agent policy gradients. *Advances in Neural Information Processing Systems* 34 (2021), 13458–13470.

[24] Shuxin Li, Youzhi Zhang, Xinrun Wang, Wanqi Xue, and Bo An. 2021. CFR-MIX: Solving imperfect information extensive-form games with combinatorial action space. *arXiv preprint arXiv:2105.08440* (2021).

[25] Yueheng Li, Guangming Xie, and Zongqing Lu. 2022. Difference advantage estimation for multi-agent policy gradients. In *International Conference on Machine Learning*. PMLR, 13066–13085.

[26] Jason R Marden, Gürdal Arslan, and Jeff S Shamma. 2007. Regret based dynamics: convergence in weakly acyclic games. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*. 1–8.

[27] Ruta Mehta, Ioannis Panageas, and Georgios Piliouras. 2015. Natural selection as an inhibitor of genetic diversity: Multiplicative weights updates algorithm and a conjecture of haploid genetics [working paper abstract]. In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science*. 73–73.

[28] Angelia Nedic and Soomin Lee. 2014. On stochastic subgradient mirror-descent algorithm with weighted averaging. *SIAM Journal on Optimization* 24, 1 (2014), 84–107.

[29] Duc Thien Nguyen, Akshat Kumar, and Hoong Chuin Lau. 2018. Credit assignment for collective multiagent RL with global rewards. In *Advances in Neural Information Processing Systems*. 8102–8113.

[30] Gerasimos Palaiopanos, Ioannis Panageas, and Georgios Piliouras. 2017. Multiplicative weights update with constant step-size in congestion games: Convergence, limit cycles and chaos. In *Advances in Neural Information Processing Systems*. 5872–5882.

[31] Bei Peng, Tabish Rashid, Christian Schroeder de Witt, Pierre-Alexandre Kamienny, Philip Torr, Wendelin Böhmer, and Shimon Whiteson. 2021. Facmac: Factored multi-agent centralised policy gradients. *Advances in Neural Information Processing Systems* 34 (2021), 12208–12221.

[32] Silviu Pitis. 2019. Rethinking the discount factor in reinforcement learning: A decision theoretic approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 7949–7956.

[33] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. 2018. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *International conference on machine learning*. PMLR, 4295–4304.

[34] Mikayel Samvelyan, Tabish Rashid, Christian Schroeder De Witt, Gregory Farquhar, Nantas Nardelli, Tim GJ Rudner, Chia-Man Hung, Philip HS Torr, Jakob Foerster, and Shimon Whiteson. 2019. The starcraft multi-agent challenge. *arXiv preprint arXiv:1902.04043* (2019).

[35] Kyunghwan Son, Sungsoo Ahn, Roben Delos Reyes, Jinwoo Shin, and Yung Yi. 2020. QOPT: Optimistic Value Function Decentralization for Cooperative Multi-Agent Reinforcement Learning. *arXiv preprint arXiv:2006.12010* (2020).

[36] Kyunghwan Son, Daewoo Kim, Wan Ju Kang, David Earl Hostallero, and Yung Yi. 2019. Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *International conference on machine learning*. PMLR, 5887–5896.

[37] Sriram Srinivasan, Marc Lanctot, Vinicius Zambaldi, Julien Pérolat, Karl Tuyls, Rémi Munos, and Michael Bowling. 2018. Actor-critic policy optimization in partially observable multiagent environments. In *Advances in neural information processing systems*. 3422–3435.

[38] Jianyu Su, Stephen Adams, and Peter Beling. 2021. Value-decomposition multi-agent actor-critics. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 11352–11360.

[39] Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, et al. 2017. Value-decomposition networks for cooperative multi-agent learning. *arXiv preprint arXiv:1706.05296* (2017).

[40] Kagan Tumer and Adrian Agogino. 2007. Distributed agent-based air traffic flow management. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*. 1–8.

[41] Jianhao Wang, Zhizhou Ren, Beining Han, Jianing Ye, and Chongjie Zhang. 2021. Towards understanding cooperative multi-agent q-learning with value factorization. *Advances in Neural Information Processing Systems* 34 (2021), 29142–29155.

[42] Jianhao Wang, Zhizhou Ren, Terry Liu, Yang Yu, and Chongjie Zhang. 2020. Qplex: Duplex dueling multi-agent q-learning. *arXiv preprint arXiv:2008.01062* (2020).

[43] David H Wolpert and Kagan Tumer. 2002. Optimal payoff functions for members of collectives. In *Modeling complexity in economic and social systems*. World Scientific, 355–369.

[44] Jiachen Yang, Alireza Nakhaei, David Isele, Kikuo Fujimura, and Hongyuan Zha. 2018. Cm3: Cooperative multi-goal multi-stage multi-agent reinforcement learning. *arXiv preprint arXiv:1809.05188* (2018).

[45] Yaodong Yang, Jianye Hao, Ben Liao, Kun Shao, Guangyong Chen, Wulong Liu, and Hongyao Tang. 2020. Qatten: A general framework for cooperative multiagent reinforcement learning. *arXiv preprint arXiv:2002.03939* (2020).

[46] Yaodong Yang, Ying Wen, Lihuan Chen, Jun Wang, Kun Shao, David Mguni, and Weinan Zhang. 2020. Multi-Agent Determinantal Q-Learning. *arXiv preprint arXiv:2006.01482* (2020).

[47] Dayong Ye, Minjie Zhang, and Yun Yang. 2015. A multi-agent framework for packet routing in wireless sensor networks. *sensors* 15, 5 (2015), 10026–10047.

[48] Wang Ying and Sang Dayong. 2005. Multi-agent framework for third party logistics in E-commerce. *Expert Systems with Applications* 29, 2 (2005), 431–436.

[49] Shangtong Zhang and Richard S Sutton. 2017. A deeper look at experience replay. *arXiv preprint arXiv:1712.01275* (2017).

[50] Shangtong Zhang, Hengshuai Yao, and Shimon Whiteson. 2021. Breaking the deadly triad with a target network. In *International Conference on Machine Learning*. PMLR, 12621–12631.

[51] Tianhao Zhang, Yueheng Li, Chen Wang, Guangming Xie, and Zongqing Lu. 2021. Fop: Factorizing optimal joint policy of maximum-entropy multi-agent reinforcement learning. In *International Conference on Machine Learning*. PMLR, 12491–12500.

[52] Meng Zhou, Ziyu Liu, Pengwei Sui, Yixuan Li, and Yuk Ying Chung. 2020. Learning Implicit Credit Assignment for Multi-Agent Actor-Critic. *arXiv preprint arXiv:2007.02529* (2020).

[53] Martin Zinkevich, Michael Johanson, Michael Bowling, and Carmelo Piccione. 2008. Regret minimization in games with incomplete information. In *Advances*

in neural information processing systems. 1729–1736.

A COMA VS COMA-N

Here we provide a pseudocode comparison of COMA and COMA-N with the one line fix highlighted in red.

Algorithm 3: COMA update for agent **a**

```
Result: Update policy for agent $a$ give by $\pi^a$
Given the action of other agents $\mathbf{u}_t^{-a}$, play action $u_t^a$ based
  on agent's history $h_t^a$ for each agent $u_t^a$ and get reward $r_t$
  and next state $s_{t+1}$
for $t$ = 1 to $T$ do
    Calculate the target network values $y_t$ using the critic
      parameter of the target $\hat{\theta}_t^c$ while batch unrolling states,
      actions and rewards.
end
for $t$ = $T$ to 1 do
    $\Delta\theta^c = \nabla_{\theta^c}(y_t - Q(s_t, \mathbf{u}))^2$
    $\Delta\theta^c \leftarrow \theta^c - \alpha\Delta\theta^c$
    Every C steps reset $\hat{\theta}_t^c = \theta_t^c$
end
for $t$ = 1 to $T$ do
    $A^a(s_t, \mathbf{u}) = Q(s_t, \mathbf{u}) - \sum_u Q(s_t, u, \mathbf{u}^{-a})\pi^a(u|h_t^a)$
    $\Delta\theta^q = \Delta\theta^a + \nabla_{\theta^a} \log \pi^a(u|h_t^a) A^a(s_t^a, \mathbf{u})$
end
$\theta_{t+1}^a = \theta_t^a + \alpha\Delta\theta^a$
```

Algorithm 4: COMA-N update for agent **a**

```
Result: Update policy for an agent $a$ given by $\pi^a$ with
          policy network having logits $\upsilon^a$
Given the action of other agents $\mathbf{u}_t^{-a}$, play action $u_t^a$ based
  on agent's history $h_t^a$ for each agent $u_t^a$ and get reward $r_t$
  and next state $s_{t+1}$
for $t$ = 1 to $T$ do
    Calculate the target network values $y_t$ using the critic
      parameter of the target $\hat{\theta}_t^c$ while batch unrolling states,
      actions and rewards.
end
for $t$ = $T$ to 1 do
    $\Delta\theta^c = \nabla_{\theta^c}(y_t - Q(s_t, \mathbf{u}))^2$
    $\Delta\theta^c \leftarrow \theta^c - \alpha\Delta\theta^c$
    Every C steps reset $\hat{\theta}_t^c = \theta_t^c$
end
for $t$ = 1 to $T$ do
    $A^a(s_t, \mathbf{u}) = Q(s_t, \mathbf{u}) - \sum_u Q(s_t, u, \mathbf{u}^{-a})\pi^a(u|h_t^a)$
    $\Delta\theta^a = \Delta\theta^a + [1/\pi^a(u|h_t^a)]\hat{\nabla}_{\theta^a}\upsilon^a(\theta^a)A^a(s_t, \mathbf{u})$
end
$\theta_{t+1}^a = \theta_t^a + \alpha\Delta\theta^a$
```

B ADDITIONAL EVALUATION DETAILS

B.0.1 StarCraft Multiagent Challenge. We show the parameters for each map in Table 2. As mentioned earlier, the values of β and β_0 denote the starting and ending value of clipping coefficient respectively. The values lr_s and lr_e denote the starting and the ending values of the actor learning rate respectively and the ratio lr_s/lr_e in the table value denote the extent to which we annealed the actor learning rate. For example, the value of 10 indicates that the value of actor learning is annealed to $1/10$ of its value. We start lr_s with value of original COMA algorithm which is 0.0005.

Map name	lr_s/lr_e	β	β_0
3m	1	5	5
25m	5	5	5
2s3z	5	45	25
1c3s5z	5	3	1.5
MMM	10	10	5
2c_vs_64zg	5	45	25
2s_vs_1sc	10	10	5

Table 2: Hyper parameters for SMAC experiments

It can be observed from the table that the maps where COMA doesn't perform well like $2s3z$ and $2c\_vs\_64zg$ (performance < 0.5) and so consequently COMA-N requires relatively higher (> 20) clipping coefficient and/or relatively lower annealing of actor learning rate. This is intuitive as difficult maps require higher and faster jumps in policy update equation which can be achieved either via learning rate or via higher value of clipping coefficient. On the other hand, for the maps like $3m$, $1c3s5z$, MMM and $2s\_vs\_1sc$ where COMA is performance is relatively better, COMA-N requires higher annealing and relatively lower value of clipping coefficient to avoid overfitting the policy network. These observations can be used when training COMA-N for any other maps. Additionally, some maps like $3m$ and $25m$ (see Figure 9) don't require annealing of β, especially when the value of clipping coefficient is already small.