NETWORKED COMMUNICATION FOR DECENTRALISED AGENTS IN MEAN-FIELD GAMES

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce networked communication to the mean-field game framework, in particular to oracle-free settings where N decentralised agents learn along a single, non-episodic run of the empirical system. We prove that our architecture has sample guarantees bounded between those of the centralised- and independent-learning cases. We provide the order of the difference in these bounds in terms of network structure and number of communication rounds, and also contribute a policy-update stability guarantee. We discuss how the sample guarantees of the three theoretical algorithms do not actually result in practical convergence. We therefore show that in practical settings where the theoretical parameters are not observed (leading to poor estimation of the Q-function), our communication scheme significantly accelerates convergence over the independent case (and sometimes even the centralised case), without relying on the assumption of a centralised learner. We contribute further practical enhancements to all three theoretical algorithms, allowing us to present their first empirical demonstrations. Our experiments confirm that we can remove several of the theoretical assumptions of the algorithms, and display the empirical convergence benefits brought by our new networked communication. We additionally show that the networked approach has significant advantages, over both the centralised and independent alternatives, in terms of robustness to unexpected learning failures and to changes in population size.

1 INTRODUCTION

The mean-field game (MFG) framework (Lasry & Lions, 2007; Huang et al., 2006) has been used to address the difficulty faced by multi-agent reinforcement learning (MARL) regarding computational scalability as the number of agents increases. It models a representative agent as interacting not with the other individuals in the population on a per-agent basis, but instead with a distribution of other agents, known as the *mean field*. The MFG framework analyses the limiting case when the population consists of an infinite number of symmetric and anonymous agents, that is, they have identical reward and transition functions which depend on the mean-field distribution rather than on the actions of specific other players. In this work we focus on MFGs with stationary population distributions ('stationary MFGs', where learning is more tractable than in non-stationary ones) (Xie et al., 2021; Anahtarci et al., 2023; Zaman et al., 2023; Yardim et al., 2023), for which the solution concept is the MFG-Nash equilibrium (MFG-NE), which reflects the situation when each agent responds optimally to the population distribution that arises when all other agents follow that same optimal behaviour. The MFG-NE can be used as an approximation for the Nash equilibrium (NE) in a finite-agent game, with the error in the solution reducing as the number of agents N tends to infinity (Anahtarci et al., 2023; Saldi et al., 2018; Yardim et al., 2024; Toumi et al., 2024; Hu & Zhang, 2024). MFGs have therefore been used to find approximate solutions for a wide variety of real-world problems involving a large but finite number of agents, which might otherwise have been too difficult to solve; see Appx. G for further details.

For large, complex many-agent systems in the real world (e.g. swarm robotics, autonomous vehicle traffic), it may be infeasible to find MFG-NEs analytically or via oracles/simulations of an infinite population (as they have been traditionally), such that learning must instead be conducted directly by the original finite population in its deployed environment. In such settings, in contrast to many previous methods, desirable qualities for MFG algorithms include: learning from the empirical distribution of N agents (without generation/manipulation of this distribution by the algorithm itself or by an external oracle); learning from a single continued system run that is not arbitrarily reset as in episodic learning; model-free learning; decentralisation; fast practical convergence; and robustness to unexpected failures of decentralised learners or changes in population size (Korecki et al., 2023).

Conversely, MFG frameworks have traditionally been largely theoretical, and methods for finding equilibria have often relied on assumptions that are too strong for real-world applications (Anahtarci et al., 2023; Lauriere et al., 2022; Perrin et al., 2020; Laurière et al., 2022; Guo et al., 2019a; Perrin et al., 2021; Elie et al., 2020; Carmona & Laurière, 2021; Cao et al., 2020; Germain et al., 2022; Fouque & Zhang, 2020; Algumaei et al., 2023; Angiuli et al., 2023); see Appx. G for an extended discussion of this related work. In particular, almost all prior work relies on a centralised controller to orchestrate the learning of all agents (Xie et al., 2021; Anahtarci et al., 2023; Zaman et al., 2023; Laurière et al., 2022; Guo et al., 2019b). However, outside of MFGs, the multi-agent systems community has recognised that the existence of a central controller is a very strong assumption, and one that can both restrict scalability by constituting a bottleneck for computation and communication, and reveal a single point of failure for the whole system (Zhang et al., 2021b; 2018; Wai et al., 2018; Zhang et al., 2021a; Chen et al., 2021; Jiang et al., 2024). For example, if the single server coordinating all of a smart city's autonomous vehicles were to crash, the entire road network would cease to operate. As an alternative, recent work has explored MFG algorithms for independent learning (Yardim et al., 2023; Mguni et al., 2018; Yongacoglu et al., 2022a;b; Grammatico et al., 2015a;b; Parise et al., 2015; Grammatico et al., 2016). However, prior works focus on theoretical sample guarantees instead of practical convergence speed, and have largely not considered robustness in the senses we address, despite fault-tolerance being an original motivation behind many-agent systems.

We address *all* of these desiderata by novelly introducing a communication network to the MFG setting. Communication networks have had success in other multi-agent settings, removing the reliance on inflexible, centralised structures (Zhang et al., 2021b; Wai et al., 2018; Zhang et al., 2021a; Chen et al., 2021; Doan et al., 2019; Lin et al., 2019; Heredia et al., 2020; Kar et al., 2013; Suttle et al., 2020). We focus on 'coordination games', i.e. where agents can increase their individual rewards by following the same strategy as others and therefore have an incentive to communicate policies, even if the MFG setting itself is technically non-cooperative. Thus our work can be applied to real-world problems in e.g. traffic signal control, formation control in swarm robotics, and consensus and synchronisation e.g. for sensor networks.

In this work, we show that when the agents' state-action value functions (Q-functions) can be only roughly estimated due to fewer samples/updates, possibly leading to high variance in policy updates, then propagating policies that are estimated to be better through the population via the communication network leads to faster convergence than that achieved by agents learning entirely independently. This is crucial in large complex environments that may be encountered in real applications, where the idealised hyperparameter choices (such as learning rates and numbers of iterations) required in previous works for theoretical convergence guarantees will be infeasible in practice. We compare our networked architecture with modified versions of earlier theoretical algorithms for the centralised and independent settings; we extend the original algorithms with experience replay buffers, without which we found them unable to demonstrate any learning in practical time. While the use of buffers means that the original theoretical sample guarantees no longer apply, we argue that this is preferable since these guarantees were in any case impractical. On this basis, we conduct numerical comparisons of the three architectures, demonstrating the benefits of communication for both convergence speed and system robustness. For further discussion of how networked communication can benefit robustness in large multi-agent systems, see Appx. E. In summary, our key contributions include the following:

- We prove that a *theoretical* version of our new networked algorithm (Alg. 1) has sample guarantees bounded between those of the centralised and independent settings for learning with a single, non-episodic run of the empirical system. We provide the order of the difference in these bounds in terms of network structure and number of communication rounds, and also contribute a policy-update stability guarantee (Sec. 3.3 and Appx. B.8).
- All three theoretical algorithms do not permit any learning in practical time; we show that in *practical* settings our communication scheme can significantly benefit convergence speed over the independent case, and sometimes even the centralised case (Sec. 3.4.1).
- We novelly modify all three theoretical algorithms (Alg. 2) to make their practical convergence feasible, most notably by including an experience replay buffer, allowing us to

contribute the first empirical demonstrations of all three algorithms (Sec. 3.4.2). An ablation study of the replay buffer is given in Appx. F.4 - agents do not seem to learn at all without it.

- Our experiments demonstrate the convergence benefits brought by our networked communication, and show we can remove several of the algorithms' theoretical assumptions (a goal shared by other work on the practicality of MFG algorithms (Cui et al., 2024)) (Sec. 4.1).
- We further demonstrate that our decentralised communication architecture brings significant benefits over both the centralised and independent alternatives in terms of robustness to unexpected learning failures and changes in population size (Sec. 4.1 and Appx. F.4).

The main paper is structured as follows: notation and preliminaries are given in Sec. 2; the theoretical algorithms and results are presented in Secs. 3.1-3.3; practical enhancements to the algorithms are given in Sec. 3.4; and experiments and discussion are provided in Sec. 4. Limitations and future work are found in Appx. H, and the broader social impact is considered in Sec. 5.

2 Preliminaries

We use the following notation. N is the number of agents in a population, with $\mathcal{S}$ and $\mathcal{A}$ representing the finite state and common action spaces, respectively. The sets $\mathcal{S}$ and $\mathcal{A}$ are equipped with the discrete metric $d(x, y) = \mathbb{1}_{x\neq y}$. The set of probability measures on a finite set $\mathcal{X}$ is denoted $\Delta_{\mathcal{X}}$, and $\mathbf{e}_x \in \Delta_{\mathcal{X}}$ for $x \in \mathcal{X}$ is a one-hot vector with only the entry corresponding to x set to 1, and all others set to 0. For time $t \geq 0$, $\hat{\mu}_t = \frac{1}{N}\sum_{i=1}^{N}\sum_{s\in\mathcal{S}} \mathbb{1}_{s_t^i=s}\mathbf{e}_s \in \Delta_{\mathcal{S}}$ is a vector denoting the empirical state distribution of the N agents at time t. The set of policies is $\Pi = \{\pi : \mathcal{S} \to \Delta_{\mathcal{A}}\}$, and the set of Q-functions is denoted $\mathcal{Q} = \{q : \mathcal{S} \times \mathcal{A} \to \mathbb{R}\}$. For $\pi, \pi' \in \Pi$ and $q, q' \in \mathcal{Q}$, we have the norms $||\pi - \pi'||_1 := \sup_{s\in\mathcal{S}} ||\pi(s) - \pi'(s)||_1$ and $||q - q'||_\infty := \sup_{s\in\mathcal{S},a\in\mathcal{A}} |q(s, a) - q'(s, a)|$.

Function $h : \Delta_{\mathcal{A}} \to \mathbb{R}_{\geq 0}$ denotes a strongly concave function, which we implement as the scaled entropy regulariser $\lambda h_{ent}(u) = -\lambda\sum_a u(a)\log u(a)$, for $a \in \mathcal{A}$, $u \in \Delta_{\mathcal{A}}$ and $\lambda > 0$. As in some earlier works (Anahtarci et al., 2023; Yardim et al., 2023; Algumaei et al., 2023; Cui & Koeppl, 2021; Guo et al., 2022; Yu & Yuan, 2023), regularisation is theoretically required to ensure the contractivity of operators and continued exploration, and hence algorithmic convergence. However, it has been recognised that modifying the RL objective in this way can bias the NE (Yardim et al., 2023; Hu & Zhang, 2024; Lauriere et al., 2022; Su & Lu, 2022). We show in our experiments that we are able to reduce λ to 0 with no detriment to convergence.

Definition 1 (N-player symmetric anonymous games)**.** *An N-player stochastic game with symmetric, anonymous agents is given by the tuple $\langle N, \mathcal{S}, \mathcal{A}, P, R, \gamma\rangle$, where $\mathcal{A}$ is the action space, identical for each agent; $\mathcal{S}$ is the identical state space of each agent, such that their initial states are $\{s_0^i\}_{i=1}^N \in \mathcal{S}^N$ and their policies are $\{\pi^i\}_{i=1}^N \in \Pi^N$. $P : \mathcal{S} \times \mathcal{A} \times \Delta_{\mathcal{S}} \to \Delta_{\mathcal{S}}$ is the transition function and $R : \mathcal{S} \times \mathcal{A} \times \Delta_{\mathcal{S}} \to [0,1]$ is the reward function, which map each agent's local state and action and the population's empirical distribution to transition probabilities and bounded rewards, respectively, i.e.*

$$s_{t+1}^i \sim P(\cdot|s_t^i, a_t^i, \hat{\mu}_t), \quad r_t^i = R(s_t^i, a_t^i, \hat{\mu}_t), \quad \forall i = 1, \dots, N.$$

The policy of an agent is given by $a_t^i \sim \pi^i(s_t^i)$, that is, each agent only observes its own state, and not the joint state or empirical distribution of the population.

Definition 2 (N-player discounted regularised return)**.** *With joint policies $\boldsymbol{\pi} := (\pi^1, \dots, \pi^N) \in \Pi^N$, initial states sampled from a distribution $\upsilon_0 \in \Delta_{\mathcal{S}}$ and $\gamma \in [0,1)$ as a discount factor, the expected discounted regularised returns of each agent i in the symmetric anonymous game are given by*

$$\Psi_h^i(\boldsymbol{\pi}, \upsilon_0) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t (R(s_t^i, a_t^i, \hat{\mu}_t) + h(\pi^i(s_t^i)))\middle|_{\substack{s_0^j \sim \upsilon_0 \\ a_t^j \sim \pi^j(s_t^j) \\ s_{t+1}^j \sim P(\cdot|s_t^j, a_t^j, \hat{\mu}_t)}}, \forall t \geq 0, j \in \{1, \dots, N\}\right].$$

Definition 3 (δ-NE)**.** *Say $\delta > 0$ and $(\pi, \boldsymbol{\pi}^{-i}) := (\pi^1, \dots \pi^{i-1}, \pi, \pi^{i+1}, \dots, \pi^N) \in \Pi^N$. An initial distribution $\upsilon_0 \in \Delta_{\mathcal{S}}$ and an N-tuple of policies $\boldsymbol{\pi} := (\pi^1, \dots, \pi^N) \in \Pi^N$ form a δ-NE ($\boldsymbol{\pi}$, υ_0) if*

$$\Psi_h^i(\boldsymbol{\pi}, \upsilon_0) \geq \max_{\pi\in\Pi} \Psi_h^i((\pi, \boldsymbol{\pi}^{-i}), \upsilon_0) - \delta \quad \forall i = 1, \dots, N.$$

At the limit as $N \to \infty$, the population of infinitely many agents can be characterised as a limit distribution $\mu \in \Delta_{\mathcal{S}}$. We denote the expected discounted return of the representative agent in the infinite-agent game - termed an MFG - as V, rather than Ψ as in the finite N-agent case.

Definition 4 (Mean-field discounted regularised return). *For a policy-population pair* $(\pi, \mu) \in \Pi \times \Delta_{\mathcal{S}}$*,*

$$V_h(\pi, \mu) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t (R(s_t, a_t, \mu) + h(\pi(s_t)))\middle|_{\substack{s_0 \sim \mu \\ a_t \sim \pi(s_t) \\ s_{t+1} \sim P(\cdot|s_t, a_t, \mu)}}\right].$$

A stationary MFG is one that has a unique population distribution that is stable with respect to a given policy, and the agents' policies are not time- or population-dependent.

Definition 5 (NE of stationary MFG). *For a policy* $\pi^* \in \Pi$ *and a population distribution* $\mu^* \in \Delta_{\mathcal{S}}$*, the pair* (π^*, μ^*) *is a stationary MFG-NE if the following optimality and stability conditions hold:*

$$\textit{optimality:} \quad V_h(\pi^*, \mu^*) = \max_{\pi} V_h(\pi, \mu^*),$$
$$\textit{stability:} \quad \mu^*(s) = \sum_{s', a'} \mu^*(s') \pi^*(a'|s') P(s|s', a', \mu^*).$$

If the optimality condition is only satisfied with $V_h(\pi^*_\delta, \mu^*_\delta) \geq \max_\pi V_h(\pi, \mu^*_\delta) - \delta$*, then* $(\pi^*_\delta, \mu^*_\delta)$ *is a* δ*-NE of the MFG, where* μ^*_δ *is obtained from the stability equation and* π^*_δ*.*

The MFG-NE is an approximate NE of the finite N-player game in which we may have originally been interested but which is difficult to solve in itself (Yardim et al., 2023; Lauriere et al., 2022):

Proposition 1 (N-player NE and MFG-NE (Thm. 1, (Anahtarci et al., 2023))). *If* (π^*, μ^*) *is a MFG-NE, then, under certain Lipschitz conditions (Anahtarci et al., 2023), for any* $\delta > 0$*, there exists* $N(\delta) \in \mathbb{N}_{>0}$ *such that, for all* $N \geq N(\delta)$*, the joint policy* $\boldsymbol{\pi} = \{\pi^*, \pi^*, \ldots, \pi^*\} \in \Pi^N$ *is a* δ*-NE of the* N*-player game.*

Remark 1. *It can be shown that* δ *can be characterised further in terms of* N*, with* (π^*, μ^*) *being an* $\mathcal{O}(\frac{1}{\sqrt{N}})$*-NE of the* N*-player symmetric anonymous game (Yardim et al., 2023).*

For our new, networked learning algorithm, we also introduce the concept of a time-varying communication network, where the links between agents that make up the network may change at each time step t. Most commonly we might think of such a network as depending on the spatial locations of decentralised agents, such as physical robots, which can communicate with neighbours that fall within a given broadcast radius. When the agents move in the environment, their neighbours and therefore communication links may change. However, the dynamic network can also depend on other factors that may or may not depend on each agent's state s^i_t. For example, even a network of fixed-location agents can change depending on which agents are active and broadcasting at a given time t, or if their broadcast radius changes, perhaps in relation to signal or battery strength.

Definition 6 (Time-varying communication network). *The time-varying communication network* $\{\mathcal{G}_t\}_{t\geq 0}$ *is given by* $\mathcal{G}_t = (\mathcal{N}, \mathcal{E}_t)$*, where* $\mathcal{N}$ *is the set of vertices each representing an agent* $i = 1, \ldots, N$*, and the edge set* $\mathcal{E}_t \subseteq \{(i,j) : i,j \in \mathcal{N}, i \neq j\}$ *is the set of undirected communication links by which information can be shared at time t.*

A network is *connected* if there is a sequence of distinct edges forming a path between each distinct pair of vertices. The *union* of a collection of graphs $\{\mathcal{G}_t, \mathcal{G}_{t+1}, \cdots, \mathcal{G}_{t+\omega}\}$ ($\omega \in \mathbb{N}$) is the graph with vertices and edge set equalling the union of the vertices and edge sets of the graphs in the collection (Jadbabaie et al., 2003). A collection is *jointly connected* if its members' union is connected. A network's *diameter* $d_{\mathcal{G}}$ is the maximum of the shortest path length between any pair of nodes.

3 Learning with networked, decentralised agents

Summary We first discuss theoretical versions of our operators and algorithm (Secs. 3.1, 3.2) to show that our networked framework has sample guarantees bounded between those of the centralised- and independent-learning cases (Sec. 3.3). We then show that our novel incorporation of an experience replay buffer (Sec. 3.4.2), along with networked communication, means that empirically we can remove many of the theoretical assumptions and practically infeasible hyperparameter choices that are required by the sample guarantees of the theoretical algorithms, in which cases we demonstrate that our networked algorithm can significantly outperform the independent algorithm, and sometimes even the centralised one (Sec. 4).

3.1 Learning with N agents from a single run

We begin by outlining the basic procedure for solving the MFG using the N-agent empirical distribution and a single, continuous system run. The two underlying operators are the same for the centralised, independent and networked architectures; in the latter two cases all agents apply the operators individually, while in the centralised setting a single 'central' agent (the agent with index $i = 1$) estimates the Q-function and computes an updated policy that is pushed to all the other agents.

We define, for $h_{\max} > 0$ and $h : \Delta_{\mathcal{A}} \to [0, h_{\max}]$, $u_{\max} \in \Delta_{\mathcal{A}}$ such that $h(u_{\max}) = h_{\max}$. We further define $q_{\max} := \frac{1+h_{\max}}{1-\gamma}$, and set $\pi_{\max} \in \Pi$ such that $\pi_{\max}(s) = u_{\max}, \forall s \in \mathcal{S}$. For any $\Delta h \in \mathbb{R}_{>0}$, we also define the convex set $\mathcal{U}_{\Delta h} := \{u \in \Delta_{\mathcal{A}} : h(u) \geq h_{\max} - \Delta h\}$.

Learning agents use the stochastic temporal difference (TD)-learning operator to repeatedly update an estimate of the Q-function of their current policy with respect to the current empirical distribution, i.e. to approximate the operator Γ_q (Def. 12, Appx. A):

Definition 7 (Stochastic TD-learning operator, simplified from Def. 4.1 in Yardim et al. (2023))**.** *We define $\mathcal{Z} := \mathcal{S} \times \mathcal{A} \times [0,1] \times \mathcal{S} \times \mathcal{A}$, and say that ζ_t^i is the transition observed by agent i at time t, given by $\zeta_t^i = (s_t^i, a_t^i, r_t^i, s_{t+1}^i, a_{t+1}^i)$. The TD-learning operator $\tilde{F}_\beta^\pi : \mathcal{Q} \times \mathcal{Z} \to \mathcal{Q}$ is defined, for any $Q \in \mathcal{Q}, \zeta_t \in \mathcal{Z}, \beta \in \mathbb{R}$, as*

$$\tilde{F}_\beta^\pi(Q, \zeta_t) = Q(s_t, a_t) - \beta \left(Q(s_t, a_t) - r_t - h(\pi(s_t)) - \gamma \left(Q(s_{t+1}, a_{t+1})\right)\right).$$

Having estimated the Q-function of their current policy, agents update this policy by selecting, for each state, a probability distribution over their actions that maximises the combination of three terms (Def. 8): 1. the value of the given state with respect to the estimated Q-function; 2. a regulariser over the action probability distribution (in practice, we maximise the scaled entropy of the distribution); 3. a metric of similarity between the new action probabilities for the given state and those of the previous policy, given by the squared two-norm of the difference between the two distributions. We can alter the importance of the similarity metric relative to the other two terms by varying a parameter η, which is equivalent to changing the learning rate of the policy update. The three terms in the maximisation function can be seen in the policy mirror ascent (PMA) operator:

Definition 8 (Policy mirror ascent operator (Def. 3.5, (Yardim et al., 2023)))**.** *For a learning rate $\eta > 0$ and $L_h := L_a + \gamma \frac{L_s K_a}{2 - \gamma K_s}$ (where these constants are defined in Assumption 1 in Appx. A), the PMA update operator $\Gamma_\eta^{md} : \mathcal{Q} \times \Pi \to \Pi$ is defined as*

$$\Gamma_\eta^{md}(Q, \pi)(s) := \underset{u \in \mathcal{U}_{L_h}}{\arg\max} \left(\langle u, q(s, \cdot) \rangle + h(u) - \tfrac{1}{2\eta} ||u - \pi(s)||_2^2 \right), \forall s \in \mathcal{S}, \forall Q \in \mathcal{Q}, \forall \pi \in \Pi.$$

The theoretical learning algorithm has three nested loops (see Lines 2, 4 and 5 of Alg. 1). The policy update is applied K times. Before the policy update in each of the K loops, agents update their estimate of the Q-function by applying the stochastic TD-learning operator M_{pg} times. Prior to the TD update in each of the M_{pg} loops, agents take M_{td} steps in the environment without updating. The M_{td} loops exist to create a delay between each TD update to reduce bias when using the empirical distribution to approximate the mean field in a single system run (Kotsalis et al., 2022). However, we find in our experiments that we are able to essentially remove the inner M_{td} loops (Sec. 4.1).

3.2 Decentralised communication between agents

In our novel algorithm Alg. 1, agents compute policy updates in a decentralised way as in the independent case (Lines 3-10), before exchanging policies with neighbours by the following method, which allows policies to spread through the population. Coupled to their updated policy π_{k+1}^i, agents generate a scalar value σ_{k+1}^i (Line 11). The value provides information that helps agents decide between policies that they may wish to adopt from neighbours. Different methods for choosing between values received from neighbours, and for generating the values in the first place, lead to different policies spreading through the population. For example, generating or choosing σ_{k+1}^i at random leads to policies being exchanged at random (required in Thm. 1), whereas generating σ_{k+1}^i as an approximation of the return of π_{k+1}^i and then selecting the highest received value of σ_{k+1}^j leads to better performing policies spreading through the population. The latter is the approach we

Algorithm 1 Networked learning with single system run

Require: loop parameters $K, M_{pg}, M_{td}, C,$ learning parameters $\eta, \{\beta_m\}_{m\in\{0,\dots,M_{pg}-1\}}, \lambda, \gamma,$ $\{\tau_k\}_{k\in\{0,\dots,K-1\}}$

Require: initial states $\{s_0^i\}_i, i = 1, \dots, N$

1: Set $\pi_0^i = \pi_{\max}, \forall i$ and $t \leftarrow 0$

2: **for** $k = 0, \dots, K-1$ **do**

3: $\quad \forall s, a, i : \hat{Q}_0^i(s,a) = Q_{\max}$

4: $\quad$ **for** $m = 0, \dots, M_{pg} - 1$ **do**

5: $\quad\quad$ **for** M_{td} iterations **do**

6: $\quad\quad\quad$ Take step $\forall i : a_t^i \sim \pi_k^i(\cdot|s_t^i), r_t^i = R(s_t^i, a_t^i, \hat{\mu}_t), s_{t+1}^i \sim P(\cdot|s_t^i, a_t^i, \hat{\mu}_t); t \leftarrow t+1$

7: $\quad\quad$ **end for**

8: $\quad\quad$ Compute TD update ($\forall i$): $\hat{Q}_{m+1}^i = \tilde{F}_{\beta_m}^{\pi_k^i}(\hat{Q}_m^i, \zeta_{t-2}^i)$ (see Def. 7)

9: $\quad$ **end for**

10: $\quad$ PMA step $\forall i : \pi_{k+1}^i = \Gamma_\eta^{md}(\hat{Q}_{M_{pg}}^i, \pi_k^i)$ (see Def. 8)

11: $\quad \forall i$: Generate σ_{k+1}^i associated with π_{k+1}^i

12: $\quad$ **for** C rounds **do**

13: $\quad\quad \forall i$: Broadcast $\sigma_{k+1}^i, \pi_{k+1}^i$

14: $\quad\quad \forall i : J_t^i = i \cup \{j \in \mathcal{N} : (i,j) \in \mathcal{E}_t\}$

15: $\quad\quad \forall i$: Select adopted$^i \sim \Pr\big(\text{adopted}^i = j\big) = \frac{\exp(\sigma_{k+1}^j/\tau_k)}{\sum_{x\in J_t^i}\exp(\sigma_{k+1}^x/\tau_k)}\ \forall j \in J_t^i$

16: $\quad\quad \forall i : \sigma_{k+1}^i \leftarrow \sigma_{k+1}^{\text{adopted}^i}, \pi_{k+1}^i \leftarrow \pi_{k+1}^{\text{adopted}^i}$

17: $\quad\quad$ Take step $\forall i : a_t^i \sim \pi_{k+1}^i(\cdot|s_t^i), r_t^i = R(s_t^i, a_t^i, \hat{\mu}_t), s_{t+1}^i \sim P(\cdot|s_t^i, a_t^i, \hat{\mu}_t); t \leftarrow t+1$

18: $\quad$ **end for**

19: **end for**

20: Return policies $\{\pi_K^i\}_i, i = 1, \dots, N$

use for accelerating empirical convergence (described in Sec. 3.4.1 on the practical running of our algorithm), albeit we use a softmax rather than a max function for selecting between received values. However, for our theoretical results, we do not focus on a specific method for generating σ_{k+1}^i, such that it can be arbitrary for Thms. 1 and 4 below, and with few restrictions for Thms. 2 and 3.

Agents broadcast their policy π_{k+1}^i and the associated σ_{k+1}^i value to their neighbours (Line 13). Agents have a certain broadcast radius, defining the structure of the possibly time-varying communication network. Of the policies and associated values received by a given agent (including its own) (Line 14), the agent selects a σ_{k+1}^j with a probability defined by a softmax function over the received values, and *adopts* the policy associated with this σ_{k+1}^i, i.e. it sets its own current π_{k+1}^i and σ_{k+1}^i to the ones it has selected (Lines 15, 16). This process continues for C communication rounds, before the Q-function estimation steps begin again. After each round, the agents take a step in the environment (Line 17), such that if the communication network is affected by the agents' states, then agents that are unconnected from any others in a given communication round might become connected in the next. (In our experiments we set C as 1 to show the benefits to convergence speed brought by even a single communication round.) We assume the softmax function is subject to a possibly time-varying temperature parameter τ_k. We discuss the effects of the values of C and τ_k, and the mechanism for generating σ_{k+1}^i, in subsequent sections.

Remark 2. *Our networked architecture is effectively a generalisation of both the centralised and independent settings (Algs. 2, 3, Yardim et al. (2023)). The independent setting is the special case where there is no communication, i.e. $C = 0$. The centralised setting is the special case when σ_{k+1}^i is generated from a unique ID for each agent, with the central learner agent assumed to generate the highest value by default. In this case we assume $\tau_k \to 0$ (such that the softmax becomes a max function), and that the communication network becomes jointly connected repeatedly, so the central learner's policy is always adopted by the entire population, assuming C is large enough that the number of jointly connected collections of graphs occurring within C is equal to the largest diameter of the union of any collection (Rajagopalan & Shah, 2010; Zhang et al., 2020).*

Remark 3. *In practice, when referring in the following to a centralised version of the networked Alg. 1, for simplicity we assume there is no communication and instead that the updated policy π^1_{k+1} of the central learner $i = 1$ is pushed to all other agents after Line 10, as in Alg. 2 of (Yardim et al., 2023).*

3.3 Properties of policy adoption

We first give two theoretical results comparing the sample guarantees of our networked case with those of the other settings; the results respectively depend on whether the networked agents select which communicated policies to adopt at random or not. We then provide the order of the difference in these bounds in the non-random case in terms of network structure and number of communication rounds. We finally give in Appx. B.8 a policy-update stability guarantee, which applies in all scenarios. Appx. A contains the full set of technical assumptions on which these theorems rely.

Theorem 1 (Networked learning with **random adoption**). *Full version in Appx. B.2. Say that π^* is the unique MFG-NE policy. Let $\varepsilon > 0$ be an arbitrary value that reduces as k increases by the following relation: $k = \frac{\log 8\varepsilon^{-1}}{\log L_{\Gamma_\eta}^{-1}}$, where the constant L_{Γ_η} is defined in Lem. 2 in Appx. A. Let us assume that $C > 0$ and $\tau_k \to \infty$, meaning that the softmax function approaches a uniform distribution such that the values of σ^i_{k+1} are arbitrary and received policies are adopted at random. Then, under the technical assumptions given in Appx. B.2, the random output $\{\pi^i_K\}_i$ of Alg. 1 preserves the sample guarantees of the independent-learning case given in Lem. 3, i.e. the output satisfies, for all agents $i = 1, \ldots, N$, $\mathbb{E}\left[||\pi^i_K - \pi^*||_1\right] \leq \varepsilon + \mathcal{O}\left(\frac{1}{\sqrt{N}}\right)$.*

Proof sketch. Random exchange of policies learnt in a decentralised manner does not change the expectation of the random output of the purely independent-learning setting i.e. where this exchange does not occur. Full proof in Appx. B.3.

Moreover we can show that if σ^i_{k+1} is generated arbitrarily and uniquely for each i, then for $\tau_k \in \mathbb{R}_{>0}$ (such that the softmax function gives a non-uniform distribution and adoption of received policies is therefore non-random), the sample complexity of the networked learning algorithm is bounded between that of the centralised and independent algorithms:

Theorem 2 (Networked learning with **non-random adoption**). *Full version in Appx. B.5. Assume that Alg. 1 is run as in Thm. 1, except now $\tau_k \in \mathbb{R}_{>0}$. ε is defined as above. Assume also that σ^i_{k+1} is generated uniquely for each i, in a manner independent of any metric related to π^i_{k+1}, e.g. σ^i_{k+1} is random or related only to the index i (so as not to bias the spread of any particular policy). Let the random output of this Algorithm be denoted as $\{\pi^{i,net}_K\}_i$. Also consider an independent-learning version of the algorithm (i.e. with the same parameters except $C = 0$) and denote its random output $\{\pi^{i,ind}_K\}_i$; and a centralised version of the algorithm with the same parameters (see Rem. 3) and denote its random output as π^{cent}_K. Then, under the technical assumptions given in Appx. B.5, for all agents $i = 1, \ldots, N$, the random outputs $\{\pi^{i,net}_K\}_i$, $\{\pi^{i,ind}_K\}_i$ and π^{cent}_K satisfy the following relations, where ub_{net}, ub_{ind} and ub_{cent} are respective upper bounds for each case:*

$$\mathbb{E}\left[||\pi^{cent}_K - \pi^*||_1\right] \leq ub_{cent}, \quad \mathbb{E}\left[||\pi^{i,net}_K - \pi^*||_1\right] \leq ub_{net}, \quad \mathbb{E}\left[||\pi^{i,ind}_K - \pi^*||_1\right] \leq ub_{ind},$$

$$\textit{where} \quad ub_{cent} \leq ub_{net} \leq ub_{ind} = \varepsilon + \mathcal{O}\left(\frac{1}{\sqrt{N}}\right).$$

Proof sketch. Fewer distinct policies in the population means there is less bias in each learner's estimation of the Q function. So methods for reducing the number of policies in the population (such as non-random adoption in the networked case) lead to faster convergence. Full proof in Appx. B.6.

Theorem 3 (Relation between communication network structure and order of difference between the frameworks' bounds)**.** *In addition to the assumptions in Thm. 2, now also assume that the communication network $\mathcal{G}_t$ remains static and connected during the C communication rounds, and that $\tau_k \to 0\ \forall k$ such that the softmax essentially becomes a max function. Assume also the diameter $d_\mathcal{G}$ of the network is equal for all k. Then we can say that the difference in the upper bounds ub_{net}, ub_{ind} and ub_{cent} from Thm. 2 depends on C and $d_\mathcal{G}$ as follows, for the tight bound big Theta (Θ):*

$$ub_{cent} + \Theta\left(f(C, d_\mathcal{G})\right) \approx ub_{net} \approx ub_{ind} - \Theta\left(1 - f(C, d_\mathcal{G})\right),$$

for the piecewise function $f(C, d_{\mathcal{G}})$ defined as

$$f(C, d_{\mathcal{G}}) = \begin{cases} \left(\left(1 - \frac{1}{d_{\mathcal{G}}}\right)^{C}\right) & \text{if } C < d_{\mathcal{G}}, \\ 0 & \text{if } C \geq d_{\mathcal{G}} \end{cases}.$$

When $C \geq d_{\mathcal{G}}$, $ub_{net} = ub_{cent}$, so for $C > d_{\mathcal{G}}$ there is no additional improvement over the centralised bound. Equally when $C = 0$, we have exactly $ub_{net} = ub_{ind}$.

Proof sketch. The difference in the architectures' convergence rates depends on the different amounts of divergence between the population's policies. In a static, connected network, max-consensus is reached when the number of communication rounds C is equal to the diameter $d_{\mathcal{G}}$, giving 0 divergence as in the centralised case. The rate of divergence decrease is approximately $\frac{1}{d_{\mathcal{G}}}$ for each of the C rounds, giving the exponential relation involving C and $d_{\mathcal{G}}$. Full proof in Appx. B.7.

Remark 4. *Thm. 3 depends on the assumptions that the communication network is static and fixed, and has the same diameter $d_{\mathcal{G}}$ for all k. If we assume instead that the network is only repeatedly jointly connected, we can replace $d_{\mathcal{G}}$ in the results in Thm. 3 with $d_{avg} \cdot \omega$, namely the average diameter of the union of each jointly connected collection of graphs multiplied by the average number of graphs in each jointly connected collection. As noted in Rem. 2, max-consensus is reached if C is large enough that the number of jointly connected collections of graphs occurring within C is equal to the largest diameter of the union of any collection, giving the centralised case; there is no added benefit to higher values of C than this.*

See Rem. 7 in Appx. B.7 for a similar remark loosening the assumption that $\tau_k \to 0\ \forall k$.

3.4 Practical running of algorithms

3.4.1 Generation of σ^i_{k+1}

The theoretical analysis in Sec. 3.3 requires algorithmic hyperparameters that render learning impractically slow in all of the centralised, independent and networked cases (see Rem. 8, Appx. B.9). For practical convergence of the algorithms, we seek to drastically increase $\{\beta_m\}$ and reduce M_{td} and M_{pg}, though this will naturally break the theoretical guarantees and give a poorer estimation of the Q-function $\hat{Q}^i_{M_{pg}}$, and hence a greater variance in the quality of the updated policies π^i_{k+1}. However, in such cases we found empirically that an appropriate method for generating σ^i_{k+1} dependent on π^i_{k+1} allows our networked algorithm to significantly outperform the independent setting, and sometimes even the centralised setting, by advantageously biasing the spread of particular policies. This is instead of generating σ^i_{k+1} arbitrarily as required in the theoretical settings in Sec. 3.3.

We do so by setting σ^i_{k+1} to a finite approximation $\widehat{\Psi^i_{h,k+1}}(\boldsymbol{\pi}_{k+1}, \upsilon_0)$ of $\Psi^i_{h,k+1}(\boldsymbol{\pi}_{k+1}, \upsilon_0)$ where $\boldsymbol{\pi}_{k+1} := (\pi^1_{k+1}, \ldots, \pi^N_{k+1})$, by tracking the discounted return for E evaluation steps. This is given by

$$\widehat{\Psi^i_{h,k+1}}(\boldsymbol{\pi}_{k+1}, \upsilon_0) = \left[\sum_{e=0}^{E} \gamma^e (R(s^i_t, a^i_t, \hat{\mu}_t) + h(\pi^i(s^i_t))) \Bigg|^{t=t+e}_{\substack{a^j_t \sim \pi^j_{k+1}(s^j_t) \\ s^j_{t+1} \sim P(\cdot | s^j_t, a^j_t, \hat{\mu}_t)}}, \forall j \in \{1, \ldots, N\} \right].$$

Generating σ^i_{k+1} in this way means the policies that are more likely to be adopted and spread through the network are those that are estimated to receive a higher return in reality, despite being generated from poorly estimated Q-functions. This explains why our networked method can in practice outperform even the centralised case, where the updated policy of the arbitrary agent $i = 1$ gets pushed to all other agents regardless of its quality (see Appx. C for a more formal explanation). Naturally the quality of the finite approximation depends on the number of evaluation steps E, but we found empirically that E can be much smaller than M_{pg} and still give marked convergence benefits.

3.4.2 Algorithm acceleration by use of experience-replay buffer

Even with networked communication, the empirical learning of our original algorithm is too slow for practical demonstration, as also in the centralised and independent cases - see the ablation study in Appx. F.4. We therefore offer a further technical contribution allowing the first practical demonstrations of all three architectures for learning from a single continued system run.

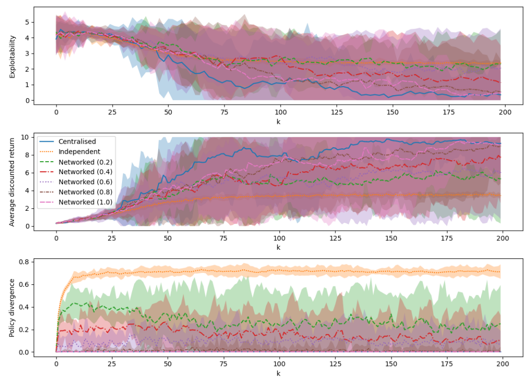

Figure 1: ‘Target agreement’ game. Even with only a single communication round, our networked case outperforms the independent case wrt. exploitability, and markedly outperforms wrt. return. The fact that the lowest broadcast radius (0.2) ends with similar exploitability to the independent case yet much higher return shows our networked algorithm can help agents find ‘preferable’ equilibria.

We modify our Alg. 1 (shown in *blue* in Alg. 2, Appx. D), as follows. Instead of using a transition ζ^i_{t-2} to compute the TD update within each M_{pg} iteration and then discarding the transition, we store the transition in a buffer (Line 9) until after the M_{pg} loops. Replay buffers are a common (MA)RL tool used especially with deep learning, precisely to improve data efficiency and reduce autocorrelation (Lin, 1992; Fedus et al., 2020; Xu et al., 2024). When learning does take place in our modified algorithm (Lines 11-16), it involves cycling through the buffer for L iterations - randomly shuffling the buffer between each - and thus conducting the TD update on each stored transition L times. This allows us to reduce the number of M_{pg} loops, as well as not requiring as small a learning rate $\{\beta_m\}$, allowing much faster learning in practice. Moreover, by shuffling the buffer before each cycle we reduce bias resulting from the dependency of samples along the single path, which may justify being able to achieve adequate stable learning even when reducing the number of M_{td} waiting steps within each M_{pg} loop. The buffer means the theoretical guarantees given in Sec. 3.3 no longer apply, but we exchange this for practical convergence times. See Appx. D for further discussion.

4 Experiments

Our technical contribution of the replay buffer to MFG algorithms for learning from continuous system runs allows us also to contribute the first empirical demonstrations of these algorithms, not just in the networked case but also in the centralised and independent cases. The latter two serve as baselines to show the advantages of the networked architecture. We follow prior works on stationary MFGs in the types of game demonstrated (Zaman et al., 2023; Lauriere et al., 2022; Algumaei et al., 2023; Lauriere, 2021; Cui et al., 2023). We focus on grid-world environments where agents can move in the four cardinal directions or remain in place. We present results from two tasks defined by the agents’ reward functions; see Appx. F.1 for full technical description of our task settings.

Cluster. Agents are rewarded for gathering together. The agents are given no indication where they should cluster, agreeing this themselves over time.

Target agreement. The agents are rewarded for visiting any of a given number of targets, but their reward is proportional to the number of other agents co-located at that target. The agents must therefore coordinate on which single target they will all meet at to maximise their individual rewards.

As well as the standard scenario for these tasks, we conduct robustness tests in two settings, reflecting those elaborated in Appx. E. The first illustrates robustness to learning failures: at every iteration k each learner (whether centralised or decentralised) fails to update its policy (i.e. Line 10 of Alg. 1 is not executed such that $\pi^i_{k+1} = \pi^i_k$) with a 50% probability. The second test illustrates robustness to increases in population size. Instead of having 250 agents throughout, the population begins with 50 agents learning normally, and a further 200 agents are added to the population at the marked point.

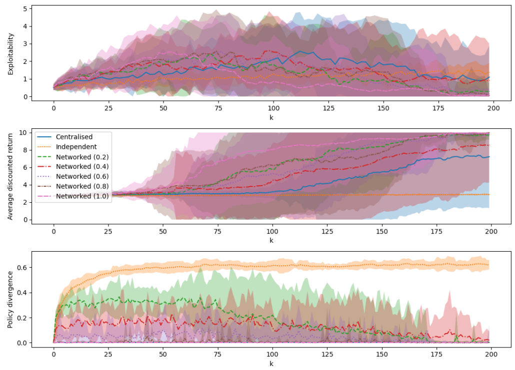

Figure 2: 'Cluster' game, testing robustness to 50% probability of policy update failure. The communication network allows agents that have successfully updated their policies to spread this information to those that have not, providing redundancy. Independent learners cannot do this and hardly appear to learn at all (no increase in return); likewise the centralised architecture is susceptible to its single point of failure. Thus our networked architecture significantly outperforms both the centralised and independent cases.

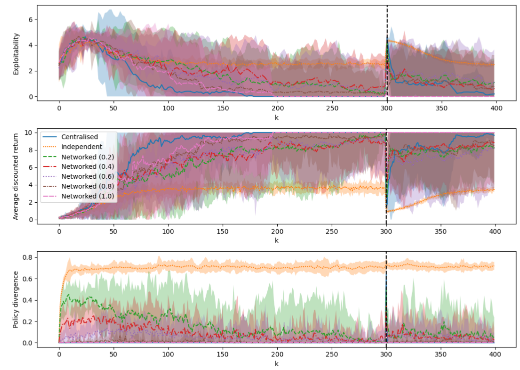

Figure 3: 'Target agreement' game, testing robustness to a five-times increase in population. The networked architectures are quickly able to spread the learnt policies to the newly arrived agents such that learning progress is minimally disturbed, whereas convergence is significantly impacted in the independent case. The largest broadcast radius (1.0), in particular, appears to suffer no disturbance at all, being much more robust than the centralised case, which takes a significant amount of time to return to equilibrium.

Experiments are evaluated via three metrics (see Appx. F.2 for a full discussion): an approximation of the **exploitability** of the joint policy $\boldsymbol{\pi}_k$; the **average discounted return of the agents' policies** π^i_k; and the **population's policy divergence**. Hyperparameters are discussed in Appx. F.3.

4.1 Discussion

We give here three example figures (**reproduced larger** in Figs. 4, 5 and 6) illustrating the benefits of the networked architecture; in each the decimals refer to each agent's broadcast radius as a fraction of the maximum possible distance in the grid (i.e. the diagonal). See figure captions for details, and Appx. F.4 for further experiments, ablation studies and discussion. For limitations and future work, see Appx. H. As well as allowing convergence within a practical number of iterations, even with only a single communication round, the combination of the buffer and the networked architecture allows us to remove in our experiments a number of the assumptions required for the theoretical algorithms:

- We significantly reduce M_{pg} while still converging within a reasonable K. With smaller values for M_{pg} (the number of samples in the buffer) and L (the number of loops through the buffer for updating the Q-function), and hence with worse estimation of the Q-function, the networked architecture outperforms the independent case to an even greater extent. This underlines its advantages in allowing faster convergence in practical settings.
- We can reduce the M_{td} parameter (theoretically required for the learner to wait between collecting samples when learning from a single system run) to 1, effectively removing the innermost loop of the nested learning algorithm (see Line 5 of Alg. 1).
- We can reduce the scaling parameter λ of the entropy regulariser to 0, i.e. we converge even without regularisation, allowing us to leave the NE unbiased, and also removing Assumption 3 (Appx. A). In general an unregularised MFG-NE is not unique (Yardim et al., 2023); the ability of the agents to coordinate on one of the multiple solutions in the centralised and networked cases may explain why they outperform the independent-learning case.
- For the PMA operator (Def. 8), we conduct the optimisation over the set $u \in \Delta_{\mathcal{A}}$ instead of $u \in \mathcal{U}_{L_h}$, i.e. we can choose from all possible probability distributions over actions instead of needing to identify the Lipschitz constants given in Assumption 1 (Appx. A).

5 Broader impact / ethics statement

As with many advances in machine learning, and those relating to multi-agent systems in particular, in the long term our research on large populations of coordinating agents could have negative social outcomes if pursued by malicious actors, including surveillance and military uses. However, our work is primarily foundational and far from deployments, and it also has a large range of potential beneficial applications (such as smart grids and disaster response). Moreover, better understanding the dynamics of large multi-agent systems (as we seek to do in this paper) can contribute to ensuring safety by reducing the risks of unintended failures or outcomes.

We hope to help mitigate potential harmful consequences of this research by fostering transparency through submitting our code in the Supplementary Material, which we commit to publishing online under license upon acceptance of the paper.

6 Reproducibility statement

The code files to run our experiments are uploaded in the Supplementary Material. We discuss the hyperparameters for our experiments in Table 1 in Appx. F.3. The technical assumptions for our theoretical results are given in Appx. A, and complete proofs are provided in Appx. B.

REFERENCES

Yves Achdou and Italo Capuzzo-Dolcetta. Mean Field Games: Numerical Methods. *SIAM Journal on Numerical Analysis*, 48(3):1136–1162, 2010. doi: 10.1137/090758477. URL `https://doi.org/10.1137/090758477`.

Yves Achdou, Pierre Cardaliaguet, François Delarue, Alessio Porretta, Filippo Santambrogio, Yves Achdou, and Mathieu Laurière. Mean Field Games and Applications: Numerical Aspects. *Mean Field Games: Cetraro, Italy 2019*, pp. 249–307, 2020.

Shubham Aggarwal, Melih Bastopcu, Sennur Ulukus, Tamer Başar, et al. A mean field game model for timely computation in edge computing systems. *arXiv preprint arXiv:2404.02898*, 2024.

Talal Algumaei, Ruben Solozabal, Reda Alami, Hakim Hacid, Merouane Debbah, and Martin Takáč. Regularization of the policy updates for stabilizing Mean Field Games. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 361–372. Springer, 2023.

Adeeba Ali, Rashid Ali, and M.F. Baig. Distributed Multi-Agent Deep Reinforcement Learning based Navigation and Control of UAV Swarm for Wildfire Monitoring. In *2023 IEEE 4th Annual Flagship India Council International Subsections Conference (INDISCON)*, pp. 1–8, 2023. doi: 10.1109/INDISCON58499.2023.10270198.

Berkay Anahtarci, Can Deha Karıksız, and Naci Saldi. Fitted Q-Learning in Mean-field Games. *ArXiv*, abs/1912.13309, 2019.

Berkay Anahtarci, Can Deha Kariksiz, and Naci Saldi. Q-learning in regularized mean-field games. *Dynamic Games and Applications*, 13(1):89–117, 2023.

David Andréen, Petra Jenning, Nils Napp, and Kirstin Petersen. Emergent structures assembled by large swarms of simple robots. In *Acadia*, pp. 54–61, 2016.

Andrea Angiuli, Jean-Pierre Fouque, and Mathieu Laurière. Unified reinforcement Q-learning for mean field game and control problems. *Mathematics of Control, Signals, and Systems*, 34(2): 217–271, 2022.

Andrea Angiuli, Jean-Pierre Fouque, Mathieu Laurière, and Mengrui Zhang. Convergence of Multi-Scale Reinforcement Q-Learning Algorithms for Mean Field Game and Control Problems. *arXiv preprint arXiv:2312.06659*, 2023.

Uğur Aydın and Naci Saldi. Robustness and Approximation of Discrete-time Mean-field Games under Discounted Cost Criterion. *arXiv preprint arXiv:2310.10828*, 2023.

Dario Bauso and Hamidou Tembine. Crowd-Averse Cyber-Physical Systems: The Paradigm of Robust Mean-Field Games. *IEEE Transactions on Automatic Control*, 61(8):2312–2317, 2016. doi: 10.1109/TAC.2015.2492038.

Dario Bauso, Hamidou Tembine, and Tamer Başar. Robust Mean Field Games with Application to Production of an Exhaustible Resource. *IFAC Proceedings Volumes*, 45(13):454–459, 2012. ISSN 1474-6670. doi: https://doi.org/10.3182/20120620-3-DK-2025.00135. URL `https://www.sciencedirect.com/science/article/pii/S1474667015377302`. 7th IFAC Symposium on Robust Control Design.

Dario Bauso, Hamidou Tembine, and Tamer Başar. Robust mean field games. *Dynamic games and applications*, 6(3):277–303, 2016.

Amani Benamor, Oussama Habachi, Inès Kammoun, and Jean-Pierre Cances. NOMA-based Power Control for Machine-Type Communications: A Mean Field Game Approach. In *2022 IEEE International Performance, Computing, and Communications Conference (IPCCC)*, pp. 338–343, 2022. doi: 10.1109/IPCCC55026.2022.9894296.

Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.

Rico Berner, Thilo Gross, Christian Kuehn, Jürgen Kurths, and Serhiy Yanchuk. Adaptive dynamical networks. *Physics Reports*, 1031:1–59, 2023.

Luis Briceño-Arias, Dante Kalise, and Francisco Silva. Proximal methods for stationary Mean Field Games with local couplings. *SIAM Journal on Control and Optimization*, 56:801–, 03 2018.

Lucian Busoniu, Robert Babuska, and Bart De Schutter. A Comprehensive Survey of Multiagent Reinforcement Learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(2):156–172, 2008. doi: 10.1109/TSMCC.2007.913919.

Cacace, Simone, Camilli, Fabio, and Goffi, Alessandro. A policy iteration method for mean field games. *ESAIM: COCV*, 27:85, 2021. doi: 10.1051/cocv/2021081. URL `https://doi.org/10.1051/cocv/2021081`.

Nicolas Cambier, Vincent Frémont, Vito Trianni, and Eliseo Ferrante. Embodied evolution of self-organised aggregation by cultural propagation. In Marco Dorigo, Mauro Birattari, Christian Blum, Anders L. Christensen, Andreagiovanni Reina, and Vito Trianni (eds.), *Swarm Intelligence*, pp. 351–359, Cham, 2018. Springer International Publishing.

Nicolas Cambier, Roman Miletitch, Vincent Fremont, Marco Dorigo, Eliseo Ferrante, and Vito Trianni. Language Evolution in Swarm Robotics: A Perspective. *Frontiers in Robotics and AI*, 7, 2020. ISSN 2296-9144. doi: 10.3389/frobt.2020.00012. URL `https://www.frontiersin.org/articles/10.3389/frobt.2020.00012`.

Nicolas Cambier, Dario Albani, Vincent Fremont, Vito Trianni, and Eliseo Ferrante. Cultural evolution of probabilistic aggregation in synthetic swarms. *Applied Soft Computing*, 113:108010, 2021. ISSN 1568-4946. doi: https://doi.org/10.1016/j.asoc.2021.108010. URL `https://www.sciencedirect.com/science/article/pii/S1568494621009327`.

Haoyang Cao, Xin Guo, and Mathieu Laurière. Connecting GANs, MFGs, and OT. *arXiv preprint arXiv:2002.04112*, 2020.

Kris Cao, Angeliki Lazaridou, Marc Lanctot, Joel Z. Leibo, Karl Tuyls, and Stephen Clark. Emergent Communication through Negotiation. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=Hk6WhagRW`.

Cardaliaguet, Pierre and Hadikhanloo, Saeed. Learning in mean field games: The fictitious play. *ESAIM: COCV*, 23(2):569–591, 2017. doi: 10.1051/cocv/2016004. URL `https://doi.org/10.1051/cocv/2016004`.

E. Carlini and F. J. Silva. A Fully Discrete Semi-Lagrangian Scheme for a First Order Mean Field Game Problem. *SIAM Journal on Numerical Analysis*, 52(1):45–67, 2014. doi: 10.1137/120902987. URL `https://doi.org/10.1137/120902987`.

René Carmona and Mathieu Laurière. Deep learning for mean field games and mean field control with applications to finance. *arXiv preprint arXiv:2107.04568*, 7, 2021.

René Carmona, Mathieu Laurière, and Zongjun Tan. Model-Free Mean-Field Reinforcement Learning: Mean-Field MDP and Mean-Field Q-Learning. *The Annals of Applied Probability*, 33(6B): 5334–5381, 2023.

Lu Chang, Liang Shan, Weilong Zhang, and Yuewei Dai. Hierarchical multi-robot navigation and formation in unknown environments via deep reinforcement learning and distributed optimization. *Robotics and Computer-Integrated Manufacturing*, 83:102570, 2023. ISSN 0736-5845. doi: https://doi.org/10.1016/j.rcim.2023.102570. URL `https://www.sciencedirect.com/science/article/pii/S0736584523000467`.

Mingzhe Chen, Deniz Gündüz, Kaibin Huang, Walid Saad, Mehdi Bennis, Aneta Vulgarakis Feljan, and H Vincent Poor. Distributed Learning in Wireless Networks: Recent Progress and Future Challenges. *IEEE Journal on Selected Areas in Communications*, 39(12):3579–3605, 2021.

Kai Cui and Heinz Koeppl. Approximately Solving Mean Field Games via Entropy-Regularized Deep Reinforcement Learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 1909–1917. PMLR, 2021.

Kai Cui, Anam Tahir, Gizem Ekinci, Ahmed Elshamanhory, Yannick Eich, Mengguang Li, and Heinz Koeppl. A Survey on Large-Population Systems and Scalable Multi-Agent Reinforcement Learning. *arXiv preprint arXiv:2209.03859*, 2022.

Kai Cui, Christian Fabian, and Heinz Koeppl. Multi-Agent Reinforcement Learning via Mean Field Control: Common Noise, Major Agents and Approximation Properties. *arXiv preprint arXiv:2303.10665*, 2023.

Kai Cui, Gökçe Dayanıklı, Mathieu Laurière, Matthieu Geist, Olivier Pietquin, and Heinz Koeppl. Learning Discrete-Time Major-Minor Mean Field Games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 9616–9625, 2024.

Constantinos Daskalakis, Paul W. Goldberg, and Christos H. Papadimitriou. The Complexity of Computing a Nash Equilibrium. In *Proceedings of the Thirty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '06, pp. 71–78, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595931341. doi: 10.1145/1132516.1132527. URL https://doi.org/10.1145/1132516.1132527.

Murad Dawood, Sicong Pan, Nils Dengler, Siqi Zhou, Angela P Schoellig, and Maren Bennewitz. Safe Multi-Agent Reinforcement Learning for Formation Control without Individual Reference Targets. *arXiv preprint arXiv:2312.12861*, 2023.

Shawon Dey and Hao Xu. Intelligent Distributed Charging Control for Large Scale Electric Vehicles: A Multi-Cluster Mean Field Game Approach. In *Proceedings of Cyber-Physical Systems and Internet of Things Week 2023*, CPS-IoT Week '23, pp. 146–151, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400700491. doi: 10.1145/3576914.3587709. URL https://doi.org/10.1145/3576914.3587709.

Thinh Doan, Siva Maguluri, and Justin Romberg. Finite-Time Analysis of Distributed TD(0) with Linear Function Approximation on Multi-Agent Reinforcement Learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 1626–1635. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/doan19a.html.

Adam Eck, Leen-Kiat Soh, and Prashant Doshi. Decision making in open agent systems. *AI Mag.*, 44(4):508–523, dec 2023. ISSN 0738-4602. doi: 10.1002/aaai.12131. URL https://doi.org/10.1002/aaai.12131.

A. E. Eiben and J. E. Smith. *What Is an Evolutionary Algorithm?*, pp. 25–48. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015. ISBN 978-3-662-44874-8. doi: 10.1007/978-3-662-44874-8_3. URL https://doi.org/10.1007/978-3-662-44874-8_3.

Romuald Elie, Julien Pérolat, Mathieu Laurière, Matthieu Geist, and Olivier Pietquin. On the Convergence of Model Free Learning in Mean Field Games. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7143–7150, Apr. 2020. doi: 10.1609/aaai.v34i05.6203. URL https://ojs.aaai.org/index.php/AAAI/article/view/6203.

Yousef Emami, Hao Gao, Kai Li, Luis Almeida, Eduardo Tovar, and Zhu Han. Age of Information Minimization using Multi-agent UAVs based on AI-Enhanced Mean Field Resource Allocation. *IEEE Transactions on Vehicular Technology*, pp. 1–14, 2024. doi: 10.1109/TVT.2024.3394235.

William Fedus, Prajit Ramachandran, Rishabh Agarwal, Yoshua Bengio, Hugo Larochelle, Mark Rowland, and Will Dabney. Revisiting Fundamentals of Experience Replay. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

Iñaki Fernández Pérez and Stéphane Sanchez. Influence of Local Selection and Robot Swarm Density on the Distributed Evolution of GRNs. In Paul Kaufmann and Pedro A. Castillo (eds.), *Applications of Evolutionary Computation*, pp. 567–582, Cham, 2019. Springer International Publishing. ISBN 978-3-030-16692-2.

Iñaki Fernández Pérez, Amine Boumaza, and François Charpillet. Maintaining Diversity in Robot Swarms with Distributed Embodied Evolution. In Marco Dorigo, Mauro Birattari, Christian Blum, Anders L. Christensen, Andreagiovanni Reina, and Vito Trianni (eds.), *Swarm Intelligence*, pp. 395–402, Cham, 2018. Springer International Publishing. ISBN 978-3-030-00533-7.

Jean-Pierre Fouque and Zhaoyu Zhang. Deep Learning Methods for Mean Field Control Problems With Delay. *Frontiers in Applied Mathematics and Statistics*, 6, 2020. ISSN 2297-4687. doi: 10.3389/fams.2020.00011. URL https://www.frontiersin.org/articles/10.3389/fams.2020.00011.

J Frédéric Bonnans, Pierre Lavigne, and Laurent Pfeiffer. Generalized conditional gradient and learning in potential mean field games. *arXiv e-prints*, pp. arXiv–2109, 2021.

Zuyue Fu, Zhuoran Yang, Yongxin Chen, and Zhaoran Wang. Actor-critic provably finds Nash equilibria of linear-quadratic mean-field games. *arXiv preprint arXiv:1910.07498*, 2019.

Sriram Ganapathi Subramanian, Pascal Poupart, Matthew E Taylor, and Nidhi Hegde. Multi Type Mean Field Reinforcement Learning. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 411–419, 2020.

Sriram Ganapathi Subramanian, Matthew E Taylor, Mark Crowley, and Pascal Poupart. Partially Observable Mean Field Reinforcement Learning. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 537–545, 2021.

Yuzhao Gao, Yiming Nie, and Hongliang Wang. A Scalable Multi-agent Reinforcement Learning Approach Based on Value Function Decomposition. In Yi Qu, Mancang Gu, Yifeng Niu, and Wenxing Fu (eds.), *Proceedings of 3rd 2023 International Conference on Autonomous Unmanned Systems (3rd ICAUS 2023)*, pp. 88–96, Singapore, 2024. Springer Nature Singapore. ISBN 978-981-97-1087-4.

Matthieu Geist, Julien Pérolat, Mathieu Laurière, Romuald Elie, Sarah Perrin, Olivier Bachem, Rémi Munos, and Olivier Pietquin. Concave Utility Reinforcement Learning: the Mean-Field Game Viewpoint. *arXiv preprint arXiv:2106.03787*, 2021.

Maximilien Germain, Joseph Mikael, and Xavier Warin. Numerical resolution of McKean-Vlasov FBSDEs using neural networks. *Methodology and Computing in Applied Probability*, 24(4): 2557–2586, 2022.

Jorge Gomes and Anders L. Christensen. Generic Behaviour Similarity Measures for Evolutionary Swarm Robotics. In *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation*, GECCO '13, pp. 199–206, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450319638. doi: 10.1145/2463372.2463398. URL https://doi.org/10.1145/2463372.2463398.

Sergio Grammatico, Basilio Gentile, Francesca Parise, and John Lygeros. A Mean Field control approach for demand side management of large populations of Thermostatically Controlled Loads. In *2015 European Control Conference (ECC)*, pp. 3548–3553, 2015a. doi: 10.1109/ECC.2015.7331083.

Sergio Grammatico, Francesca Parise, and John Lygeros. Constrained linear quadratic deterministic mean field control: Decentralized convergence to Nash equilibria in large populations of heterogeneous agents. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pp. 4412–4417, 2015b. doi: 10.1109/CDC.2015.7402908.

Sergio Grammatico, Francesca Parise, Marcello Colombino, and John Lygeros. Decentralized Convergence to Nash Equilibria in Constrained Deterministic Mean Field Control. *IEEE Transactions on Automatic Control*, 61(11):3315–3329, 2016. doi: 10.1109/TAC.2015.2513368.

Yue Guan, Sai Zou, Haixia Peng, Wei Ni, Yanglong Sun, and Hongfeng Gao. Cooperative UAV Trajectory Design for Disaster Area Emergency Communications: A Multiagent PPO Method. *IEEE Internet of Things Journal*, 11(5):8848–8859, 2024. doi: 10.1109/JIOT.2023.3320796.

Xin Guo, Anran Hu, Renyuan Xu, and Junzi Zhang. Learning Mean-Field Games. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019a. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/030e65da2b1c944090548d36b244b28d-Paper.pdf.

Xin Guo, Anran Hu, Renyuan Xu, and Junzi Zhang. Learning Mean-Field Games. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019b. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/030e65da2b1c944090548d36b244b28d-Paper.pdf.

Xin Guo, Renyuan Xu, and Thaleia Zariphopoulou. Entropy Regularization for Mean Field Games with Learning. *Math. Oper. Res.*, 47(4):3239–3260, nov 2022. ISSN 0364-765X. doi: 10.1287/moor.2021.1238. URL https://doi.org/10.1287/moor.2021.1238.

Xin Guo, Anran Hu, Renyuan Xu, and Junzi Zhang. A General Framework for Learning Mean-Field Games. *Mathematics of Operations Research*, 48(2):656–686, 2023.

Evert Haasdijk, Nicolas Bredeche, and Agoston E Eiben. Combining environment-driven adaptation and task-driven optimisation in evolutionary robotics. *PloS one*, 9(6):e98466, 2014.

Emma Hart, Andreas Steyven, and Ben Paechter. Improving Survivability in Environment-Driven Distributed Evolutionary Algorithms through Explicit Relative Fitness and Fitness Proportionate Communication. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, GECCO '15, pp. 169–176, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450334723. doi: 10.1145/2739480.2754688. URL https://doi.org/10.1145/2739480.2754688.

Paulo Heredia, Hasan Ghadialy, and Shaoshuai Mou. Finite-Sample Analysis of Distributed Q-learning for Multi-Agent Networks. In *2020 American Control Conference (ACC)*, pp. 3511–3516, 2020. doi: 10.23919/ACC45564.2020.9147428.

Anran Hu and Junzi Zhang. Mf-oml: Online mean-field reinforcement learning with occupation measures for large population games. *arXiv preprint arXiv:2405.00282*, 2024. URL https://arxiv.org/abs/2405.00282.

Tianfeng Hu, Zhiqun hu, Zhaoming Lu, and Xiangming Wen. Dynamic traffic signal control using mean field multi-agent reinforcement learning in large scale road-networks. *IET Intelligent Transport Systems*, 04 2023. doi: 10.1049/itr2.12364.

Han Huang and Rongjie Lai. Unsupervised Solution Operator Learning for Mean-Field Games via Sampling-Invariant Parametrizations. *arXiv preprint arXiv:2401.15482*, 2024.

Jianhui Huang and Minyi Huang. Robust Mean Field Linear-Quadratic-Gaussian Games with Unknown L^2-Disturbance. *SIAM Journal on Control and Optimization*, 55(5):2811–2840, 2017. doi: 10.1137/15M1014437. URL https://doi.org/10.1137/15M1014437.

Jiawei Huang, Niao He, and Andreas Krause. Model-Based RL for Mean-Field Games is not Statistically Harder than Single-Agent RL. *arXiv preprint arXiv:2402.05724*, 2024a.

Jiawei Huang, Batuhan Yardim, and Niao He. On the Statistical Efficiency of Mean-Field Reinforcement Learning with General Function Approximation . In Sanjoy Dasgupta, Stephan Mandt, and Yingzhen Li (eds.), *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pp. 289–297. PMLR, 02–04 May 2024b. URL https://proceedings.mlr.press/v238/huang24a.html.

Kuang Huang, Xuan Di, Qiang Du, and Xi Chen. A game-theoretic framework for autonomous vehicles velocity control: Bridging microscopic differential games and macroscopic mean field games. *Discrete and Continuous Dynamical Systems - B*, 25(12):4869–4903, 2020. ISSN 1531-3492. doi: 10.3934/dcdsb.2020131.

Minyi Huang, Roland P. Malhamé, and Peter E. Caines. Large population stochastic dynamic games: closed-loop McKean-Vlasov systems and the Nash certainty equivalence principle. *Communications in Information & Systems*, 6(3):221 – 252, 2006.

A. Jadbabaie, Jie Lin, and A.S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001, 2003. doi: 10.1109/TAC.2003.812781.

Natasha Jaques, Angeliki Lazaridou, Edward Hughes, Caglar Gulcehre, Pedro Ortega, DJ Strouse, Joel Z Leibo, and Nando De Freitas. Social influence as intrinsic motivation for multi-agent deep reinforcement learning. In *International conference on machine learning*, pp. 3040–3049. PMLR, 2019.

Jiechuan Jiang, Kefan Su, and Zongqing Lu. Fully Decentralized Cooperative Multi-Agent Reinforcement Learning: A Survey. *arXiv preprint arXiv:2401.04934*, 2024.

Soummya Kar, José M. F. Moura, and H. Vincent Poor. $\mathcal{QD}$-Learning: A Collaborative Distributed Strategy for Multi-Agent Reinforcement Learning Through $\mathrm{Consensus} + \mathrm{Innovations}$. *IEEE Transactions on Signal Processing*, 61(7):1848–1862, 2013. doi: 10.1109/TSP.2013.2241057.

Marcin Korecki, Damian Dailisan, and Dirk Helbing. How Well Do Reinforcement Learning Approaches Cope With Disruptions? The Case of Traffic Signal Control. *IEEE Access*, 11: 36504–36515, 2023. doi: 10.1109/ACCESS.2023.3266644.

Georgios Kotsalis, Guanghui Lan, and Tianjiao Li. Simple and Optimal Methods for Stochastic Variational Inequalities, II: Markovian Noise and Policy Evaluation in Reinforcement Learning. *SIAM Journal on Optimization*, 32(2):1120–1155, 2022. doi: 10.1137/20M1381691. URL https://doi.org/10.1137/20M1381691.

Jean-Michel Lasry and Pierre-Louis Lions. Mean Field Games. *Japanese Journal of Mathematics*, 2 (1):229–260, 2007.

Mathieu Lauriere. Numerical Methods for Mean Field Games and Mean Field Type Control. *Mean field games*, 78(221-282), 2021.

Mathieu Laurière, Sarah Perrin, Matthieu Geist, and Olivier Pietquin. Learning Mean Field Games: A Survey. *arXiv preprint arXiv:2205.12944*, 2022.

Mathieu Lauriere, Sarah Perrin, Sertan Girgin, Paul Muller, Ayush Jain, Theophile Cabannes, Georgios Piliouras, Julien Perolat, Romuald Elie, Olivier Pietquin, and Matthieu Geist. Scalable Deep Reinforcement Learning Algorithms for Mean Field Games. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 12078–12095. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/lauriere22a.html.

Stéphane Le Ménec. Swarm Guidance Based on Mean Field Game Concepts. *International Game Theory Review*, 0(0):2440008, 0. doi: 10.1142/S0219198924400085. URL https://doi.org/10.1142/S0219198924400085.

Kiyeob Lee, Desik Rengarajan, Dileep Kalathil, and Srinivas Shakkottai. Reinforcement Learning for Mean Field Games with Strategic Complementarities . In Arindam Banerjee and Kenji Fukumizu (eds.), *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 2458–2466. PMLR, 13–15 Apr 2021. URL https://proceedings.mlr.press/v130/lee21b.html.

Joel Z. Leibo, Vinicius Zambaldi, Marc Lanctot, Janusz Marecki, and Thore Graepel. Multi-Agent Reinforcement Learning in Sequential Social Dilemmas. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '17, pp. 464–473, Richland, SC, 2017. International Foundation for Autonomous Agents and Multiagent Systems.

David L. Leottau, Javier Ruiz del Solar, and Robert Babuka. Decentralized Reinforcement Learning of Robot Behaviors. *Artificial Intelligence*, 256:130–159, 2018. ISSN 0004-3702. doi: https://doi.org/10.1016/j.artint.2017.12.001. URL https://www.sciencedirect.com/science/article/pii/S0004370217301674.

Zongxi Li, A Max Reppen, and Ronnie Sircar. A Mean Field Games Model for Cryptocurrency Mining. *Management Science*, 70(4):2188–2208, 2024.

Long-Ji Lin. Self-Improving Reactive Agents Based on Reinforcement Learning, Planning and Teaching. *Mach. Learn.*, 8(3–4):293–321, may 1992. ISSN 0885-6125. doi: 10.1007/BF00992699. URL https://doi.org/10.1007/BF00992699.

Yixuan Lin, Kaiqing Zhang, Zhuoran Yang, Zhaoran Wang, Tamer Başar, Romeil Sandhu, and Ji Liu. A Communication-Efficient Multi-Agent Actor-Critic Algorithm for Distributed Reinforcement Learning. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 5562–5567, 2019. doi: 10.1109/CDC40024.2019.9029257.

Zefang Lv, Liang Xiao, Yousong Du, Guohang Niu, Chengwen Xing, and Wenyuan Xu. Multi-Agent Reinforcement Learning based UAV Swarm Communications Against Jamming. *IEEE Transactions on Wireless Communications*, pp. 1–1, 2023. doi: 10.1109/TWC.2023.3268082.

Zhuangzhuang Ma, Lei Shi, Kai Chen, Jinliang Shao, and Yuhua Cheng. Multi-Agent Bipartite Flocking Control over Cooperation-Competition Networks with Asynchronous Communications. *IEEE Transactions on Signal and Information Processing over Networks*, pp. 1–12, 2024. doi: 10.1109/TSIPN.2024.3384817.

Patrick Mannion, Jim Duggan, and Enda Howley. *An Experimental Review of Reinforcement Learning Algorithms for Adaptive Traffic Signal Control*, pp. 47–66. Springer International Publishing, Cham, 2016. ISBN 978-3-319-25808-9. doi: 10.1007/978-3-319-25808-9_4. URL https://doi.org/10.1007/978-3-319-25808-9_4.

Weichao Mao, Haoran Qiu, Chen Wang, Hubertus Franke, Zbigniew T. Kalbarczyk, Ravishankar K. Iyer, and Tamer Başar. A mean-field game approach to cloud resource management with function approximation. In *Proceedings of the 36th Conference on Advances in Neural Information Processing Systems (NIPS 2022)*, volume 36, pp. 1–12, New Orleans, LA, USA, 2022. Curran Associates, Inc.

Stephen Mcaleer, JB Lanier, Roy Fox, and Pierre Baldi. Pipeline PSRO: A Scalable Approach for Finding Approximate Nash Equilibria in Large Games. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 20238–20248. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/e9bcd1b063077573285ae1a41025f5dc-Paper.pdf.

Kevin R McKee, Ian Gemp, Brian McWilliams, Edgar A Duéñez-Guzmán, Edward Hughes, and Joel Z Leibo. Social diversity and social preferences in mixed-motive reinforcement learning. *arXiv preprint arXiv:2002.02325*, 2020.

Emily Meigs, Francesca Parise, Asuman E. Ozdaglar, and Daron Acemoglu. Optimal dynamic information provision in traffic routing. *CoRR*, abs/2001.03232, 2020. URL https://arxiv.org/abs/2001.03232.

David Mguni, Joel Jennings, and Enrique Munoz de Cote. Decentralised Learning in Systems With Many, Many Strategic Agents. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018. doi: 10.1609/aaai.v32i1.11586. URL https://ojs.aaai.org/index.php/AAAI/article/view/11586.

Li Miao, Shuai Li, Xiangjuan Wu, and Bingjie Liu. Mean-Field Stackelberg Game-Based Security Defense and Resource Optimization in Edge Computing. *Applied Sciences*, 14(9), 2024. ISSN 2076-3417. doi: 10.3390/app14093538. URL https://www.mdpi.com/2076-3417/14/9/3538.

Rajesh Mishra, Sriram Vishwanath, and Deepanshu Vasal. Model-free Reinforcement Learning for Mean Field Games. *IEEE Transactions on Control of Network Systems*, pp. 1–11, 2023. doi: 10.1109/TCNS.2023.3264934.

Rajesh K Mishra, Deepanshu Vasal, and Sriram Vishwanath. Model-free Reinforcement Learning for Non-stationary Mean Field Games. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pp. 1032–1037, 2020. doi: 10.1109/CDC42340.2020.9304340.

Jun Moon and Tamer Başar. Linear Quadratic Risk-Sensitive and Robust Mean Field Games. *IEEE Transactions on Automatic Control*, 62(3):1062–1077, 2017. doi: 10.1109/TAC.2016.2579264.

Behrang Monajemi Nejad, Sid Ahmed Attia, and Jorg Raisch. Max-consensus in a max-plus algebraic setting: The case of fixed communication topologies. In *2009 XXII International Symposium on Information, Communication and Automation Technologies*, pp. 1–7, 2009. doi: 10.1109/ICAT.2009.5348437.

Daniel Jarne Ornia, Pedro J. Zufiria, and Manuel Mazo Jr. Mean Field Behavior of Collaborative Multiagent Foragers. *IEEE Transactions on Robotics*, 38(4):2151–2165, 2022. doi: 10.1109/TRO.2022.3152691.

James Orr and Ayan Dutta. Multi-Agent Deep Reinforcement Learning for Multi-Robot Applications: A Survey. *Sensors*, 23(7), 2023. ISSN 1424-8220. doi: 10.3390/s23073625. URL `https://www.mdpi.com/1424-8220/23/7/3625`.

Francesca Parise, Sergio Grammatico, Basilio Gentile, and John Lygeros. Network Aggregative Games and Distributed Mean Field Control via Consensus Theory. *arXiv preprint arXiv:1506.07719*, 2015.

Bhrij Patel, Wesley A Suttle, Alec Koppel, Vaneet Aggarwal, Brian M Sadler, Amrit Singh Bedi, and Dinesh Manocha. Global Optimality without Mixing Time Oracles in Average-reward RL via Multi-level Actor-Critic. *arXiv preprint arXiv:2403.11925*, 2024.

Julien Perolat, Sarah Perrin, Romuald Elie, Mathieu Laurière, Georgios Piliouras, Matthieu Geist, Karl Tuyls, and Olivier Pietquin. Scaling up Mean Field Games with Online Mirror Descent. *arXiv preprint arXiv:2103.00623*, 2021.

Julien Pérolat, Sarah Perrin, Romuald Elie, Mathieu Laurière, Georgios Piliouras, Matthieu Geist, Karl Tuyls, and Olivier Pietquin. Scaling Mean Field Games by Online Mirror Descent. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '22, pp. 1028–1037, Richland, SC, 2022. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9781450392136.

Sarah Perrin, Julien Pérolat, Mathieu Laurière, Matthieu Geist, Romuald Elie, and Olivier Pietquin. Fictitious Play for Mean Field Games: Continuous Time Analysis and Applications. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Sarah Perrin, Mathieu Laurière, Julien Pérolat, Matthieu Geist, Romuald Élie, and Olivier Pietquin. Mean Field Games Flock! The Reinforcement Learning Way. In *IJCAI*, 2021.

Sarah Perrin, Mathieu Laurière, Julien Pérolat, Romuald Élie, Matthieu Geist, and Olivier Pietquin. Generalization in mean field games by learning master policies. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 9413–9421, 2022.

Nancirose Piazza, Vahid Behzadan, and Stefan Sarkadi. The Power in Communication: Power Regularization of Communication for Autonomy in Cooperative Multi-Agent Reinforcement Learning. *arXiv preprint arXiv:2404.06387*, 2024.

Abraham Prieto, Francisco Bellas, Pedro Trueba, and Richard J Duro. Real-time optimization of dynamic problems through distributed embodied evolution. *Integrated Computer-Aided Engineering*, 23(3):237–253, 2016.

Shreevatsa Rajagopalan and Devavrat Shah. Distributed Averaging in Dynamic Networks. In *Proceedings of the ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '10, pp. 369–370, New York, NY, USA, 2010. Association for Computing Machinery. ISBN 9781450300384. doi: 10.1145/1811039.1811091. URL `https://doi.org/10.1145/1811039.1811091`.

Navid Rashedi, Mohammad Amin Tajeddini, and Hamed Kebriaei. Markov game approach for multi-agent competitive bidding strategies in electricity market. *IET Generation, Transmission & Distribution*, 10:3756–3763(7), November 2016. ISSN 1751-8687. URL `https://digital-library.theiet.org/content/journals/10.1049/iet-gtd.2016.0075`.

Naci Saldi, Tamer Başar, and Maxim Raginsky. Markov–Nash Equilibria in Mean-Field Games with Discounted Cost. *SIAM Journal on Control and Optimization*, 56(6):4256–4287, 2018. doi: 10.1137/17M1112583. URL https://doi.org/10.1137/17M1112583.

Mikayel Samvelyan, Tabish Rashid, Christian Schroeder de Witt, Gregory Farquhar, Nantas Nardelli, Tim G. J. Rudner, Chia-Man Hung, Philip H. S. Torr, Jakob Foerster, and Shimon Whiteson. The StarCraft Multi-Agent Challenge. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '19, pp. 2186–2188, Richland, SC, 2019. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9781450363099.

Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. Safe, multi-agent, reinforcement learning for autonomous driving. *arXiv preprint arXiv:1610.03295*, 2016.

Ali Shavandi and Majid Khedmati. A multi-agent deep reinforcement learning framework for algorithmic trading in financial markets. *Expert Systems with Applications*, 208:118124, 2022. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2022.118124. URL https://www.sciencedirect.com/science/article/pii/S0957417422013082.

Shigen Shen, Chenpeng Cai, Yizhou Shen, Xiaoping Wu, Wenlong Ke, and Shui Yu. MFGD3QN: Enhancing Edge Intelligence Defense against DDoS with Mean-Field Games and Dueling Double Deep Q-network. *IEEE Internet of Things Journal*, pp. 1–1, 2024. doi: 10.1109/JIOT.2024.3387090.

Hamid Shiri, Jihong Park, and Mehdi Bennis. Massive Autonomous UAV Path Planning: A Neural Network Based Mean-Field Game Theoretic Approach. In *2019 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6. IEEE, 2019.

Kefan Su and Zongqing Lu. Divergence-Regularized Multi-Agent Actor-Critic. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 20580–20603. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/su22b.html.

Jayakumar Subramanian and Aditya Mahajan. Reinforcement Learning in Stationary Mean-Field Games. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '19, pp. 251–259, Richland, SC, 2019. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9781450363099.

Sriram Ganapathi Subramanian, Matthew E Taylor, Mark Crowley, and Pascal Poupart. Decentralized Mean Field Games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 9439–9447, 2022.

Wesley Suttle, Zhuoran Yang, Kaiqing Zhang, Zhaoran Wang, Tamer Başar, and Ji Liu. A Multi-Agent Off-Policy Actor-Critic Algorithm for Distributed Reinforcement Learning. *IFAC-PapersOnLine*, 53(2):1549–1554, 2020.

Richard S Sutton and Andrew G Barto. *Reinforcement Learning: An Introduction*. MIT press, 2018.

Huaze Tang, Yuanquan Hu, Fanfan Zhao, Junji Yan, Ting Dong, and Wenbo Ding. M3ARL: Moment-Embedded Mean-Field Multi-Agent Reinforcement Learning for Continuous Action Space. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7250–7254, 2024. doi: 10.1109/ICASSP48485.2024.10448058.

Hamidou Tembine, Raul Tempone, and Pedro Vilanova. Mean-Field Learning: a Survey. *arXiv preprint arXiv:1210.4657*, 2012.

Amoolya Tirumalai and John S. Baras. A Robust Mean-field Game of Boltzmann-Vlasov-like Traffic Flow. In *2022 American Control Conference (ACC)*, pp. 556–561, 2022. doi: 10.23919/ACC53348.2022.9867331.

Noureddine Toumi, Roland Malhame, and Jerome Le Ny. A mean field game approach for a class of linear quadratic discrete choice problems with congestion avoidance. *Automatica*, 160:111420, 2024. ISSN 0005-1098. doi: https://doi.org/10.1016/j.automatica.2023.111420. URL https://www.sciencedirect.com/science/article/pii/S0005109823005873.

Torsten Trimborn, Martin Frank, and Stephan Martin. Mean field limit of a behavioral financial market model. *Physica A: Statistical Mechanics and its Applications*, 505:613–631, 2018. ISSN 0378-4371. doi: https://doi.org/10.1016/j.physa.2018.03.079. URL https://www.sciencedirect.com/science/article/pii/S0378437118303984.

Pedro Trueba, Abraham Prieto, Francisco Bellas, and Richard J. Duro. Embodied Evolution for Collective Indoor Surveillance and Location. In *Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation*, GECCO Companion '15, pp. 1241–1242, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450334884. doi: 10.1145/2739482.2768490. URL https://doi.org/10.1145/2739482.2768490.

Nino Vieillard, Olivier Pietquin, and Matthieu Geist. Munchausen Reinforcement Learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 4235–4246. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/2c6a0bae0f071cbbf0bb3d5b11d90a82-Paper.pdf.

Oriol Vinyals, Igor Babuschkin, Junyoung Chung, Michael Mathieu, Max Jaderberg, Wojtek Czarnecki, Andrew Dudzik, Aja Huang, Petko Georgiev, Richard Powell, Timo Ewalds, Dan Horgan, Manuel Kroiss, Ivo Danihelka, John Agapiou, Junhyuk Oh, Valentin Dalibard, David Choi, Laurent Sifre, Yury Sulsky, Sasha Vezhnevets, James Molloy, Trevor Cai, David Budden, Tom Paine, Caglar Gulcehre, Ziyu Wang, Tobias Pfaff, Toby Pohlen, Dani Yogatama, Julia Cohen, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy Lillicrap, Chris Apps, Koray Kavukcuoglu, Demis Hassabis, and David Silver. AlphaStar: Mastering the Real-Time Strategy Game StarCraft II. https://deepmind.com/blog/alphastar-mastering-real-time-strategy-game-starcraft-ii/, 2019a.

Oriol Vinyals, Igor Babuschkin, Wojciech M. Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H. Choi, Richard Powell, Timo Ewalds, Petko Georgiev, Junhyuk Oh, Dan Horgan, Manuel Kroiss, Ivo Danihelka, Aja Huang, L. Sifre, Trevor Cai, John P. Agapiou, Max Jaderberg, Alexander Sasha Vezhnevets, Rémi Leblond, Tobias Pohlen, Valentin Dalibard, David Budden, Yury Sulsky, James Molloy, Tom Le Paine, Caglar Gulcehre, Ziyun Wang, Tobias Pfaff, Yuhuai Wu, Roman Ring, Dani Yogatama, Dario Wünsch, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy P. Lillicrap, Koray Kavukcuoglu, Demis Hassabis, Chris Apps, and David Silver. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, pp. 1–5, 2019b.

Hoi-To Wai, Zhuoran Yang, Zhaoran Wang, and Mingyi Hong. Multi-Agent Reinforcement Learning via Double Averaging Primal-Dual Optimization. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pp. 9672–9683, Red Hook, NY, USA, 2018. Curran Associates Inc.

Lingxiao Wang, Zhuoran Yang, and Zhaoran Wang. Breaking the Curse of Many Agents: Provable Mean Embedding Q-Iteration for Mean-Field Reinforcement Learning. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020a.

Ximing Wang, Yuhua Xu, Jin Chen, Chunguo Li, Xin Liu, Dianxiong Liu, and Yifan Xu. Mean Field Reinforcement Learning Based Anti-Jamming Communications for Ultra-Dense Internet of Things in 6G. In *2020 International Conference on Wireless Communications and Signal Processing (WCSP)*, pp. 195–200, 2020b. doi: 10.1109/WCSP49889.2020.9299742.

Yao Wang, Chungang Yang, Tong Li, Xinru Mi, Lixin Li, and Zhu Han. A Survey On Mean-Field Game for Dynamic Management and Control in Space-Air-Ground Network. *IEEE Communications Surveys & Tutorials*, pp. 1–1, 2024. doi: 10.1109/COMST.2024.3393369.

Samuel Wiggins, Yuan Meng, Rajgopal Kannan, and Viktor Prasanna. Characterizing Speed Performance of Multi-Agent Reinforcement Learning. *arXiv preprint arXiv:2309.07108*, 2023.

Peiliang Wu, Liqiang Tian, Qian Zhang, Bingyi Mao, and Wenbai Chen. MARRGM: Learning Framework for Multi-agent Reinforcement Learning via Reinforcement Recommendation and Group Modification. *IEEE Robotics and Automation Letters*, pp. 1–8, 2024a. doi: 10.1109/LRA.2024.3389813.

Zida Wu, Mathieu Laurière, Samuel Jia Cong Chua, Matthieu Geist, Olivier Pietquin, and Ankur Mehta. Population-aware Online Mirror Descent for Mean-Field Games by Deep Reinforcement Learning. *arXiv preprint arXiv:2403.03552*, 2024b.

Qiaomin Xie, Zhuoran Yang, Zhaoran Wang, and Andreea Minca. Learning While Playing in Mean-Field Games: Convergence and Optimality. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 11436–11447. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/xie21g.html.

Linjie Xu, Zichuan Liu, Alexander Dockhorn, Diego Perez-Liebana, Jinyu Wang, Lei Song, and Jiang Bian. Higher Replay Ratio Empowers Sample-Efficient Multi-Agent Reinforcement Learning. *arXiv preprint arXiv:2404.09715*, 2024.

Chungang Yang, Haoxiang Dai, Jiandong Li, Yue Zhang, and Zhu Han. Distributed Interference-Aware Power Control in Ultra-Dense Small Cell Networks: A Robust Mean Field Game. *IEEE Access*, 6:12608–12619, 2018a. doi: 10.1109/ACCESS.2018.2799138.

Yaodong Yang, Rui Luo, Minne Li, Ming Zhou, Weinan Zhang, and Jun Wang. Mean Field Multi-Agent Reinforcement Learning. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 5571–5580. PMLR, 10–15 Jul 2018b. URL https://proceedings.mlr.press/v80/yang18d.html.

Yaoqi Yang, Bangning Zhang, Daoxing Guo, Renhui Xu, Neeraj Kumar, and Weizheng Wang. Mean Field Game and Broadcast Encryption-Based Joint Data Freshness Optimization and Privacy Preservation for Mobile Crowdsensing. *IEEE Transactions on Vehicular Technology*, 72(11): 14860–14874, 2023. doi: 10.1109/TVT.2023.3282694.

Batuhan Yardim, Semih Cayci, Matthieu Geist, and Niao He. Policy Mirror Ascent for Efficient and Independent Learning in Mean Field Games. In *International Conference on Machine Learning*, pp. 39722–39754. PMLR, 2023.

Batuhan Yardim, Artur Goldman, and Niao He. When is Mean-Field Reinforcement Learning Tractable and Relevant? *arXiv preprint arXiv:2402.05757*, 2024.

Bora Yongacoglu, Gürdal Arslan, and Serdar Yüksel. Independent Learning in Mean-Field Games: Satisficing Paths and Convergence to Subjective Equilibria. *arXiv preprint arXiv:2209.05703*, 2022a.

Bora Yongacoglu, Gürdal Arslan, and Serdar Yüksel. Independent Learning and Subjectivity in Mean-Field Games. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pp. 2845–2850, 2022b. doi: 10.1109/CDC51059.2022.9992399.

Hidekazu Yoshioka, Motoh Tsujimura, and Yumi Yoshioka. Numerical analysis of an extended mean field game for harvesting common fishery resource. *Computers & Mathematics with Applications*, 165:88–105, 2024. ISSN 0898-1221. doi: https://doi.org/10.1016/j.camwa.2024.04.003. URL https://www.sciencedirect.com/science/article/pii/S0898122124001615.

Hanfei Yu, Jian Li, Yang Hua, Xu Yuan, and Hao Wang. Cheaper and faster: Distributed deep reinforcement learning with serverless computing. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(15):16539–16547, Mar. 2024. doi: 10.1609/aaai.v38i15.29592. URL https://ojs.aaai.org/index.php/AAAI/article/view/29592.

Xiang Yu and Fengyi Yuan. Time-inconsistent mean-field stopping problems: A regularized equilibrium approach. *arXiv preprint arXiv:2311.00381*, 2023.

Muhammad Aneeq Uz Zaman, Alec Koppel, Sujay Bhatt, and Tamer Basar. Oracle-free Reinforcement Learning in Mean-Field Games along a Single Sample Path. In *International Conference on Artificial Intelligence and Statistics*, pp. 10178–10206. PMLR, 2023.

Kaiqing Zhang, Zhuoran Yang, Han Liu, Tong Zhang, and Tamer Basar. Fully Decentralized Multi-Agent Reinforcement Learning with Networked Agents. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 5872–5881. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/zhang18n.html.

Kaiqing Zhang, Yang Liu, Ji Liu, Mingyan Liu, and Tamer Basar. Distributed learning of average belief over networks using sequential observations. *Automatica*, 115:108857, 2020. ISSN 0005-1098. doi: https://doi.org/10.1016/j.automatica.2020.108857. URL https://www.sciencedirect.com/science/article/pii/S0005109820300558.

Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. Decentralized Multi-Agent Reinforcement Learning with Networked Agents: Recent Advances. *Frontiers of Information Technology & Electronic Engineering*, 22(6):802–814, 2021a.

Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. *"Multi-Agent Reinforcement Learning: A Selective Overview of Theories and Algorithms"*, pp. 321–384. Springer International Publishing, Cham, 2021b. ISBN 978-3-030-60990-0. doi: 10.1007/978-3-030-60990-0_12. URL https://doi.org/10.1007/978-3-030-60990-0_12.

Shangtong Zhang and Richard S Sutton. A deeper look at experience replay. *arXiv preprint arXiv:1712.01275*, 2017.

Lianmin Zheng, Jiacheng Yang, Han Cai, Ming Zhou, Weinan Zhang, Jun Wang, and Yong Yu. MAgent: A Many-Agent Reinforcement Learning Platform for Artificial Collective Intelligence. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

TECHNICAL APPENDICES

A FURTHER DEFINITIONS AND ASSUMPTIONS FOR THEOREMS IN SEC. 3.3

Assumption 1 (Lipschitz continuity of P and R, from Assumption 1, Yardim et al. (2023))**.** *There exist constants* $K_\mu, K_s, K_a, L_\mu, L_s, L_a \in \mathbb{R}_{\geq 0}$ *such that* $\forall s, s' \in \mathcal{S}, \forall a, a' \in \mathcal{A}, \forall \mu, \mu' \in \Delta_{\mathcal{S}}$,

$$||P(\cdot|s,a,\mu) - P(\cdot|s',a',\mu')||_1 \leq K_\mu||\mu - \mu'||_1 + K_s d(s,s') + K_a d(a,a'),$$
$$|R(s,a,\mu) - R(s',a',\mu')| \leq L_\mu||\mu - \mu'||_1 + L_s d(s,s') + L_a d(a,a').$$

Definition 9 (Population update operator, from Def. 3.1, Yardim et al. (2023))**.** *The single-step population update operator* $\Gamma_{pop} : \Delta_{\mathcal{S}} \times \Pi \to \Delta_{\mathcal{S}}$ *is defined as,* $\forall s \in \mathcal{S}$:

$$\Gamma_{pop}(\mu,\pi)(s) := \sum_{s' \in \mathcal{S}} \sum_{a' \in \mathcal{A}} \mu(s')\pi(a'|s')P(s|s',a',\mu).$$

Let us use the short hand notation $\Gamma^n_{pop}(\mu,\pi) := \underbrace{\Gamma_{pop}(\ldots\Gamma_{pop}(\Gamma_{pop}(\mu,\pi),\pi),\ldots,\pi)}_{n\ times}$.

We recall that Γ_{pop} is known to be Lipschitz:

Lemma 1 (Lipschitz population updates, from Lem. 3.2, Yardim et al. (2023))**.** Γ_{pop} *is Lipschitz with*

$$||\Gamma_{pop}(\mu,\pi) - \Gamma_{pop}(\mu',\pi')||_1 \leq L_{pop,\mu}||\mu - \mu'||_1 + \frac{K_a}{2}||\pi - \pi'||_1,$$

where $L_{pop,\mu} := \left(\frac{K_s}{2} + \frac{K_a}{2} + K_\mu\right)$, $\forall \pi \in \Pi, \mu \in \Delta_{\mathcal{S}}$.

For stationary MFGs the population distribution must be stable with respect to a policy, requiring that $\Gamma_{pop}(\cdot,\pi)$ is contractive $\forall \pi \in \Pi$:

Assumption 2 (Stable population, from Assumption 2, Yardim et al. (2023))**.** *Population updates are stable, i.e.* $L_{pop,\mu} < 1$.

Definition 10 (Stable population operator Γ^∞_{pop}, from Def. 3.3, Yardim et al. (2023))**.** *Given Assumption 2, the operator* $\Gamma^\infty_{pop} : \Pi \to \Delta_{\mathcal{S}}$ *maps a given policy to its unique stable population distribution such that* $\Gamma_{pop}(\Gamma^\infty_{pop}(\pi),\pi) = \Gamma^\infty_{pop}(\pi)$, *i.e. the unique fixed point of* $\Gamma_{pop}(\cdot,\pi) : \Delta_{\mathcal{S}} \to \Delta_{\mathcal{S}}$.

Definition 11 (Q_h and q_h functions)**.** *We define, for any pair* $(s,a) \in \mathcal{S} \times \mathcal{A}$:

$$Q_h(s,a|\pi,\mu) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t (R(s_t,a_t,\mu) + h(\pi(s_t))) \middle| \substack{s_0=s,\ s_{t+1}\sim P(\cdot|s_t,a_t,\mu),\\ a_0=a,\ a_{t+1}\sim\pi(\cdot|s_{t+1})}, \forall t \geq 0\right]$$

and

$$q_h(s,a|\pi,\mu) := R(s,a,\mu) + \gamma \sum_{s',a'} P(s'|s,a,\mu)\pi(a'|s')Q_h(s',a'|\pi,\mu).$$

Definition 12 (Γ_q operator)**.** *The operator* $\Gamma_q : \Pi \times \Delta_{\mathcal{S}} \to \mathcal{Q}$ *mapping population-policy pairs to Q-functions is defined as* $\Gamma_q(\pi,\mu) := q_h(\cdot,\cdot|\pi,\mu) \in \mathcal{Q}\ \forall \pi \in \Pi, \mu \in \Delta_{\mathcal{S}}$.

We also assume that the regulariser h ensures that all actions at all states are explored with non-zero probability:

Assumption 3 (Persistence of excitation, from Assumption 3, Yardim et al. (2023))**.** *We assume there exists* $p_{inf} > 0$ *such that:*

1. $\pi_{\max}(a|s) \geq p_{inf}\ \forall s \in \mathcal{S}, a \in \mathcal{A}$,
2. *For any* $\pi \in \Pi$ *and* $q \in \mathcal{Q}$ *that satisfy,* $\forall (s,a) \in \mathcal{S} \times \mathcal{A}$, $\pi(a|s) \geq p_{inf}$ *and* $0 \leq q(s,a) \leq Q_{\max}$, *it holds that* $\Gamma^{md}_\eta(q,\pi)(a|s) \geq p_{inf}$, $\forall (s,a) \in \mathcal{S} \times \mathcal{A}$.

Assumption 4 (Sufficient mixing, from Assumption 4, Yardim et al. (2023))**.** *For any* $\pi \in \Pi$ *satisfying* $\pi(a|s) \geq p_{inf} > 0\ \forall s \in \mathcal{S}, a \in \mathcal{A}$, *and any initial states* $\{s^i_0\}_i \in \mathcal{S}^N$, *there exist* $T_{mix} > 0, \delta_{mix} > 0$ *such that* $\mathbb{P}(s^j_{T_{mix}} = s'|\{s^i_0\}_i) \geq \delta_{mix}$, $\forall s' \in \mathcal{S}, j \in [N]$.

Definition 13 (Nested learning operator). *For a learning rate $\eta > 0$, $\Gamma_\eta : \Pi \to \Pi$ is defined as*

$$\Gamma_\eta(\pi) := \Gamma_\eta^{md}(\Gamma_q(\pi, \Gamma_{pop}^{\infty}(\pi)), \pi).$$

Lemma 2 (Lipschitz continuity of Γ_η, from Lem. 3.7, (Yardim et al., 2023))). *For any $\eta > 0$, the operator $\Gamma_\eta : \Pi \to \Pi$ is Lipschitz with constant L_{Γ_η} on $(\Pi, ||\cdot||_1)$.*

B Full theorems and complete proofs

B.1 Sample guarantees of independent-learning case

Lemma 3 (Independent learning, from Thm. 4.5, Yardim et al. (2023)). *Define $t_0 := \frac{16(1+\gamma)^2}{((1-\gamma)\delta_{mix}p_{inf})^2}$. Assume that Assumptions 1, 2, 3 and 4 hold, that $\eta > 0$ satisfies $L_{\Gamma_\eta} < 1$, and that π^* is the unique MFG-NE. The learning rates are $\beta_m = \frac{2}{(1-\gamma)(t_0+m-1)}$ $\forall m \geq 0$, and let $\varepsilon > 0$ be arbitrary. There exists a problem-dependent constant $a \in [0, \infty)$ such that if $K = \frac{\log 8\varepsilon^{-1}}{\log L_{\Gamma_\eta}^{-1}}$, $M_{pg} > \mathcal{O}(\varepsilon^{-2-a})$ and $M_{td} > \mathcal{O}(log^2 \varepsilon^{-1})$, then the random output $\{\pi_K^i\}_i$ of Alg. 1 when run with $C = 0$ (such that there is no communication) satisfies for all agents $i = 1, \ldots, N$,*

$$\mathbb{E}\left[||\pi_K^i - \pi^*||_1\right] \leq \varepsilon + \mathcal{O}\left(\frac{1}{\sqrt{N}}\right).$$

B.2 Full version of Thm. 1

Theorem 1 (Networked learning with random adoption). *For p_{inf} and δ_{mix} defined in Assumptions 3 and 4 respectively, define $t_0 := \frac{16(1+\gamma)^2}{((1-\gamma)\delta_{mix}p_{inf})^2}$. Assume that Assumptions 1, 2, 3 and 4 hold, and that π^* is the unique MFG-NE policy. For L_{Γ_η} defined in Lem. 2, we assume $\eta > 0$ satisfies $L_{\Gamma_\eta} < 1$. The learning rates are $\beta_m = \frac{2}{(1-\gamma)(t_0+m-1)}$ $\forall m \geq 0$, and let $\varepsilon > 0$ be arbitrary. Assume also that $C > 0$, with $\tau_k \to \infty$. There exists a problem-dependent constant $a \in [0, \infty)$ such that if $K = \frac{\log 8\varepsilon^{-1}}{\log L_{\Gamma_\eta}^{-1}}$, $M_{pg} > \mathcal{O}(\varepsilon^{-2-a})$ and $M_{td} > \mathcal{O}(log^2 \varepsilon^{-1})$, then the random output $\{\pi_K^i\}_i$ of Alg. 1 preserves the sample guarantees of the independent-learning case given in Lem. 3, i.e. the output satisfies, for all agents $i = 1, \ldots, N$,*

$$\mathbb{E}\left[||\pi_K^i - \pi^*||_1\right] \leq \varepsilon + \mathcal{O}\left(\frac{1}{\sqrt{N}}\right).$$

(Proof in Appx. B.3.)

B.3 Proof of Thm. 1

Proof. If $\tau_k \to \infty$, the softmax function that defines the probability of a received policy being adopted in Line 15 of Alg. 1 gives a uniform distribution. Policies are thus exchanged at random between communicating agents an arbitrary $C > 0$ times, which does not affect the random output of the algorithm, such that the random output satisfies the same expectation as if $C = 0$. □

B.4 Conditional TD learning from a single continuous run of the empirical distribution of N agents

Lemma 4 (Conditional TD learning from a single continuous run of the empirical distribution of N agents, from Thm. 4.2, Yardim et al. (2023)). *Define $t_0 := \frac{16(1+\gamma)^2}{((1-\gamma)\delta_{mix}p_{inf})^2}$. Assume Assumption 4 holds and let policies $\{\pi^i\}_i$ be given such that $\pi^i(a|s) \geq p_{inf}$ $\forall i$. Assume Lines 3-9 of Alg. 1 are run with policies $\{\pi^i\}_i$, arbitrary initial agents states $\{s_0^i\}_i$, learning rates $\beta_m = \frac{2}{(1-\gamma)(t_0+m-1)}$, $\forall m \geq 0$ and $M_{pg} > \mathcal{O}(\varepsilon^{-2})$, $M_{td} > \mathcal{O}(\log \varepsilon^{-1})$. If $\bar{\pi} \in \Pi$ is an arbitrary policy, $\Delta := \sum_{i=1}^N ||\pi^i - \bar{\pi}||_1$ and $Q^* := Q_h(\cdot, \cdot|\bar{\pi}, \mu_{\bar{\pi}})$, then the random output $\hat{Q}_{M_{pg}}^i$ of Lines 3-9 satisfies*

$$\mathbb{E}\left[||\hat{Q}_{M_{pg}}^i - Q^*||_\infty\right] \leq \varepsilon + \mathcal{O}\left(\frac{1}{\sqrt{N}} + \frac{1}{N}\Delta + ||\pi^i - \bar{\pi}||_1\right).$$

B.5 Full version of Thm. 2

Theorem 2 (Networked learning with non-random adoption)**.** *Assume that Assumptions 1, 2, 3 and 4 hold, and that Alg. 1 is run with learning rates and constants as defined in Thm. 1, except now $\tau_k \in \mathbb{R}_{>0}$. Assume that σ^i_{k+1} is generated uniquely for each i, in a manner independent of any metric related to π^i_{k+1}, e.g. σ^i_{k+1} is random or related only to the index i (so as not to bias the spread of any particular policy). Let the random output of this Algorithm be denoted as $\{\pi^{i,net}_K\}_i$. Also consider an independent-learning version of the algorithm (i.e. with the same parameters except $C = 0$) and denote its random output $\{\pi^{i,ind}_K\}_i$; and a centralised version of the algorithm with the same parameters (see Rem. 3) and denote its random output as π^{cent}_K. Then for all agents $i = 1, \dots, N$, the random outputs $\{\pi^{i,net}_K\}_i$, $\{\pi^{i,ind}_K\}_i$ and π^{cent}_K satisfy the following relations, where ub_{net}, ub_{ind} and ub_{cent} are respective upper bounds for each case:*

$$\mathbb{E}\left[||\pi^{cent}_K - \pi^*||_1\right] \leq ub_{cent}, \quad \mathbb{E}\left[||\pi^{i,net}_K - \pi^*||_1\right] \leq ub_{net}, \quad \mathbb{E}\left[||\pi^{i,ind}_K - \pi^*||_1\right] \leq ub_{ind},$$

$$\textit{where} \quad ub_{cent} \leq ub_{net} \leq ub_{ind} = \varepsilon + \mathcal{O}\left(\frac{1}{\sqrt{N}}\right).$$

(Proof in Appx. B.6.)

B.6 Proof of Thm. 2

Proof. We build off the proof of our Lem. 3, given in Thm. D.9 of Yardim et al. (2023), where the sample guarantees of the independent case are worse than those of the centralised algorithm as a result of the divergence between the decentralised policies due to the stochasticity of the PMA updates. For an arbitrary policy $\bar{\pi}_k \in \Pi$, for all $k = 0, 1, \dots, K$ define the policy divergence as the random variable $\Delta_k := \sum_{i=1}^N ||\pi^i_k - \bar{\pi}_k||_1$. We can say that $\Delta_{k,cent} = 0 \; \forall k$ is the divergence in the centralised case, while in the networked case the policy divergence is $\Delta_{k+1,c}$ after communication round $c \in 1, \dots, C$. The independent case is equivalent to the scenario when $C = 0$, such that its policy divergence can be written $\Delta_{k+1,0}$.

For $\tau_k \in \mathbb{R}_{>0}$, the adoption probability $\Pr\left(\text{adopted}^i = \sigma^j_{k+1}\right) = \frac{\exp(\sigma^j_{k+1}/\tau_k)}{\sum_{x=1}^{[J^i_t]} \exp(\sigma^x_{k+1}/\tau_k)}$ (as in Line 15 of Alg. 1) is higher for some $j \in J^i_t$ than for others. This means that for $c > 0$ for which there are communication links in the population, in expectation the number of unique policies in the population will decrease, as it will likely become that $\pi^i_{k+1} = \pi^j_{k+1}$ for some $i, j \in \{1, \dots, N\}$. As such, $\Delta_{k+1,cent} \leq \mathbb{E}[\Delta_{k+1,C}] \leq \mathbb{E}[\Delta_{k+1,0}]$, i.e. the policy divergence in the independent-learning case is expected to be greater than or equal to that of the networked case.

The proof of Lem. 3 given in Thm. D.9 of Yardim et al. (2023) ends with, for constants χ and ξ,

$$\mathbb{E}\left[||\pi^i_K - \pi^*||_1\right] \leq 2L^K_{\Gamma_\eta} + \frac{\chi}{1 - L_{\Gamma_\eta}} + \xi \sum_{k=1}^{K-1} L^{K-k-1}_{\Gamma_\eta} \mathbb{E}[\Delta_k],$$

where in our context the policy divergence in the independent case $\mathbb{E}[\Delta_{k+1}]$ is equivalent to $\mathbb{E}[\Delta_{k+1,C}]$ when $C = 0$, i.e. $\mathbb{E}[\Delta_{k+1,0}]$.

Thus, for all agents $i = 1, \dots, N$, the random outputs $\{\pi^{i,net}_K\}_i$, $\{\pi^{i,ind}_K\}_i$ and π^{cent}_K satisfy:

$$\mathbb{E}\left[||\pi^{i,ind}_K - \pi^*||_1\right] \leq ub_{ind} = 2L^K_{\Gamma_\eta} + \frac{\chi}{1 - L_{\Gamma_\eta}} + \xi \sum_{k=1}^{K-1} L^{K-k-1}_{\Gamma_\eta} \mathbb{E}[\Delta_{k,0}],$$

$$\mathbb{E}\left[||\pi^{i,net}_K - \pi^*||_1\right] \leq ub_{net} = 2L^K_{\Gamma_\eta} + \frac{\chi}{1 - L_{\Gamma_\eta}} + \xi \sum_{k=1}^{K-1} L^{K-k-1}_{\Gamma_\eta} \mathbb{E}[\Delta_{k,C}],$$

$$\mathbb{E}\left[||\pi^{cent}_K - \pi^*||_1\right] \leq ub_{cent} = 2L^K_{\Gamma_\eta} + \frac{\chi}{1 - L_{\Gamma_\eta}} + \xi \sum_{k=1}^{K-1} L^{K-k-1}_{\Gamma_\eta} \mathbb{E}[\Delta_{k,cent}].$$

Since $\Delta_{k+1,cent} \leq \mathbb{E}\left[\Delta_{k+1,C}\right] \leq \mathbb{E}\left[\Delta_{k+1,0}\right]$, we obtain our result, i.e.

$$ub_{cent} \ \leq \ ub_{net} \leq \ ub_{ind} \ = \ \varepsilon + \mathcal{O}\left(\frac{1}{\sqrt{N}}\right).$$

□

Remark 5. *It may help to see that our result is a consequence of the following. Denote $\hat{Q}^{i,net}_{M_{pg}}$, $\hat{Q}^{i,ind}_{M_{pg}}$ and $\hat{Q}^{cent}_{M_{pg}}$ as the random outputs of Lines 3-9 of Alg. 1 in the networked, independent and centralised cases respectively. In Lem. 4, we can see that policy divergence gives bias terms in the estimation of the Q-value. Therefore, given $\Delta_{k+1,cent} \leq \mathbb{E}\left[\Delta_{k+1,C}\right] \leq \mathbb{E}\left[\Delta_{k+1,0}\right]$, we can also say*

$$\mathbb{E}\left[||\hat{Q}^{cent}_{M_{pg}} - Q^*||_\infty\right] \ \leq \ \mathbb{E}\left[||\hat{Q}^{i,net}_{M_{pg}} - Q^*||_\infty\right] \ \leq \ \mathbb{E}\left[||\hat{Q}^{i,ind}_{M_{pg}} - Q^*||_\infty\right].$$

In other words, the networked case will require the same or fewer outer iterations K to reduce the variance caused by this bias than the independent case requires (where the bias is non-vanishing), and the same or more iterations than the centralised case requires.

B.7 Proof of Thm. 3

Proof. From the proof of Thm. 2 in Appx. B.6 we have:

$$\mathbb{E}\left[||\pi^{i,ind}_K - \pi^*||_1\right] \leq ub_{ind} = 2L^K_{\Gamma_\eta} + \frac{\chi}{1-L_{\Gamma_\eta}} + \xi \sum_{k=1}^{K-1} L^{K-k-1}_{\Gamma_\eta}\mathbb{E}\left[\Delta_{k,0}\right],$$

$$\mathbb{E}\left[||\pi^{i,net}_K - \pi^*||_1\right] \leq ub_{net} = 2L^K_{\Gamma_\eta} + \frac{\chi}{1-L_{\Gamma_\eta}} + \xi \sum_{k=1}^{K-1} L^{K-k-1}_{\Gamma_\eta}\mathbb{E}\left[\Delta_{k,C}\right],$$

$$\mathbb{E}\left[||\pi^{cent}_K - \pi^*||_1\right] \leq ub_{cent} = 2L^K_{\Gamma_\eta} + \frac{\chi}{1-L_{\Gamma_\eta}} + \xi \sum_{k=1}^{K-1} L^{K-k-1}_{\Gamma_\eta}\mathbb{E}\left[\Delta_{k,cent}\right].$$

With a static, connected network and $\tau_k \to 0 \ \forall k$, max-consensus is always reached after $C = d_\mathcal{G}$ communication rounds, such that $\Delta_{k,cent} = \Delta_{k,d_\mathcal{G}} = 0$ (Nejad et al., 2009). The convergence rate of the max-consensus algorithm is $\frac{1}{d_\mathcal{G}}$ (Nejad et al., 2009), i.e. there is a decrease in the number of policies in the population by a factor of approximately $\frac{1}{d_\mathcal{G}}$ with each communication round up to $C = d_\mathcal{G}$, and therefore there is also a decrease in the policy divergence $\mathbb{E}\left[\Delta_{k,c}\right]$ by a factor of approximately $\frac{1}{d_\mathcal{G}}$ with each communication round. Thus

$$\mathbb{E}\left[\Delta_{k,c+1}\right] \approx \mathbb{E}\left[\Delta_{k,c}\right] - \left(\mathbb{E}\left[\Delta_{k,c}\right] \times \frac{1}{d_\mathcal{G}}\right), \text{simplifying to}$$

$$\mathbb{E}\left[\Delta_{k,c+1}\right] \approx \mathbb{E}\left[\Delta_{k,c}\right] \times \left(1 - \frac{1}{d_\mathcal{G}}\right).$$

By induction

$$\mathbb{E}\left[\Delta_{k,C}\right] \approx \mathbb{E}\left[\Delta_{k,0}\right] \times \left(\left(1 - \frac{1}{d_\mathcal{G}}\right)^C\right),$$

however, we know that $\Delta_{k,d_\mathcal{G}} = 0$, so we can more accurately use the piecewise function $f(C, d_\mathcal{G})$, defined as:

$$f(C, d_\mathcal{G}) = \begin{cases} \left(\left(1 - \frac{1}{d_\mathcal{G}}\right)^C\right) & \text{if } C < d_\mathcal{G}, \\ 0 & \text{if } C \geq d_\mathcal{G} \end{cases},$$

giving

$$\mathbb{E}\left[\Delta_{k,C}\right] \approx \mathbb{E}\left[\Delta_{k,0}\right] \times f(C, d_\mathcal{G}).$$

We can therefore also say:

$$ub_{ind} = 2L^K_{\Gamma_\eta} + \frac{\chi}{1-L_{\Gamma_\eta}} + \xi \sum_{k=1}^{K-1} L^{K-k-1}_{\Gamma_\eta}\mathbb{E}\left[\Delta_{k,0}\right],$$

$$ub_{net} \approx 2L_{\Gamma_\eta}^K + \frac{\chi}{1 - L_{\Gamma_\eta}} + \xi \sum_{k=1}^{K-1} L_{\Gamma_\eta}^{K-k-1} \mathbb{E}\left[\Delta_{k,0}\right] \times f(C, d_{\mathcal{G}}),$$

$$ub_{cent} = 2L_{\Gamma_\eta}^K + \frac{\chi}{1 - L_{\Gamma_\eta}}.$$

We therefore firstly have

$$ub_{ind} - ub_{net} \approx \xi \sum_{k=1}^{K-1} L_{\Gamma_\eta}^{K-k-1} \mathbb{E}\left[\Delta_{k,0}\right] - \xi \sum_{k=1}^{K-1} L_{\Gamma_\eta}^{K-k-1} \mathbb{E}\left[\Delta_{k,0}\right] \times f(C, d_{\mathcal{G}}),$$

which simplifies to

$$ub_{ind} - ub_{net} \approx \xi \sum_{k=1}^{K-1} L_{\Gamma_\eta}^{K-k-1} \mathbb{E}\left[\Delta_{k,0}\right] \times \left(1 - f(C, d_{\mathcal{G}})\right).$$

This gives us one of the results, where we focus on the functional dependence on C and $d_{\mathcal{G}}$ by using the tight bound big Theta (Θ):

$$ub_{net} \approx ub_{ind} - \Theta\left(1 - f(C, d_{\mathcal{G}})\right).$$

Secondly, we have

$$ub_{net} \approx ub_{cent} + \xi \sum_{k=1}^{K-1} L_{\Gamma_\eta}^{K-k-1} \mathbb{E}\left[\Delta_{k,0}\right] \times f(C, d_{\mathcal{G}}),$$

giving us the second result

$$ub_{net} \approx ub_{cent} + \Theta\left(f(C, d_{\mathcal{G}})\right).$$

□

Remark 6. *If it is always σ^1_{k+1} and π^1_{k+1} that is adopted by the whole population (i.e. $i = 1$), then this is exactly the same as the centralised case. If the σ^j_{k+1} and π^j_{k+1} that gets adopted has different j for each k then this is akin to a version of the centralised setting where the index of the central learning agent may differ for each k.*

Remark 7. *Thm. 3 assumes $\tau_k \to 0\ \forall k$. If we assume instead $\tau_k \in \mathbb{R}_{>0}$, then we have $ub_{net} \to ub_{ind}$ as $C \to 0$, and $ub_{net} \to ub_{cent}$ as $C \to \infty$. This is because the spread of policies is now probabilistic rather than deterministic, and depends on the interplay of τ_k with how large are the differences in the received values of σ^j_{k+1}. Therefore consensus (and hence reduction in divergence between policies) is reached only asymptotically. This applies to both static, connected networks and to repeatedly jointly connected ones, assuming the latter becomes jointly connected infinitely often.*

B.8 Policy-update stability guarantee

Theorem 4 (Policy-update stability guarantee)**.** *Let Alg. 1 run as per Thm. 1 or Thm. 2, and say that ε_k is the error term at iteration $k = \frac{\log 8\varepsilon_k^{-1}}{\log L_{\Gamma_\eta}^{-1}}$. For all agents i, the maximum possible distance between $\pi_k^{i,net}$ and $\pi_{k+1}^{i,net}$ is given by $\mathbb{E}\left[||\pi_k^{i,net} - \pi_{k+1}^{i,net}||_1\right] \leq \varepsilon_k + \varepsilon_{k+1} + \mathcal{O}\left(\frac{1}{\sqrt{N}}\right)$. This bound provides policy-update stability guarantees during the learning process; moreover the bound shrinks with each successive k since ε_k decreases with k. Equivalent analysis can also be conducted for both the centralised and independent cases.*

Proof. Thms. 1 and 2 bound the difference between each agent's current policy π_k^i and the unique equilibrium policy π^*, with the difference depending on the bias term ε_k that relates to the iteration k as indicated. Policies π_k^i and π_{k+1}^i fall within balls centred on π^* with radii of $\varepsilon_k + \mathcal{O}\left(\frac{1}{\sqrt{N}}\right)$ and $\varepsilon_{k+1} + \mathcal{O}\left(\frac{1}{\sqrt{N}}\right)$ respectively. This means that the maximum possible distance between π_k^i and π_{k+1}^i is the sum of these radii, i.e. $\mathbb{E}\left[||\pi_k^i - \pi_{k+1}^i||_1\right] \leq \varepsilon_k + \varepsilon_{k+1} + \mathcal{O}\left(\frac{1}{\sqrt{N}}\right)$, giving the result. □

B.9 REMARK ON THEORETICAL HYPERPARAMETERS WHEN USED IN PRACTICAL SETTINGS

Remark 8. *The theoretical analysis in Sec. 3.3 and Appx. B requires algorithmic hyperparameters (Thms. 1 and 2) that render convergence impractically slow in all of the centralised, independent and networked cases. (Indeed Yardim et al. (2023) do not provide empirical demonstrations of their algorithms for the centralised and independent cases.) In particular, the values of δ_{mix} and p_{inf} give rise to very large t_0, causing very small learning rates $\{\beta_m\}_{m\in\{0,\dots,M_{pg}-1\}}$, and necessitating very large values for M_{td} and M_{pg}.*

C EXPLANATION OF NETWORKED AGENTS OUTPERFORMING CENTRALISED AGENTS IN PRACTICAL SETTINGS

After the PMA step in Line 17 of Alg. 2, we have randomly updated policies $\{\pi^i_{k+1}\}_i$, $i = 1, ..., N$, where the randomness stems both from each agent's independent collection of samples and from the random sampling of each one's buffer when updating the Q-functions $\hat{Q}^i_{M_{pg}}$. The policies' associated finitely approximated returns are $\{\sigma^i_{k+1}\}_i$.

Let us assume for simplicity that in the networked case the population shares a single policy after the C communication rounds, i.e. that this has been enabled by the connectivity and diameter of the communication network $\mathcal{G}_t$ and the values of C and $\tau_k \in \mathbb{R}_{>0}$, as per Rems. 4 and 7. Call this network consensus policy π^{net}_{k+1}, and its associated finitely approximated return $\sigma^{\text{net}}_{k+1}$. Recall that the centralised case, where the updated policy of arbitrary agent $i = 1$ is automatically pushed to all the others, is equivalent to a networked case where policy consensus is reached on a random one of the policies $\{\pi^i_k\}_i$, $i = 1, ..., N$; call this policy arbitrarily given to the whole population π^{cent}_{k+1}, and its associated finitely approximated return $\sigma^{\text{cent}}_{k+1}$.

Since π^{cent}_{k+1} is chosen at random regardless of its quality, in expectation $\sigma^{\text{cent}}_{k+1}$ will be the mean value of $\{\sigma^i_{k+1}\}_i$ for each k, though there will be high variance. Conversely, the softmax adoption probability (Line 27 of Alg. 2) for the networked case is such that in expectation the π^{net}_{k+1} that gets adopted by the whole networked population will have higher than average $\sigma^{\text{net}}_{k+1}$ (indeed if $\tau_k \to 0$ it will have the highest $\sigma^{\text{net}}_{k+1}$ for each k). That is, the probability distribution is weighted by their relative estimated performance. As such, $\mathbb{E}[\sigma^{\text{net}}_{k+1}] > \mathbb{E}[\sigma^{\text{cent}}_{k+1}]$. There is also less variance in $\sigma^{\text{net}}_{k+1}$ than $\sigma^{\text{cent}}_{k+1}$, as the former is biased towards higher values.

If the networked case results in higher average σ_{k+1} being adopted than the centralised case, then the policies of which σ_{k+1} gives an approximated return are also biased towards being better performing, and with less variance in quality. Thus the networked agents can improve their return and converge faster than the centralised agents, by choosing updates in a more principled manner. This intuition applies even if we loosen the assumption that the networked population converges on a single consensus policy within the C communication rounds. (The same logic can of course also be applied to understand why networked agents outperform entirely independent ones.) It is significant that the communication scheme not only allows us to avoid the undesirable assumption of a centralised learner, but even to outperform it.

D ALGORITHM ACCELERATION BY USE OF EXPERIENCE-REPLAY BUFFER (FURTHER DETAILS)

The intuition behind the better learning efficiency resulting from our experience replay buffer in Alg. 2 is as follows. The value of a state-action pair p is dependent on the values of subsequent states reached, but the value of p is only updated when the TD update is conducted on p, rather than every time a subsequent pair is updated. By learning from each stored transition multiple times, we not only make repeated use of the reward and transition information in each costly experience, but also repeatedly update each state-action pair in light of its likewise updated subsequent states.

We leave β_m fixed across all iterations, as we found empirically that this yields sufficient learning. We have not experimented with decreasing β as l increases, though this may benefit learning.

The transitions in the buffer are discarded after the replay cycles and a new buffer is initialised for the next iteration k, as in Line 4. As such the space complexity of the buffer only grows linearly with the number of M_{pg} iterations within each outer loop k, rather than with the number of K loops.

Algorithm 2 Networked learning with experience replay

Require: loop parameters $K, M_{pg}, M_{td}, C, L, \mathrm{E}$, learning parameters η, β, λ, γ, $\{\tau_k\}_{k\in\{0,\dots,K-1\}}$
Require: initial states $\{s_0^i\}_i, i = 1, \dots, N$

1: Set $\pi_0^i = \pi_{\max}, \forall i$ and $t \leftarrow 0$
2: **for** $k = 0, \dots, K-1$ **do**
3:     $\forall s, a, i : \hat{Q}_0^i(s, a) = Q_{\max}$
4:     $\forall i$: Empty i's buffer
5:     **for** $m = 0, \dots, M_{pg} - 1$ **do**
6:         **for** M_{td} iterations **do**
7:             Take step $\forall i : a_t^i \sim \pi_k^i(\cdot|s_t^i), r_t^i = R(s_t^i, a_t^i, \hat{\mu}_t), s_{t+1}^i \sim P(\cdot|s_t^i, a_t^i, \hat{\mu}_t); t \leftarrow t + 1$
8:         **end for**
9:         $\forall i$: Add ζ_{t-2}^i to i's buffer
10:     **end for**
11:     **for** $l = 0, \dots, L - 1$ **do**
12:         $\forall i$: Shuffle buffer
13:         **for** transition ζ_b^i in i's buffer **do** ($\forall i$)
14:             Compute TD update ($\forall i$): $\hat{Q}_{m+1}^i = \tilde{F}_\beta^{\pi_k^i}(\hat{Q}_m^i, \zeta_{t-2}^i)$ (see Def. 7)
15:         **end for**
16:     **end for**
17:     PMA step $\forall i : \pi_{k+1}^i = \Gamma_\eta^{md}(\hat{Q}_{M_{pg}}^i, \pi_k^i)$ (see Def. 8)
18:     $\forall i : \sigma_{k+1}^i = 0$
19:     **for** $e = 0, \dots, E - 1$ evaluation steps **do**
20:         Take step $\forall i : a_t^i \sim \pi_{k+1}^i(\cdot|s_t^i), r_t^i = R(s_t^i, a_t^i, \hat{\mu}_t), s_{t+1}^i \sim P(\cdot|s_t^i, a_t^i, \hat{\mu}_t)$
21:         $\forall i : \sigma_{k+1}^i = \sigma_{k+1}^i + \gamma^e(r_t^i + h(\pi_{k+1}^i(s_t^i)))$
22:         $t \leftarrow t + 1$
23:     **end for**
24:     **for** C rounds **do**
25:         $\forall i$: Broadcast $\sigma_{k+1}^i, \pi_{k+1}^i$
26:         $\forall i : J_t^i = i \cup \{j \in \mathcal{N} : (i, j) \in \mathcal{E}_t\}$
27:         $\forall i$: Select $\text{adopted}^i \sim \Pr\big(\text{adopted}^i = j\big) = \frac{\exp(\sigma_{k+1}^j/\tau_k)}{\sum_{x\in J_t^i}\exp(\sigma_{k+1}^x/\tau_k)}\ \forall j \in J_t^i$
28:         $\forall i : \sigma_{k+1}^i \leftarrow \sigma_{k+1}^{\text{adopted}^i}, \pi_{k+1}^i \leftarrow \pi_{k+1}^{\text{adopted}^i}$
29:         Take step $\forall i : a_t^i \sim \pi_{k+1}^i(\cdot|s_t^i), r_t^i = R(s_t^i, a_t^i, \hat{\mu}_t), s_{t+1}^i \sim P(\cdot|s_t^i, a_t^i, \hat{\mu}_t); t \leftarrow t + 1$
30:     **end for**
31: **end for**
32: Return policies $\{\pi_K^i\}_i, i = 1, \dots, N$

E Extended discussion on robustness of communication networks in MFGs and related experimental settings

We consider two scenarios to which we desire real-world many-agent systems (e.g. robotic swarms or autonomous vehicle traffic) to be robust; these scenarios form the basis of our experiments on robustness (see Sec. 4 and Figs. 2, 3, 8 and 9). The networked setup affords population **fault-tolerance** and **online scalability**, which are motivating qualities of many-agent systems.

Firstly, we consider a scenario in which the learning/updating procedure of agents fails with a certain probability within each iteration, in which cases $\pi_{k+1}^i = \pi_k^i$ (see Figs. 2 and 8 for our experimental results and discussion of this scenario). In real-life decentralised settings, this might be particularly liable to occur since the updating process might only be synchronised between agents by internal clock ticks, such that some agents may not complete their update in the allotted time but will nevertheless be required to take the next step in the environment. Such failures slow the

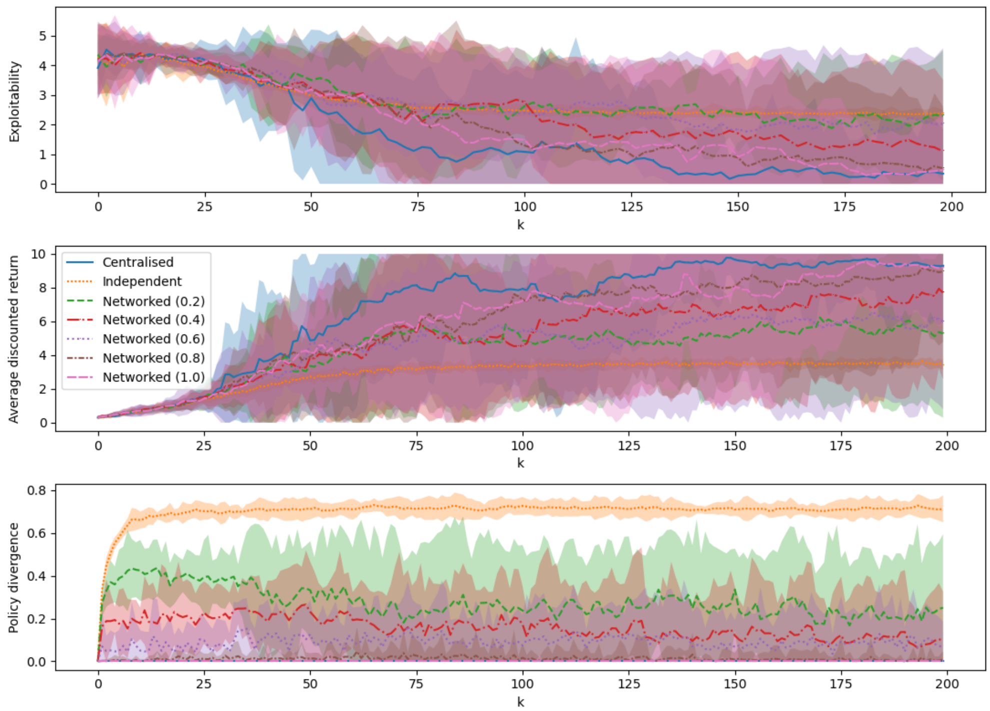

Figure 4: Larger version of Fig. 1. 'Target agreement' game. Even with only a single communication round, our networked case outperforms the independent case wrt. exploitability, and markedly outperforms wrt. return. The fact that the lowest broadcast radius (0.2) ends with similar exploitability to the independent case yet much higher return shows our networked algorithm can help agents find 'preferable' equilibria.

improvement of the population in the independent case, and in the centralised setting it means no improvement occurs at all in any iteration in which failure occurs, as there is a single point of failure. Networked communication instead provides redundancy in case of failures, with the updated policies of any agents that have managed to learn spreading through the population to those that have not. This feature thus ensures that improvement can continue for potentially the whole population even if a high number of agents do not manage to learn at a given iteration.

Secondly, we may want to arbitrarily increase the size of a population of agents that are already learning or operating in the environment (we can imagine extra fleets of autonomous cars or drones being deployed) - see Appx. G for comparison with other works considering this type of robustness (Eck et al., 2023; Gao et al., 2024; Dawood et al., 2023; Wu et al., 2024b). A purely independent setting would require all the new agents to learn a policy individually given the existing distribution, and the process of their following and improving policies from scratch may itself disturb the NE that has already been achieved by the original population. With a communication network, however, the policies that have been learnt so far can quickly be shared with the new agents in a decentralised way, hopefully before their unoptimised policies can destabilise the current NE. This would provide, for example, a way to bootstrap a large population from a smaller pre-trained group, if training were considered expensive in a given setting. See Figs. 3 and 9 for our experimental results and discussion of this scenario.

F Experiments

Experiments were conducted on a MacBook Pro, Apple M1 Max chip, 32 GB, 10 cores. We use `scipy.optimize.minimize` (employing Sequential Least Squares Programming) to conduct the optimisation step in Def. 8, and the JAX framework to accelerate and vectorise some elements of our code.

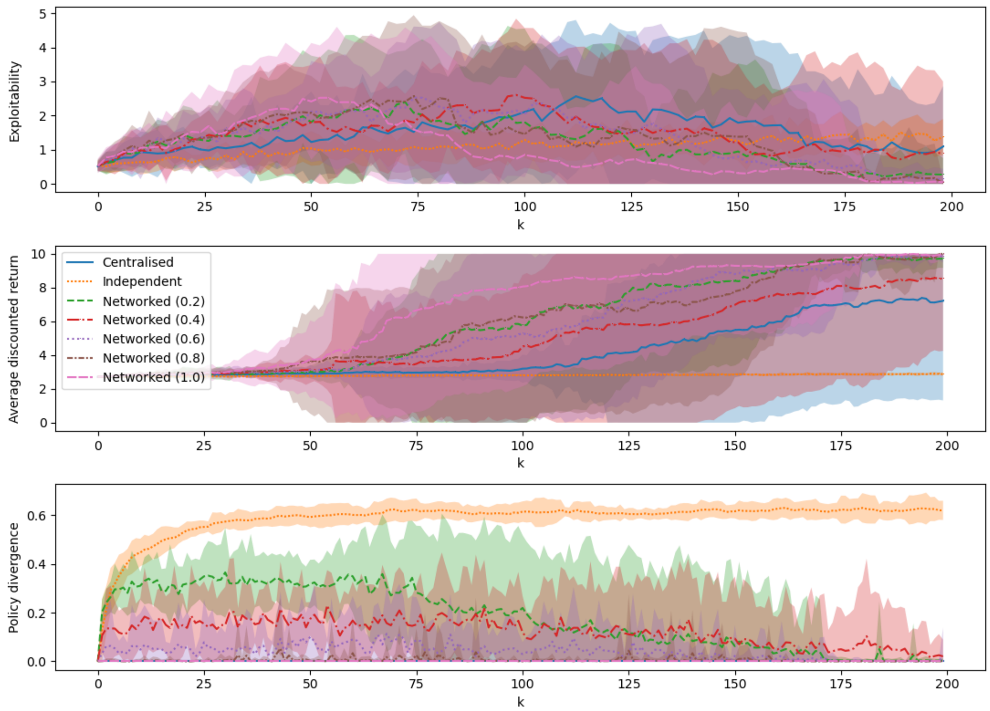

Figure 5: Larger version of Fig. 2. ‘Cluster’ game, testing robustness to 50% probability of policy update failure. The communication network allows agents that have successfully updated their policies to spread this information to those that have not, providing redundancy. Independent learners cannot do this and hardly appear to learn at all (no increase in return); likewise the centralised architecture is susceptible to its single point of failure. Thus our networked architecture significantly outperforms both the centralised and independent cases.

For reproducibility, the code to run our experiments is provided with our Supplementary Material, and will be made publicly available upon publication.

F.1 Games

We conduct numerical tests with two games (defined by the agents’ objectives), chosen for being particularly amenable to intuitive and visualisable understanding of whether the agents are learning behaviours that are appropriate and explainable for the respective objective functions. In all cases, rewards are normalised in [0,1] after they are computed.

Cluster. This is the inverse of the ‘exploration’ game in (Lauriere et al., 2022), where in our case agents are encouraged to gather together by the reward function $R(s_t^i, a_t^i, \hat{\mu}_t) = \log(\hat{\mu}_t(s_t^i))$. That is, agent i receives a reward that is logarithmically proportional to the fraction of the population that is co-located with it at time t. We give the population no indication where they should cluster, agreeing this themselves over time.

Agree on a single target. Unlike in the above ‘cluster’ game, the agents are given options of locations at which to gather, and they must reach consensus among themselves. If the agents are co-located with one of a number of specified targets $\phi \in \Phi$ (in our experiments we place one target in each of the four corners of the grid), and other agents are also at that target, they get a reward proportional to the fraction of the population found there; otherwise they receive a penalty of -1. In other words, the agents must coordinate on which of a number of mutually beneficial points will be their single gathering place. The reward function is given by $R(s_t^i, a_t^i, \hat{\mu}_t) = r_{targ}(r_{collab}(\hat{\mu}_t(s_t^i)))$,

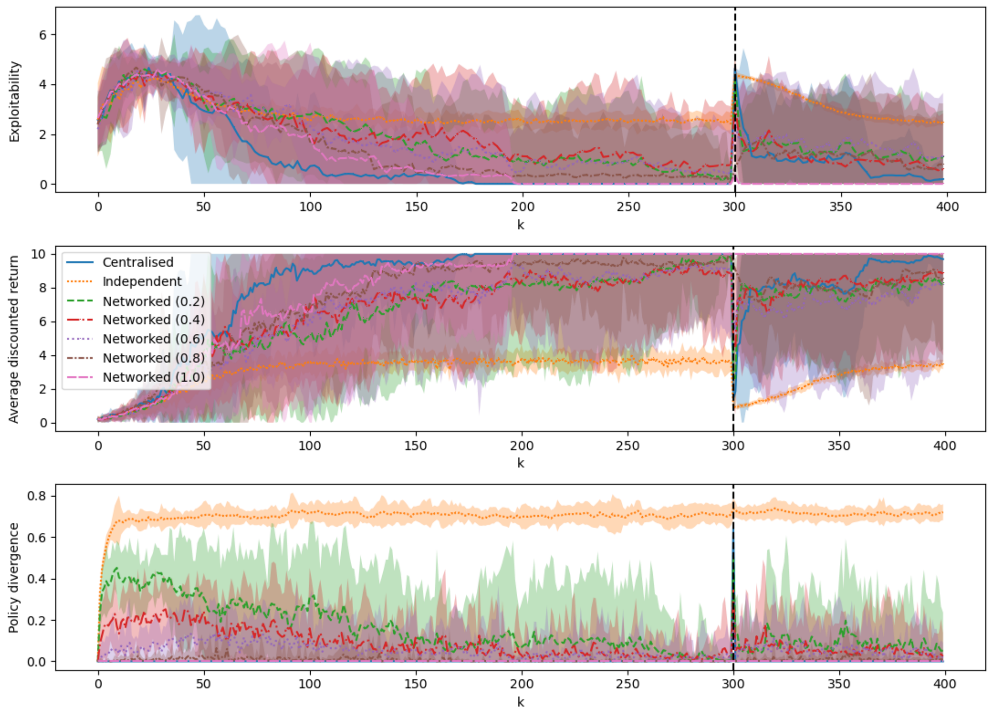

Figure 6: Larger version of Fig. 3. 'Target agreement' game, testing robustness to a five-times increase in population. The networked architectures are quickly able to spread the learnt policies to the newly arrived agents such that learning progress is minimally disturbed, whereas convergence is significantly impacted in the independent case. The largest broadcast radius (1.0), in particular, appears to suffer no disturbance at all, being much more robust than the centralised case, which takes a significant amount of time to return to equilibrium.

where

$$r_{targ}(x) = \begin{cases} x & \text{if } \exists \phi \in \Phi \text{ s.t. } \mathrm{dist}(s_t^i, \phi) = 0 \\ -1 & \text{otherwise,} \end{cases}$$

$$r_{collab}(x) = \begin{cases} x & \text{if } \hat{\mu}_t(s_t^i) > 1/N \\ -1 & \text{otherwise.} \end{cases}$$

F.2 Experimental metrics

To give as informative results as possible about both performance and proximity to the NE, we provide several metrics for each experiment. All metrics are plotted with 2-sigma error bars (2 $\times$ standard deviation), computed over the 10 trials (each with a random seed) of the system evolution in each setting. This is computed based on a call to `numpy.std` for each metric over each run.

F.2.1 Exploitability

Works on MFGs frequently use the *exploitability* metric to evaluate how close a given policy π is to a NE policy π^* (Lauriere et al., 2022; Perrin et al., 2020; Laurière et al., 2022; Algumaei et al., 2023; Pérolat et al., 2022). The metric quantifies how much an agent can benefit by deviating from the policy pursued by the rest of the population, by measuring the difference between the return given by a policy that maximises the expected discounted regularised (via h) reward V_h for a given population distribution, and the return given by the policy that gives rise to this distribution. If π has a large exploitability then an agent can significantly improve its return by deviating from π, meaning that π is far from π^*, whereas an exploitability of 0 implies that $\pi = \pi^*$. Denote by μ^π the distribution generated when π is the policy followed by all of the population aside from the deviating agent; then

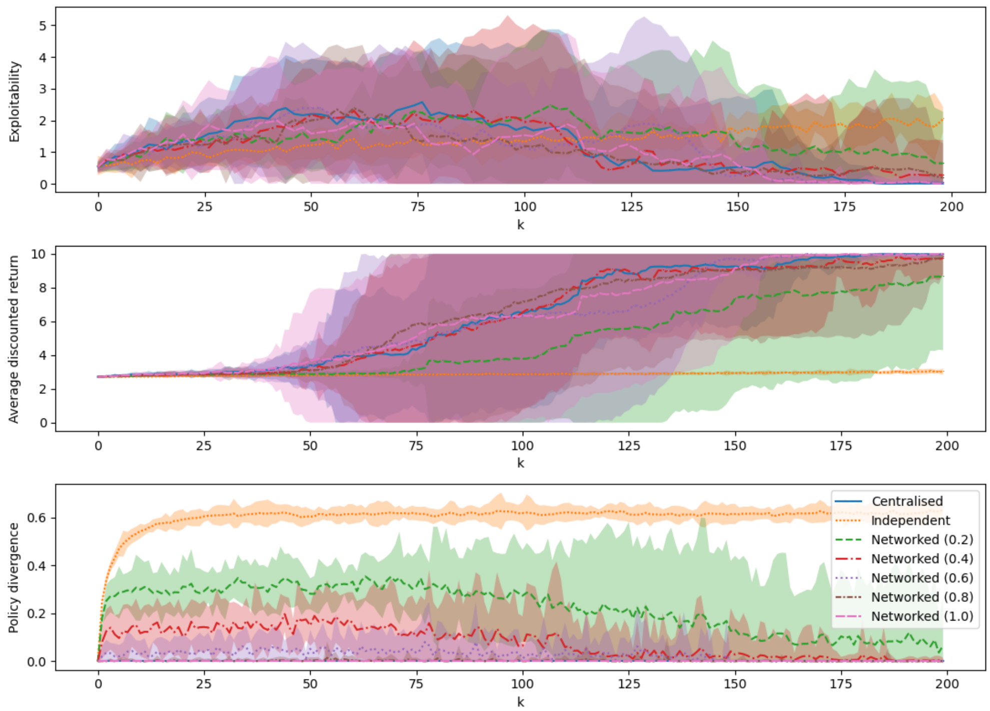

Figure 7: 'Cluster' game. Even with only a single communication round, our networked architecture significantly outperforms the independent case, which hardly appears to be learning at all. All broadcast radii except the smallest (0.2) appear to match and at times even outperform the centralised case.

the exploitability of policy π is defined as:

$$\mathcal{E}(\pi) = \max_{\pi'} V_h(\pi', \mu^\pi) - V_h(\pi, \mu^\pi).$$

Since we do not have access to the exact best response policy $\arg\max_{\pi'} V_h(\pi', \mu^\pi)$, we instead approximate the exploitability metric, similarly to (Perrin et al., 2021), as follows. We freeze the policy of all agents apart from a deviating agent, for which we store its current policy and then conduct 40 k loops of policy improvement (we found that 40 iterations was enough to converge to a policy that maximised V_h for the given population distribution). To approximate the expectations, we take the best return of the deviating agent across the 40 k loops, as well as the mean of all the other agents' returns across these same loops. We then revert the agent back to its stored policy, before learning continues for all agents. As such, the quality of our approximation is limited by the number of policy improvement rounds, which must be restricted for the sake of running speed of the experiments. Due to the expensive computations required for this metric, we evaluate it on alternate k iterations.

Since prior works conducting empirical testing have generally focused on the centralised setting, evaluations have not had to consider the exploitability metric when not all agents are following a single policy π_k, as may occur in the independent or networked settings. The method described above for approximating exploitability involves calculating the mean return of all non-deviating agents' policies. While this is π_k in the centralised case, if the non-deviating agents do not share a single policy, then this method is in fact approximating the exploitability of their joint policy $\boldsymbol{\pi}_k^{-d}$, where d is the deviating agent.

F.2.2 Average discounted return

We record the average discounted return of the agents' policies π_k^i during the M_{pg} steps - this allows us to observe that settings that converge to similar exploitability values may not have similar average

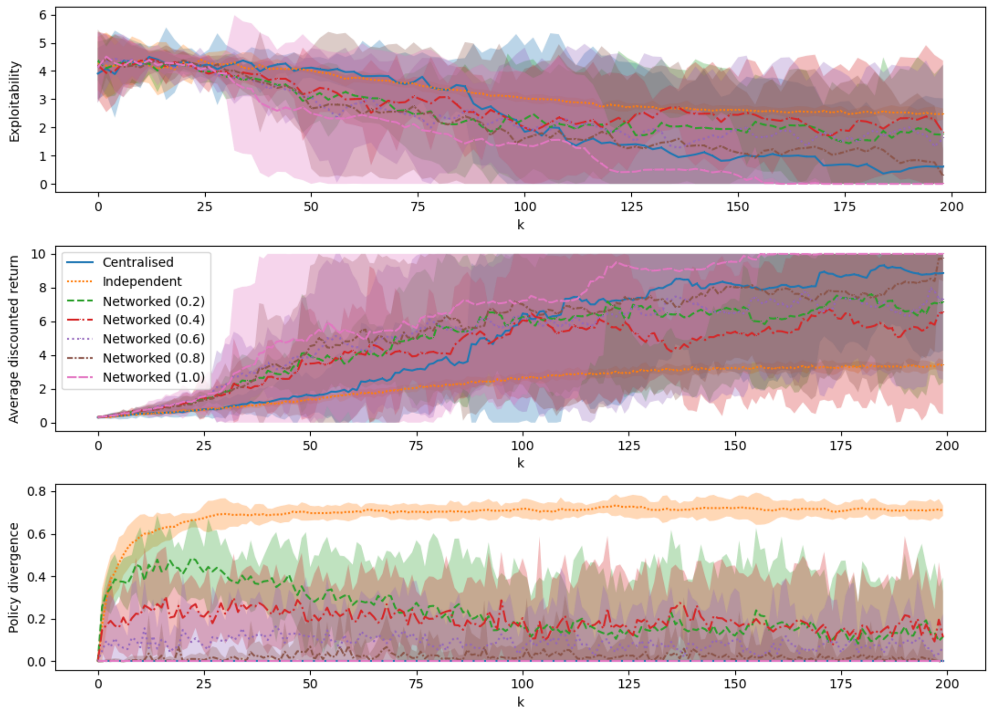

Figure 8: 'Target agreement' game, testing robustness to 50% probability of policy update failure. All the networked cases significantly outperform the independent case and also learn much faster than the centralised case for long periods. The communication network allows agents that have successfully updated their policies to spread this information to those that have not, providing redundancy. Independent learners cannot do this so have even slower convergence than normal; likewise the centralised architecture is susceptible to its single point of failure, hence learning can be slower than in the networked case.

agent returns, suggesting that some algorithms are better than others at finding not just NE, but preferable NE. See for example Fig. 13, where the networked agents converge to similar exploitability as the independent agents, but receive higher average reward.

F.2.3 Policy divergence

We record the population's average policy divergence $\frac{1}{N}\Delta_k := \frac{1}{N}\sum_{i=1}^{N} ||\pi_k^i - \pi_k^1||_1$ for the arbitrary policy $\bar{\pi} = \pi^1$. This allows us to demonstrate that populations approaching the NE (i.e. with joint exploitability approaching zero) do not necessarily actually share a single policy π^* as suggested by the theoretical sample guarantees in Sec. 3.3. Our experimental plots show that this is particularly often the case in the independent setting. The greater divergence in the independent case also indicates why convergence is slower here (see Rem. 5).

F.3 Hyperparameters

See Table 1 for our hyperparameter choices. In general, we seek to show that our networked algorithm is robust to 'poor' choices of hyperparameters e.g. low numbers of iterations, as may be required when aiming for practical convergence times in complex real-world problems. By contrast, the convergence speed of the independent-learning algorithm (and sometimes also the centralised algorithm) suffers much more significantly without idealised hyperparameter choices. As such, our experimental demonstrations in the plots generally involve hyperparameter choices at the low end of the values we tested during our research.

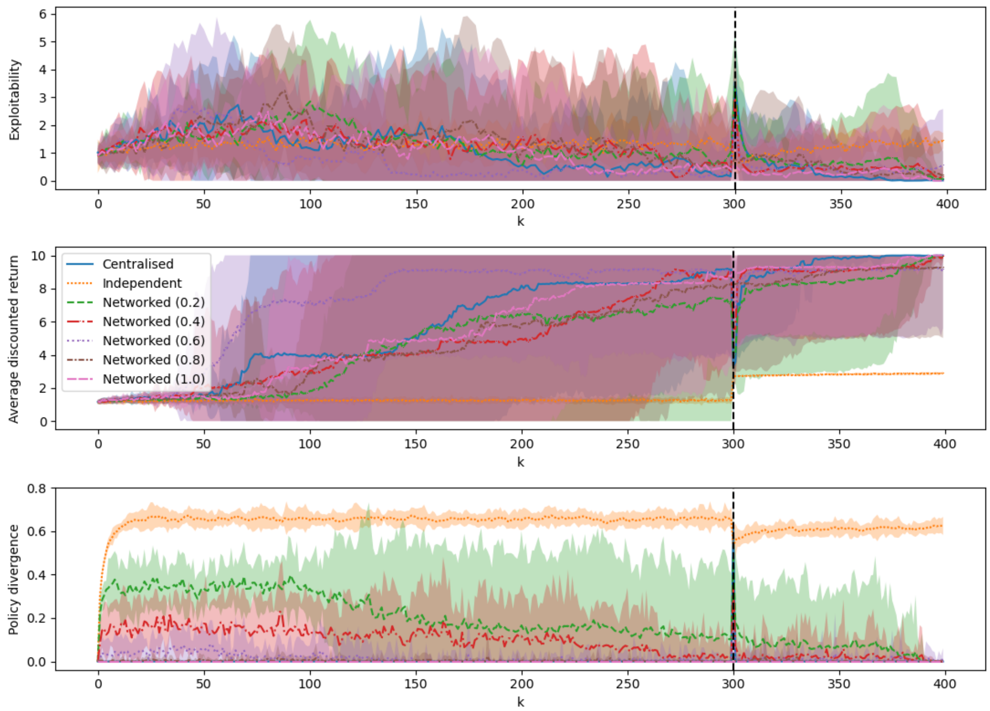

Figure 9: 'Cluster' game, testing robustness to a five-times increase in population. While the independent algorithm appears to enjoy similar exploitability to the other cases (see Rem. 9), we can see from its average return that it is not in fact learning at all; while the return rises after the increase in population size this is only because there are now more agents with which to be co-located, rather than because learning has progressed. Since here, unlike in the 'target agreement' game in Fig. 3, independent agents have hardly improved their return in the first place, we do not see the adverse effect that the addition of agents to the population has on the progress of learning. All networked cases perform similarly to or significantly outperform the centralised case, and all significantly outperform the independent case in terms of return. The communication network allows the learnt policies to quickly spread to the newly arrived agents, such that the progression of learning is minimally disturbed, without needing to rely on the assumption of a centralised learner. The fact that, in all cases, the return prior to the population increase is lower than in Fig. 7, is reflective of the fact that the error in the solution reduces as N tends to infinity.

We can group our hyperparameters into those controlling the size of the experiment, those controlling the number of iterations of each loop in the algorithm and those affecting the learning/policy updates or policy adoption $(\beta, \eta, \lambda, \tau, \gamma)$.

F.4 Additional experiments and discussion

In this section we showcase results with our standard hyperparameter choices continuing from those shown in Sec. 4.1 (Figs. 7, 8 and 9), and we also vary several hyperparameters to show their effects on convergence (Figs. 10 - 13). We also give an ablation study of our replay buffer in Figs. 14 and 15.

Remark 9. *Note that the reward structure of our coordination games is such that exploitability sometimes increases from its initial value before it decreases down to 0. This is because agents are rewarded proportionally to how many other agents are co-located with them: when agents are evenly dispersed at the beginning of the run, it is difficult for even a deviating, best-responding agent to significantly increase its reward. However, once some agents start to aggregate, a best-responding agent can take advantage of this to substantially increase its reward (giving higher exploitability), before all the other agents catch up and aggregate at a single point, reducing the exploitability down to 0. Due to this arc, in some of our plots the independent case may have lower exploitability at certain points than the other architectures, but this is not necessarily a sign of good performance.*

Table 1: Hyperparameters

Hyper-param.	Value	Comment
Gridsize	8x8 / 16x16	Most experiments are run on the smaller grid, while Figs. 10 and 11 showcase learning in a larger state space.
Trials	10	We run 10 trials with different random seeds for each experiment. We plot the mean and 2-sigma error bars for each metric across the trials.
Pop.	250	We tested N in {25,50,100,200,250,500}, with the networked architecture generally performing equally well with all population sizes ≥ 50. We chose 250 for our demonstrations, to show that our algorithm can handle large populations, indeed often larger than those demonstrated in other mean-field works, especially for grid-world environments (Yongacoglu et al., 2022a; Cui & Koeppl, 2021; Cui et al., 2023; Guo et al., 2023; Subramanian & Mahajan, 2019; Yang et al., 2018b; Ganapathi Subramanian et al., 2021; 2020; Subramanian et al., 2022), while also being feasible to simulate wrt. time and computation constraints. In experiments testing robustness to population increase, the population instead begins at 50 agents and has 200 added at the marked point.
K	200 / 400	K is chosen to be large enough to see exploitability reducing, and converging where possible.
M_{pg}	500 / 1000	We wish to illustrate the benefits of our networked architecture and replay buffer in reducing the number of loops required for convergence, i.e. we wish to select a low value that still permits learning. We tested M_{pg} in {300,500, 600,800,1000,1200,1300,1400,1500,1800,2000,2500,3000}, and chose 500 for demonstrations on the 8x8 grids, and 1000 for the 16x16 grids. It may be possible to optimise these values further in combination with other hyperparameters.
M_{td}	1	We tested M_{td} in {1,2,10,100}, and found that we could still achieve convergence with $M_{td} = 1$. This is much lower than the requirements of the theoretical algorithms, essentially allowing us to remove the innermost nested learning loop.
C	1	We tested C in {1,20,50,300}. We choose 1 to show the convergence benefits brought by even a single communication round, even in networks that may have limited connectivity; higher C has even better performance.
L	100	As with M_{pg}, we wish to select a low value that still permits learning. We tested L in {50,100,200,300,400,500}. In combination with our other hyperparameters, we found $L \leq 50$ led to less good results, but it may be possible to optimise this hyperparameter further.
E	100	We tested E in {100,300,1000}, and choose the lowest value to show the benefit to convergence even from very few evaluation steps. It may be possible to reduce this value further and still achieve similar results.
γ	0.9	Standard choice across RL literature.
β	0.1	We tested β in {0.01,0.1} and found 0.1 to be small enough for sufficient learning at an acceptable speed. Further optimising this hyperparameter (including by having it decay with increasing $l \in 0, \ldots, L-1$, rather than leaving it fixed) may lead to better results.
η	0.01	We tested η in {0.001,0.01,0.1,1,10} and found that 0.01 gave stable learning that progressed sufficiently quickly.
λ	0	We tested λ in {0,0.0001,0.001,0.01,0.1,1}. Since we can reduce λ to 0 with no detriment to empirical convergence, we do so in order not to bias the NE.
τ_k	cf. comment	For fixed τ_k $\forall k$, we tested {1,10,100,1000}. In our experiments for fixed τ_k the value is 100 (see Figs. 12 and 13); this yields learning, but does not perform as well as if we anneal τ_k as follows. We begin with $\tau_0 = 10000/(10 ** \lceil (K-1)/10 \rceil)$, and multiply τ_k by 10 whenever $k \bmod 10 = 1$ i.e. every 10 iterations. Further optimising the annealing process may lead to better results.

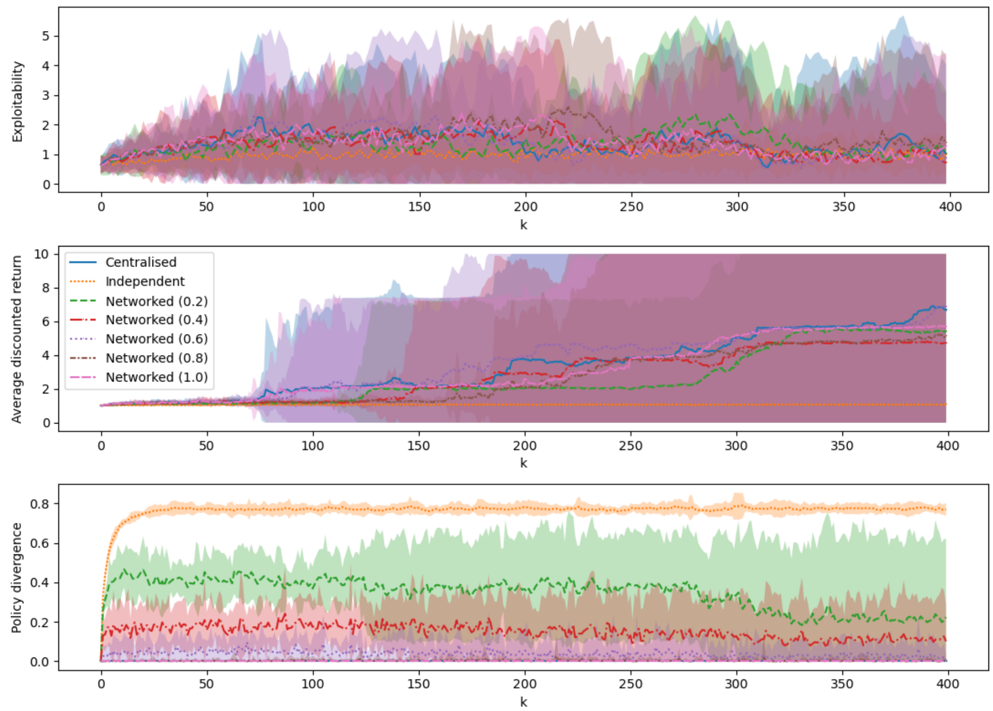

Figure 10: ‘Cluster’ game on the larger 16x16 grid. While the independent-learning case has similar exploitability to the other settings, we can see that it is not actually learning to increase its return at all, making this an undesirable equilibrium. (I.e. agents are moving about randomly so there is little a deviating agent can do to increase its reward, hence exploitability is low even though the agents are not in fact clustered - see Rem. 9.) All the networked settings perform similarly to the centralised case and significantly outperform the return of the independent agents.

In fact, we can see in some such cases that the independent case is not learning at all, with the independent agents' average return not increasing and the exploitability staying level rather than ultimately decreasing (see, for example, Figs. 2, 7, 9 and 10).

In our additional experiments, where the results are discussed fully in each figure's caption, the factors we vary to show the effects on convergence are as follows:

- **Grid size**. Figs. 10 and 11 show the result of learning on a grid of size 16x16 instead of 8x8 as in all other experiments. There is at times greater differentiation in this setting than in the 8x8 grid between the performances of the different broadcast radii of the networked architecture (as is to be expected in a less densely populated environment). The networked architecture continues to significantly outperform the independent case for most broadcast radii, and sometimes even the centralised case.
- **Ablation study of softmax temperature annealing scheme**. Figs. 12 and 13 illustrate the effect of fixed $\{\tau_k\}_{k\in\{0,\dots,K-1\}} = 100$, where the networked architecture does not perform as well as if we use the stepped annealing scheme employed in all the other experiments and detailed in Table 1. The intuition behind the better performance achieved with the annealing scheme is as follows. If we begin with $\tau_k \to 0$ (such that the softmax approaches being a max function), we heavily favour the adoption of the highest rewarded policies to speed up progress in the early stages of learning. Subsequently we increase τ_k in steps, promoting greater randomness in adoption, so that as the agents come closer to equilibrium, poorer policy updates that nevertheless receive a high return (due to randomness) do not introduce too much instability to learning and prevent convergence.
- **Ablation study of experience replay buffer**. Figs. 14 and 15 illustrate how crucial is our incorporation of the experience replay buffer. Without it, as in the original theoretical

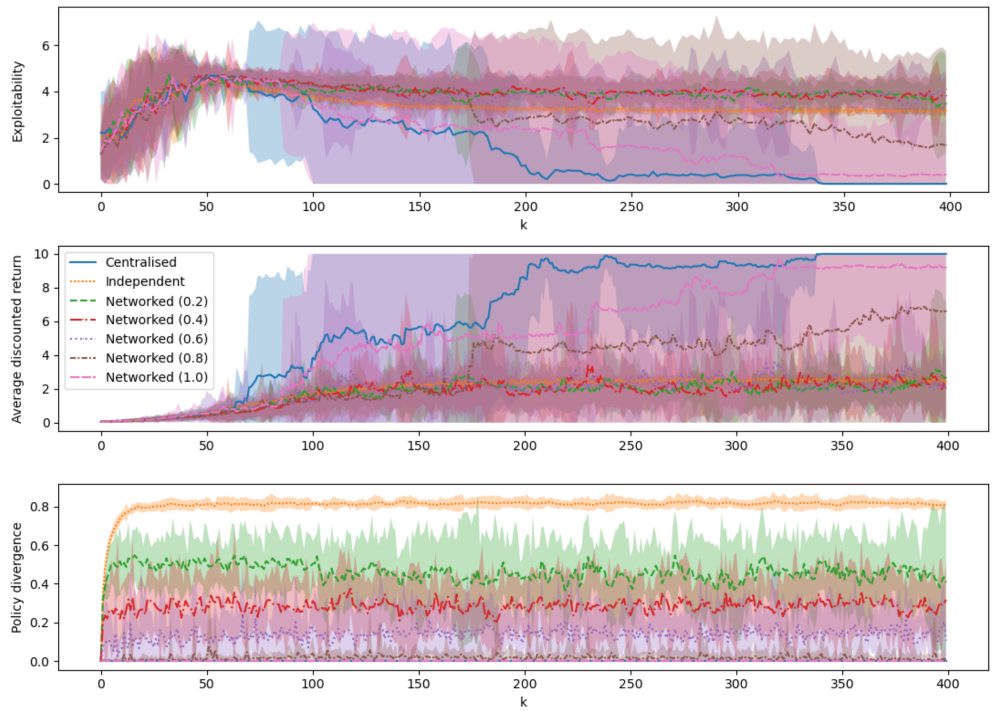

Figure 11: ‘Target agreement’ game on the larger 16x16 grid. There is greater differentiation in this setting than in the 8x8 grid (Fig. 1) between the different broadcast radii in the networked cases, as is to be expected in a less densely populated environment. The two largest broadcast radii (1.0 and 0.8), which have the most connected networks, significantly outperform the independent case in terms of both exploitability and return. However, the other broadcast radii perform similarly to the independent case.

version of the algorithms, there is no noticeable improvement in any of the agents’ returns, i.e. no noticeable learning, even after $K = 400$ iterations. When removing the buffer for these experiments we run the core learning section of the algorithm as in Lines 3-10 of Alg. 1, keeping the hyperparameters the same as in our main experiments, i.e. $M_{pg} = 500$, $M_{td} = 1$, etc. (see Tab. 1).

G EXTENDED RELATED WORK

Multi-agent reinforcement learning (MARL) (Zhang et al., 2021b; Busoniu et al., 2008) is a generalisation of reinforcement learning (Sutton & Barto, 2018) that has recently seen empirical success in a variety of domains, underpinned by breakthroughs in deep learning, including robotics (Leottau et al., 2018; Lv et al., 2023; Orr & Dutta, 2023; Guan et al., 2024; Ali et al., 2023), smart autonomy and infrastructures (Shalev-Shwartz et al., 2016; Mannion et al., 2016), complex games (Samvelyan et al., 2019; Vinyals et al., 2019a; Berner et al., 2019), economics (Rashedi et al., 2016; Shavandi & Khedmati, 2022), social science and cooperative AI (Leibo et al., 2017; Cao et al., 2018; Jaques et al., 2019; McKee et al., 2020). However, it has been computationally difficult to scale MARL algorithms beyond configurations with agents numbering in the low tens, as the joint state and action spaces grow exponentially with the number of agents (Xie et al., 2021; Lauriere et al., 2022; Perrin et al., 2020; Shavandi & Khedmati, 2022; Daskalakis et al., 2006; Vinyals et al., 2019b; Mcaleer et al., 2020). Nevertheless, the value of reasoning about interactions among very large populations of agents has been recognised, and an informal distinction is sometimes drawn between multi- and *many*-agent systems (Wang et al., 2020a; Zheng et al., 2018; Cui et al., 2022). The latter situation can be more useful (as in cases where better solutions arise from the presence of more agents (Orr & Dutta, 2023; Ornia et al., 2022; Shiri et al., 2019; Eck et al., 2023)), more parallelisable (Andréen et al., 2016),

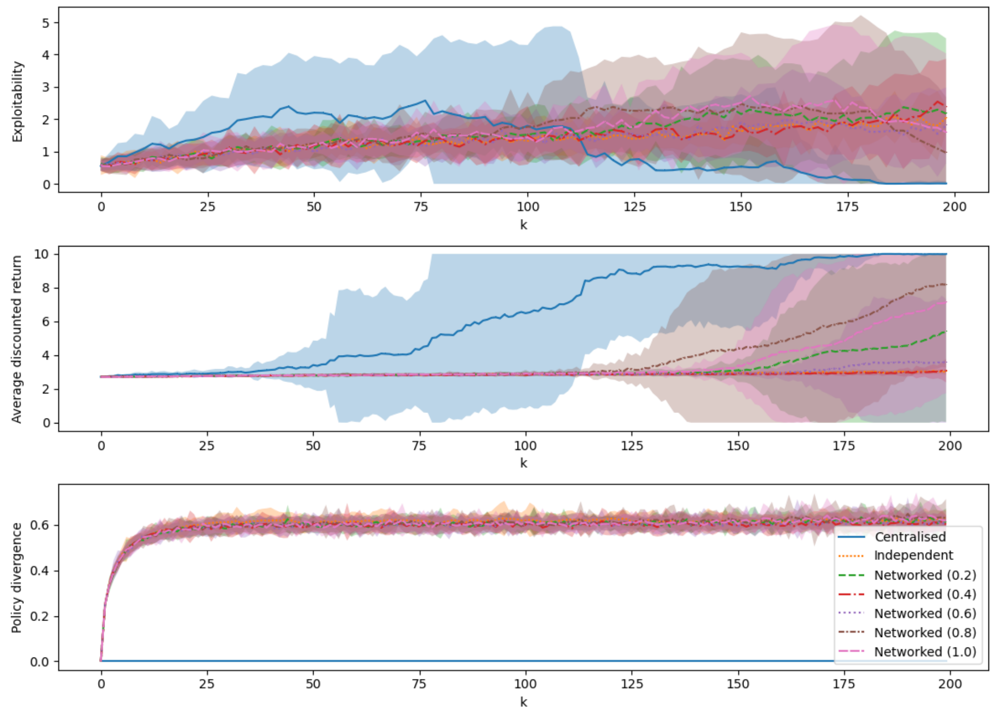

Figure 12: 'Cluster' game with τ_k fixed as 100 for all k; compare this to Fig. 7 where τ_k is annealed. Without the annealing scheme, the networked architecture appears to perform similarly to the independent case in terms of exploitability, but several broadcast radii outperform the independent case in terms of return, demonstrating that our networked algorithm can still help agents find 'preferable' equilibria. However, whereas with annealing the networked architecture converges similarly to the centralised case, here it performs less well.

more fault tolerant (Chang et al., 2023), or otherwise more reflective of certain real-world systems involving large numbers of decision makers (Eck et al., 2023; Rashedi et al., 2016; Shavandi & Khedmati, 2022; Meigs et al., 2020). Indeed, MFGs have been applied to a wide variety of real world problems, including financial markets (Trimborn et al., 2018); cryptocurrency mining (Li et al., 2024); autonomous vehicles (Huang et al., 2020); traffic signal control (Hu et al., 2023); resource management in fisheries (Yoshioka et al., 2024); crowdsensing (Yang et al., 2023); electric vehicle charging (Dey & Xu, 2023); communication networks (Wang et al., 2024; 2020b); swarms (Le Ménec, 0); data collection by UAVs (Emami et al., 2024); edge computing (Aggarwal et al., 2024; Shen et al., 2024; Miao et al., 2024); cloud resource management (Mao et al., 2022); smart grids, and other large-scale cyber-physical systems (Bauso & Tembine, 2016; Mishra et al., 2023; Benamor et al., 2022).

Our networked communication framework possesses all of the following desirable qualities for mean-field algorithms when applied to large, complex real-world many-agent systems: learning from the empirical distribution of N agents without generation or manipulation of this distribution by the algorithm itself or by an external oracle; learning from a single continued system run that is not arbitrarily reset as in episodic learning (also referred to in other works as a single sample path/trajectory (Zaman et al., 2023; Yardim et al., 2023)); model-free learning; decentralisation; fast practical convergence; and robustness to unexpected failures of decentralised learners or changes in the size of the population.

Conversely, as we emphasise in Sec. 1, the MFG framework was originally mainly theoretical (Lasry & Lions, 2007; Huang et al., 2006). The MFG-NE is traditionally found by solving a coupled system of dynamical equations: a forward evolution equation for the mean-field distribution, and a backwards equation for the representative agent's optimal response to the mean-field distribution, as in Def. 5 (Yoshioka et al., 2024); crucially, these methods relied on the assumption of an infinite population

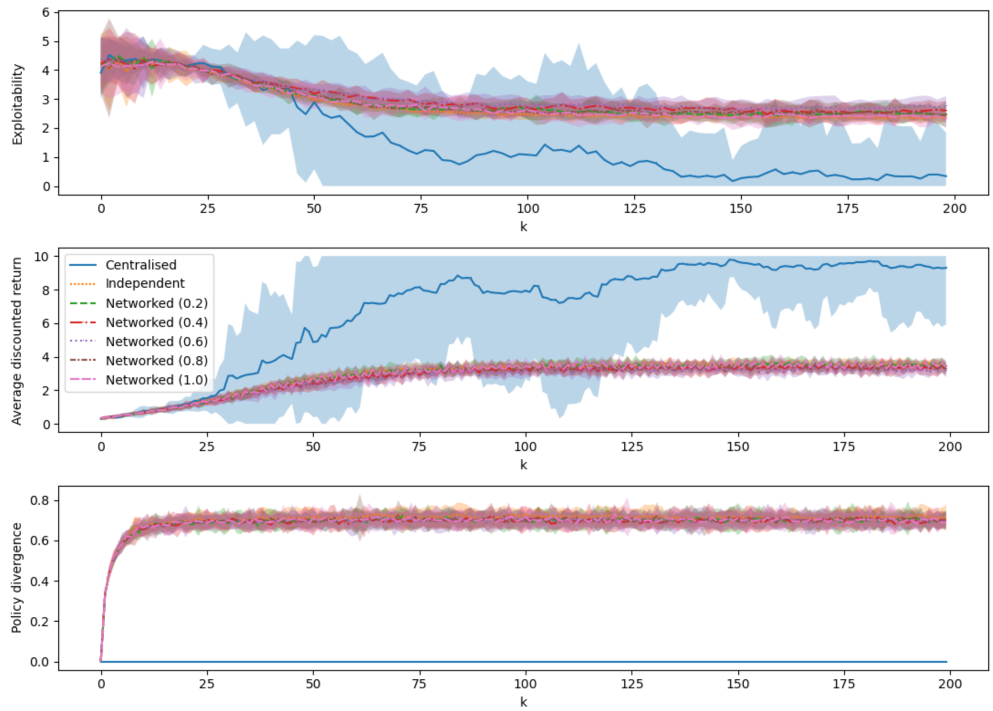

Figure 13: ‘Target agreement’ game with τ_k fixed as 100 for all k. Without our annealing scheme for the softmax temperature, the networked architecture does not outperform the independent case. Compare this to Fig. 1 which shows the benefit of annealing τ_k.

(Laurière et al., 2022). Early work solved the coupled equations using numerical methods that did not scale well for more complex state and action spaces (Achdou & Capuzzo-Dolcetta, 2010; Carlini & Silva, 2014; Briceño-Arias et al., 2018; Achdou et al., 2020); or, even if they could handle higher-dimensional problems, the methods were based on known models of the environment's dynamics (i.e. they were model-based) (Anahtarci et al., 2023; Guo et al., 2019a; Carmona & Laurière, 2021; Cao et al., 2020; Germain et al., 2022; Fouque & Zhang, 2020; Huang et al., 2024b;a), and/or computed a best-response to the mean-field distribution (Huang et al., 2006; Lauriere et al., 2022; Perrin et al., 2020; Laurière et al., 2022; Guo et al., 2019a; Perrin et al., 2021; Elie et al., 2020; Algumaei et al., 2023). The latter approach is both computationally inefficient in non-trivial settings (Yardim et al., 2023; Laurière et al., 2022), and in many cases is not convergent (as in general it does not induce a contractive operator) (Lauriere et al., 2022; Cui & Koeppl, 2021). Subsequent work, including our own, has therefore moved towards model-free and/or policy-improvement scenarios (Mishra et al., 2023; Laurière et al., 2022; Guo et al., 2023; Subramanian & Mahajan, 2019; Angiuli et al., 2022; Mishra et al., 2020; Cacace, Simone et al., 2021; Perolat et al., 2021; Lee et al., 2021), possibly with learning taking place by observing N-agent *empirical* population distributions (Yardim et al., 2023; Hu & Zhang, 2024; Yongacoglu et al., 2022a).

Most prior works, including algorithms designed to solve the MFG using an N-agent empirical distribution, have also assumed an oracle that can generate samples of the game dynamics (for any distribution) to be provided to the learning agent (Anahtarci et al., 2023; Guo et al., 2019a; 2023; Anahtarci et al., 2019; Fu et al., 2019), or otherwise that the algorithm has direct control over the population distribution at each time step, such as in cases when the agents' policies and distribution are updated on different timescales (Angiuli et al., 2023), with the *fictitious play* method being particularly popular (Xie et al., 2021; Zaman et al., 2023; Mao et al., 2022; Lauriere et al., 2022; Perrin et al., 2020; 2021; Mguni et al., 2018; Cui et al., 2024; Lauriere, 2021; Subramanian & Mahajan, 2019; Angiuli et al., 2022; Tembine et al., 2012; Cardaliaguet, Pierre & Hadikhanloo, Saeed, 2017; Geist et al., 2021; Frédéric Bonnans et al., 2021). In practice, many-agent problems may not admit such arbitrary generation or manipulation (for example, in the context of robotics or

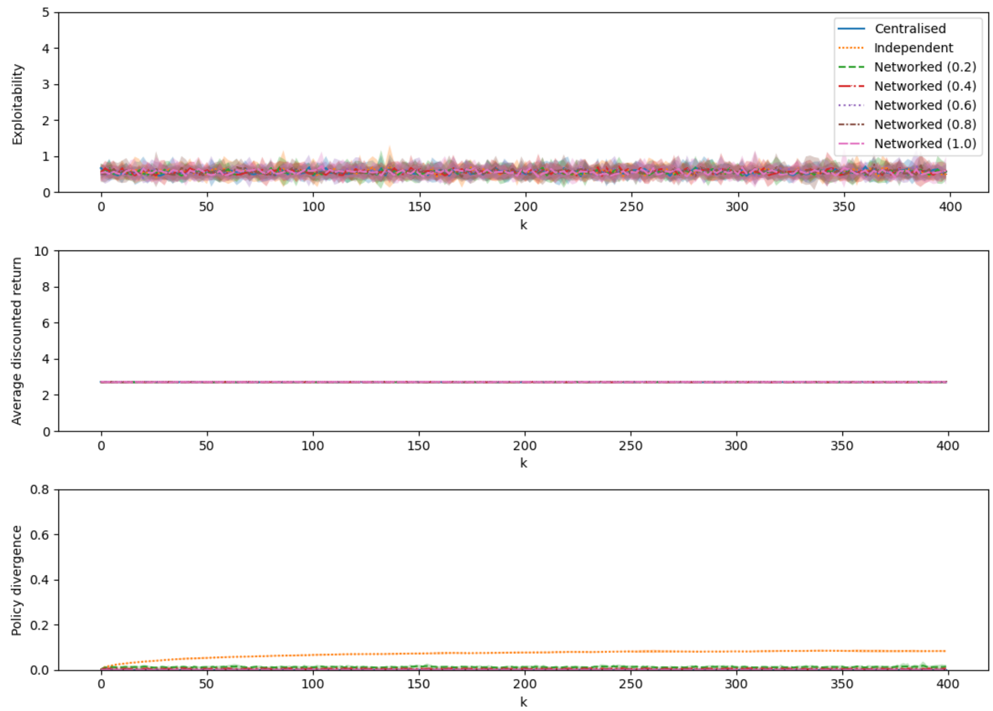

Figure 14: 'Cluster' game with our experience replay buffer removed. There is no noticeable improvement in any of the agents' returns, i.e. no noticeable learning, even after $K = 400$ iterations.

controlling vehicle traffic), and so a desirable quality of learning algorithms is that they update only the agents' policies, rather than being able to arbitrarily reset their states. Learning may thus also need to leverage continuing, rather than episodic, tasks (Sutton & Barto, 2018). Yardim et al. (2023), Yongacoglu et al. (2022a) and our own work therefore present algorithms that seek the MFG-NE using only a single run of the empirical population. Almost all prior work relies on a centralised controller to conduct learning on behalf of all the agents (Xie et al., 2021; Anahtarci et al., 2023; Zaman et al., 2023; Laurière et al., 2022; Guo et al., 2019b). More recent work, including our own, has explored MFG algorithms for decentralised learning with N agents (Yardim et al., 2023; Mguni et al., 2018; Yongacoglu et al., 2022a;b; Grammatico et al., 2015a;b; Parise et al., 2015; Grammatico et al., 2016).

Naturally, inter-agent communication is most applicable in settings where learning takes place along a continuing system run, rather than the distribution being manipulated by an oracle or arbitrarily reset for new episodes, since these imply a level of external control over the population that results in centralised learning. Equally, it is in situations of learning from finite numbers of real, deployed agents (rather than simulated settings) that we are most likely to be concerned with fault tolerance. As such, our work is most closely related to Yardim et al. (2023) and Yongacoglu et al. (2022a), which provide algorithms for centralised and independent learning with empirical distributions along continued system runs: we contribute a networked learning algorithm in this setting. Yongacoglu et al. (2022a) empirically demonstrates an independent learning algorithm when agents observe compressed information about the mean-field distribution as well as their local state, but they do not compare this to any other algorithms or baselines. Yardim et al. (2023) compares algorithms for centralised and independent learning theoretically, but does not provide empirical demonstrations. In contrast, in addition to providing theoretical guarantees, we empirically demonstrate our networked learning algorithm, where agents observe only their local state, in comparison to both centralised and independent baselines, as well as concerning ourselves with the speed of practical convergence and robustness, unlike these works.

Improving the training speed and sample efficiency of (deep) (multi-agent) RL is gaining increasing attention (Yu et al., 2024; Wiggins et al., 2023; Wu et al., 2024a; Patel et al., 2024), though our

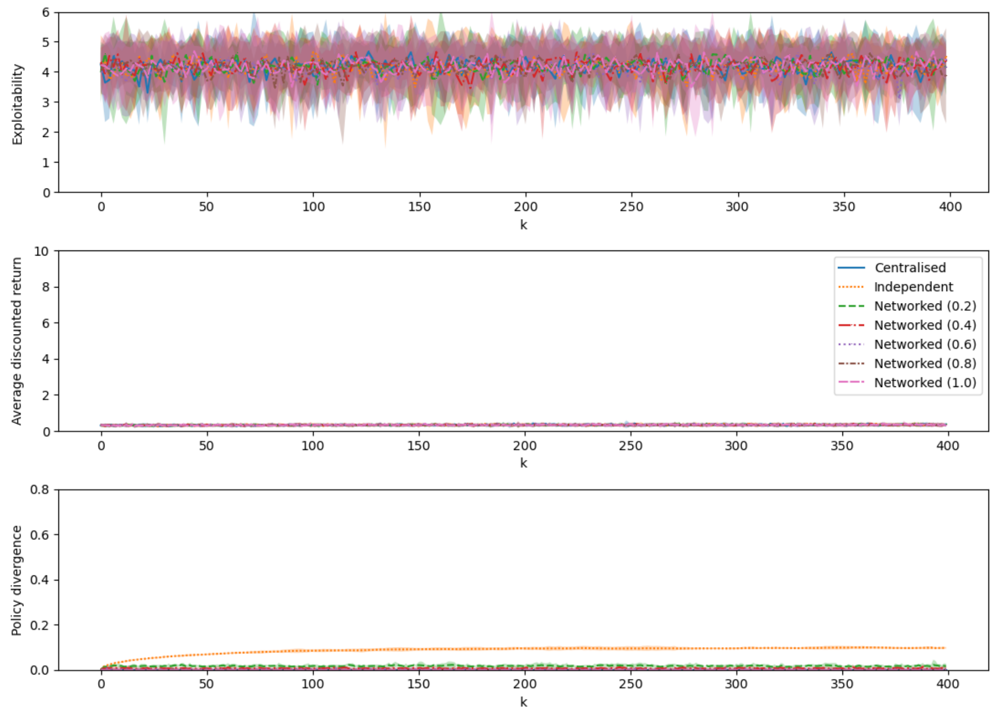

Figure 15: 'Target agreement' game with our experience replay buffer removed. There is no noticeable improvement in any of the agents' returns, i.e. no noticeable learning, even after $K = 400$ iterations.

own work is one of the only on MFGs to be concerned with this. Huang & Lai (2024) trains on a distribution of MFG configurations to speed up inference on unseen problems, but does not learn online in a decentralised manner as in our own work. Similarly, while some attention has been given to the robustness of multi-agent systems to varying numbers of agents, where it is sometimes referred to as 'ad-hoc teaming', 'open-agent systems', 'scalability' or 'generalisation' (Eck et al., 2023), it has more commonly been addressed in MARL (Gao et al., 2024; Dawood et al., 2023) than in MFGs (Wu et al., 2024b). Wu et al. (2024b) presents an MFG approach that allows new agents to join the population during *execution*, but training itself takes place offline in a centralised, episodic manner. Our networked communication framework presented in the current work, on the other hand, allows decentralised agents to join the population during online learning and to have minimal impact on the learning process by adopting policies from existing members of the population through communication.

An existing area of work called 'robust mean-field games' studies the robustness of these games to uncertainty in the transition and reward functions (Bauso & Tembine, 2016; Bauso et al., 2012; 2016; Moon & Başar, 2017; Huang & Huang, 2017; Yang et al., 2018a; Tirumalai & Baras, 2022; Aydın & Saldi, 2023), but does not consider fault-tolerance, despite this being one of the original motivations behind many-agent systems. On the other hand, we focus on robustness to failures and changes in the agent population itself.

We note a similarity between 1. our method for deciding which policies to propagate through the population (described in Sec. 3.4.1) and 2. the computation of evaluation/fitness functions within evolutionary algorithms to indicate which solutions are desirable to keep in the population for the next generation (Eiben & Smith, 2015). Moreover, the research avenue broadly referred to as 'distributed embodied evolution' involves swarms of agents independently running evolutionary algorithms while operating within a physical/simulated environment and communicating behaviour parameters to neighbours (Haasdijk et al., 2014; Trueba et al., 2015), and is therefore even more similar to our setting, where decentralised RL updates are computed locally and then shared with neighbours. In distributed embodied evolution, the computed fitness of solutions helps determine both which are preserved by agents during local updates, and also which are chosen for broadcast or adoption between neighbours (Hart et al., 2015; Fernández Pérez et al., 2018; Fernández Pérez & Sanchez, 2019). Indeed, some works on distributed embodied evolution specifically consider features or rewards relating to the joint behaviour of the whole population (Gomes & Christensen, 2013; Prieto et al., 2016), similar to MFGs. The adjacent research area of cultural/language evolution for swarm robotics (Cambier et al., 2020; 2018; 2021) has similarly demonstrated the combination of evolutionary approaches and multi-agent communication networks for self-organised behaviours in swarms. However, unlike our own work, none of these areas employ reinforcement learning in the update of policies or the computation of the fitness functions.

We preempt objections that communication with neighbours might violate the anonymity that is characteristic of the mean-field paradigm, by emphasising that the communication in our algorithm takes place outside of the ongoing learning-and-updating parts of each iteration. Thus the core learning assumptions of the mean-field paradigm are unaffected, as they essentially apply at a different level of abstraction (a convenient approximation) to the reality we face of N agents that interact within the same environment. Indeed, prior works have combined networks with mean-field theory, such as using a mean field to describe adaptive dynamical networks (Berner et al., 2023).

H LIMITATIONS AND FUTURE WORK

Our algorithm for the networked case (Alg. 1), as well as prior work on the centralised and independent cases (Yardim et al., 2023), all have multiple nested loops. This is a potential limitation for real-world implementation, since the decentralised agents might be sensitive to failures in synchronising these loops. However, in practice, we show that our networked architecture provides redundancy and robustness (which the independent-learning algorithm lacks) in case of learning failures that may result from the necessities of synchronisation (see Appx. E). We have also shown that networked communication in combination with the replay buffer allows us to reduce the hyperparameter M_{td} to 1, essentially removing the inner 'waiting' loop. Nevertheless, our algorithm still features multiple loops, and future work lies in simplifying the algorithms further to aid practical implementation, possibly by techniques such as asynchronous communication (Ma et al., 2024).

Since the MFG setting is technically non-cooperative, we have preempted objections that agents would not have incentive to communicate their policies by focusing on coordination games, i.e. where agents seek to maximise only their individual returns, but receive higher rewards when they follow the same strategy as other agents. In this case they stand to benefit by exchanging their policies with others. Nevertheless, in real-world settings, the communication network could be vulnerable to malfunctioning agents or adversarial actors poisoning the equilibrium by broadcasting untrue policy information. It is outside the scope of this paper to analyse how much false information would have to be broadcast by how many agents to affect the equilibrium, but real-world applications may need to compute this and prevent it. Future research to mitigate this risk might build on work such as Piazza et al. (2024), where 'power regularisation' of information flow is proposed to limit the adverse effects of communication by misaligned agents.

While our MFG *algorithms* are designed to handle arbitrarily large numbers of agents (and theoretically perform better as $N \to \infty$), the *code* for our experiments naturally still suffers from a bottleneck of computational speed when simulating agents that in the real world would be acting and learning in parallel, since the GPU can only process JAX-vectorised elements in batches of a certain size.

Our experiments are based on relatively small toy examples that clearly demonstrate the advantages of our new approach, but which lack the complexity of the real-world applications to which we wish to address the approach. Moreover in our current experiments only the reward function depends on the mean-field distribution, and not the transition function, even though this is possible in theory; we will explore this element in future experiments. It is feasible that in more complex problems, it may not be possible to reduce hyperparameter values to the same extent we have demonstrated in our experimental examples.

Moreover, real-world examples would likely require handling larger and continuous state/action spaces (the latter perhaps building on related work such as Tang et al. (2024)), which in turn may require (non-linear) function approximation. Future work therefore involves incorporating neural networks into our networked communication architecture for oracle-free, non-episodic MFG settings. Extending our algorithms in this way, which would depend on modifying the PMA step (Vieillard et al., 2020; Wu et al., 2024b), would allow us to introduce communication networks to MFGs with *non-stationary* equilibria, in addition to those with larger state/action spaces. Our method for non-stationary games will likely have agents' policies depending both on their local state and also on the population distribution (Mishra et al., 2020; Laurière et al., 2022; Perrin et al., 2022; Carmona et al., 2023), but such a high-dimensional observation object would require moving beyond tabular settings to those of function approximation. The present work demonstrates the benefits of the networked communication architecture when the Q-function is poorly estimated and introduces experience relay buffers to the setting of learning from a continuous run of the empirical system. Both elements are an important bridge to employing (non-linear) function approximation in this setting, where the problems of data efficiency and imprecise value estimation can be even more acute, and where we may want to employ experience replay buffers to provide uncorrelated data to train the neural networks (Zhang & Sutton, 2017). When the policy functions are approximated rather than tabular, our agents would communicate the functions' parameters instead of the whole policy as now.

In our future work with non-stationary equilibria, where agents' policies will also depend on the population distribution, it may be a strong assumption to suppose that decentralised agents with local state observations and limited communication radius would be able to observe the entire population distribution. We will therefore explore a framework of networked agents estimating the empirical distribution from only their local neighbourhood as in (Ganapathi Subramanian et al., 2021), and possibly also improving this estimation by communicating with neighbours (Yongacoglu et al., 2022a), such that this useful information spreads through the network along with policy parameters.