Networked Communication for Decentralised Agents in Mean-Field Games and Control

Patrick Benjamin

Jesus College
University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy

Michaelmas 2025

Abstract

The mean-field framework analyses the limiting case when an infinite number of agents have common reward and transitions functions, and interact with each other not on a per-agent basis, but instead through a distribution over the other agents' states (the mean field). The framework can provide approximate solutions for the equivalent problems involving large but finite populations, which can be much harder to solve in themselves. The framework can therefore be used to tackle computational scalability issues facing other paradigms such as multi-agent reinforcement learning (MARL), with applications considered in a wide variety of real-world problems.

However the framework has traditionally been largely theoretical, and classical approaches usually involve assumptions or algorithmic settings that might be restrictive when applied to very large populations of agents deployed in the real world. In particular, centralised methods have typically been used, despite the fact that a single central coordinator is arguably a strong and undesirable assumption for large populations in the real world. On the other hand, entirely independent agents often learn impractically slowly.

We therefore introduce decentralised, networked communication to the framework and show that it mitigates the drawbacks of both baseline architectures. We first introduce it to the non-cooperative mean-field game (MFG) to compare with existing theoretical sample guarantees for the other architectures, before moving from tabular to function approximation settings, and then ultimately to the cooperative mean-field control (MFC) problem. We similarly build extensive theory for these latter settings, proving that our communication scheme actually permits faster learning than both the independent and the centralised alternatives under certain conditions. In all settings we demonstrate the benefits to learning speed experimentally, and we also provide additional studies showing that our networked populations are more robust than the other architectures to unpredicted shocks that may occur in real-world, non-idealised settings.

Networked Communication for Decentralised Agents in Mean-Field Games and Control

Patrick Benjamin

Jesus College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Michaelmas 2025

Acknowledgements

I am very grateful to my supervisor Professor Alessandro Abate, for being excited about my journey and having faith in my ideas right from when we first met. I have really appreciated his openness and flexibility in letting me explore my research interests and determine my own path, and his willingness to support and champion me while giving me the confidence to develop as an independent researcher.

I am eternally in debt to my course administrator Wendy Poole, who has been magnificent and a saviour in ways I could never have predicted. She has made the PhD experience better in every way and I cannot imagine having done it without her. From making the course extremely sociable and fun, to rescuing me from all sorts of travel scrapes, Wendy has found solutions not just to problems but also to things I did not even realise could be improved. Most importantly, she has been a genuine friend.

Through the PhD I have met many wonderful people. Old and new friends have been a source of invaluable support throughout, and they are too numerous to mention but they know who they are. Nevertheless in this context I would particularly like to thank friends I have made via my course: Kelsey Doerksen, Matthew Jackson, Ben Gutteridge, Tim Reichelt and Alec Edwards. As well as providing technical help and administrative advice, they have all become very important people in my life.

Words cannot describe how grateful I am for my partner and best friend Katie, who is always my greatest supporter. She has been beside me through all the frustrations and joys, and while ceaselessly empathetic she has remained clear-eyed when I could not. I am always thankful for her proofreading, her help in finding the best way to phrase something, and her constant words of encouragement and understanding, and am endlessly grateful to have her as my partner during the PhD and for life.

Abstract

The mean-field framework analyses the limiting case when an infinite number of agents have common reward and transitions functions, and interact with each other not on a per-agent basis, but instead through a distribution over the other agents' states (the mean field). The framework can provide approximate solutions for the equivalent problems involving large but finite populations, which can be much harder to solve in themselves. The framework can therefore be used to tackle computational scalability issues facing other paradigms such as multi-agent reinforcement learning (MARL), with applications considered in a wide variety of real-world problems.

However the framework has traditionally been largely theoretical, and classical approaches usually involve assumptions or algorithmic settings that might be restrictive when applied to very large populations of agents deployed in the real world. In particular, centralised methods have typically been used, despite the fact that a single central coordinator is arguably a strong and undesirable assumption for large populations in the real world. On the other hand, entirely independent agents often learn impractically slowly.

We therefore introduce decentralised, networked communication to the framework and show that it mitigates the drawbacks of both baseline architectures. We first introduce it to the non-cooperative mean-field game (MFG) to compare with existing theoretical sample guarantees for the other architectures, before moving from tabular to function approximation settings, and then ultimately to the cooperative mean-field control (MFC) problem. We similarly build extensive theory for these latter settings, proving that our communication scheme actually permits faster learning than both the independent and the centralised alternatives under certain conditions. In all settings we demonstrate the benefits to learning speed experimentally, and we also provide additional studies showing that our networked populations are more robust than the other architectures to unpredicted shocks that may occur in real-world, non-idealised settings.

Contents

List of Abbreviations

MFG . . . . . Mean-field game

MFC . . . . . Mean-field control

NE . . . . . Nash equilibrium

MFG-NE . . . Mean-field game Nash equilibrium

RL . . . . . Reinforcement learning

MARL . . . . Multi-agent reinforcement learning

OMD . . . . . Online Mirror Descent

MOMD . . . . Munchausen Online Mirror Descent

1 Introduction

Contents

1.1 Motivation

The game-theoretic mean-field framework [1, 2], inspired by theoretical physics, models a representative agent as interacting not with the rest of the population on a per-agent basis, but instead with a distribution over the other agents, known as the *mean field*. The framework typically 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. The mean-field game (MFG) is a non-cooperative scenario where each agent seeks to maximise its individual return, to which the solution is a MFG Nash equilibrium (MFG-NE).

Alternatively we can consider a cooperative scenario called a mean-field *control* (MFC) problem, where the population seeks to maximise a social welfare criterion such as the average return received by the agents. The MFG-NE and the MFC social optimum can respectively be used as approximate solutions to the associated finite-agent game/problem, which are harder to solve in themselves, with the error in the solution reducing as the number of agents N tends to infinity [3–13].

For example, it might be very difficult to directly find a solution for a million agents. However this can be circumvented by finding the solution for the infinite population, where analysis is simplified by taking place in the mathematical limit of population size, and then applying that back to the million agents. Moreover, once the solution is found it does not depend on the size of the deployed population, so some of the million agents could leave without requiring the rest of the population to compute a new solution, or the population could grow to 10 million and the original solution would work even better than before. Furthermore, although the analysis assumes an infinite population, there are numerous ways to represent and generate this population's behaviour. While it might be calculated analytically or by extrapolating from the behaviour of a single generic agent that is assumed to represent the whole population, more recent work has involved simulating the infinite population by drawing finite random samples [14], or indeed by deploying an empirical population consisting of a finite number of agents that is assumed to be representative of the infinite population [15]. The latter case, i.e. using a finite population to simulate the infinite one, might be particularly desirable when the infinite limit is being used expressly to find an approximate solution for the finite-population problem - the same finite population that was of original interest might be used to find the mean-field solution that approximates its own solution. See Rem. 1.2.1 for further discussion.

MFGs and MFC can therefore been used to address the difficulty faced by multi-agent reinforcement learning (MARL), which has seen empirical success in a variety of domains, but has been computationally difficult to scale beyond configurations with agents numbering in the low tens, as the joint state and action

spaces grow exponentially with the number of agents [14, 16–24]. Nevertheless, the value of reasoning about interactions among very large populations of agents (hundreds/millions) has been recognised for real-world applications, and an informal distinction is sometimes drawn between multi- and *many*-agent systems [25–27]. The latter situation can be more useful (as in cases where better solutions arise from the presence of more agents [28–31]), more parallelisable [32], more fault-tolerant [33], or otherwise more reflective of certain real-world systems involving large numbers of decision makers, such as financial markets [21, 34] or smart infrastructures with large populations of autonomous vehicles [31, 35]. Works have therefore considered applying the mean-field framework (particularly MFGs) 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 - note though that these are generally idealised simulated environments serving as proxies for real-world ones. Examples include:

- financial markets [12, 36–49]; ticket pricing [50]; the green economy [51, 52]; electricity markets [53, 54];
- autonomous vehicles [55–59]; traffic signal control [60]; ride-hailing platforms [61]; electric vehicle charging [62–64];
- cryptocurrency mining [65, 66]; edge computing [67–70]; cloud resource management [71, 72]; smart grids and other large-scale cyber-physical systems [73–80];
- swarms [81, 82]; defence [83]; communication networks [84–90]; data collection by UAVs [91];
- social network modelling [92]; crowd modelling [93]; crowdsensing [94];
- pollution regulation [95]; resource management in fisheries [96]; and political governance [97, 98].

We argue that the fact that these studies have generally been modelling exercises rather than actually applied to real-world settings is at least in part due to classical algorithms being conceived in ways that may not be practical for real-world deployments (for example, they try to find solutions via analytical methods or oracles/simulations of an infinite population [8, 14, 19, 20, 71, 99–115]). Equally while MFGs have been well-studied, MFC has received less attention, despite possibly being more useful for engineering collective behaviours to achieve global objectives, such as in consensus, synchronisation, rendezvous, exploration, coverage or task allocation problems [7]. Though we do not claim to have finally solved these issues nor present real-world case studies ourselves, our work is motivated by the desire to remove some of the obstacles towards mean-field algorithms that might be more realistic and practical.

For large, complex many-agent systems of physical decision makers deployed in the real world, such as swarm robotics or autonomous vehicle traffic, it may be infeasible to find mean-field solutions via analytical methods or oracles as they have been traditionally, such that learning must instead be conducted directly by an actual finite population in its deployed environment. In contrast to the restrictive assumptions of many previous methods, we argue that in such deployed scenarios desirable qualities for mean-field algorithms include the following, which motivate the setting of the algorithms we present:

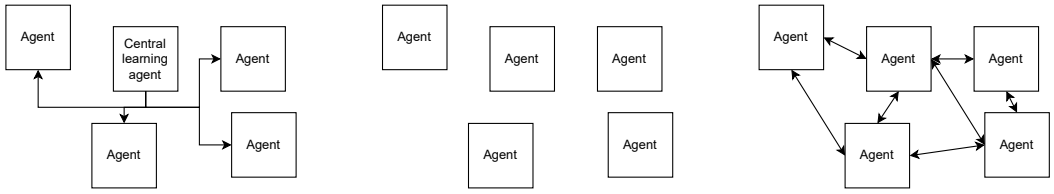
- the ability to learn from an empirical distribution of N agents (i.e. this distribution is generated only by the policies of the agents, rather than being updated or manipulated directly by the algorithm itself or an external oracle/simulator);
- learning online from a single, non-episodic system run (also referred to in other works as a single sample path/trajectory [15, 99]) - i.e. similar to the above, the population is not arbitrarily reset by an external controller, since it might be impractical to repeatedly reset the large deployed population;
- learning without reliance on a model of the system;

- decentralisation;
- fast practical convergence [116];
- and robustness to unexpected failures of decentralised learners or changes in population size [117].

We give a detailed comparison with prior methods, which omit one or more of these desiderata, in Ch. 2. However we elaborate here on one quality in particular, namely the fact that almost all prior work relies on a centralised node to learn on behalf of all the agents. In this context 'centralised' does not necessarily imply global observability of the whole population's actions - which would generally make computation infeasible given the complexity of the problem - but rather that learning is only conducted from the samples of a single representative agent, whose policy updates are assumed to be automatically pushed to the rest of the population by the central node [8, 14, 15, 99, 100, 109, 118–128]. For this reason, whilst 'centralised learning' is the term used in prior works, we generally refer to 'central-agent learning' to reduce confusion.

The use of a central learner in mean-field algorithms naturally reflects the simplifying assumption of the framework, namely that since we are considering a limit distribution of symmetrical agents, we can study a representative agent that interacts with this anonymous population. Some classical works on MFC also justify their centralised methods via the similar but distinct reasoning that MFC problems can be interpreted as optimisation problems from the perspective of a central social planner. In central-agent algorithms in both MFG and MFC, often the empirical mean field of the actual population is not used to compute rewards or transitions, with the central learner instead updating an estimate of the mean field based only on its own policy, which is in turn used as input to oracle-like reward and transition functions [109, 120, 125].

However, recent works on MFGs/MFC, as in other areas of the multi-agent systems community, have recognised that the existence of a central coordinator is a very strong assumption in complex, real-world settings even without global



(a) Central-agent. **(b)** Independent. **(c)** Decentralised network.

Figure 1.1: The three learning architectures for the mean-field framework. The classical approach is the central-agent architecture, but this may be unrealistic in practice and presents a bottleneck and single point of failure. An independent learning architecture avoids these downsides, but has much worse theoretical sample guarantees and empirical learning speed. We propose the decentralised networked architecture, which brings benefits over the other architectures in terms of both learning speed and robustness.

observability, and one that can both restrict scalability by constituting a bottleneck for computation and communication, and reveal a vulnerable single point of failure for the whole system [7, 15, 129–137]. 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, some recent works have explored MFG/MFC algorithms that involve the N individual agents in the empirical population learning policies for themselves without relying on a central node [13, 15, 54, 115, 138–144]. However, those works do not meet one or more of our other desiderata for deployed algorithms. For example they might be model-based [13]; in the MFG case they generally focus on existence proofs for equilibria or 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 compare our research with these other works more fully in the related work in Ch. 2.

Our main contribution is that we introduce decentralised, networked communication to the mean-field framework, allowing us to address all of our desiderata (see Fig. 1.1). Communication networks have had success in other multi-agent settings, removing the reliance on inflexible, centralised structures [129, 131–133, 145–149]. In the chapters that follow we show that as well as allowing agents to learn without

the assumption of access to centrally provided information, the communication network brings two important benefits:

- Populations using our networked communication scheme can learn significantly faster than agents learning entirely independently, and in certain settings they can learn faster even than the central-agent populations.
- The networked architecture affords robustness to unexpected failures of decentralised learners and changes in population size.

1.2 Contributions

Our work introduces networked communication to the mean-field framework in several steps; we describe these, and the chapters into which they fall, in the following. Please note that we present related work in Ch. 2, general preliminaries in Ch. 3 and future work in Ch. 7.

1.2.1 Chapter 4 - MFGs in tabular settings

We firstly introduce our most important use of the communication network, namely a scheme whereby agents can adopt policies communicated to them from neighbours. In this chapter we focus on MFGs with stationary population distributions (*stationary MFGs*, where learning is more tractable than in non-stationary ones) [8, 14, 15, 99, 150, 151]. It may appear counter-intuitive to offer a policy communication algorithm in a setting that is non-cooperative. We offer a number of responses to pre-empt such concerns:

- As already mentioned, most methods for solving MFGs involve a central learner pushing its policy to the rest of the population. This is also a type of communication, and there is no reason selfish agents should want to accept identical policies from a central node any more than they should want to selectively adopt policies communicated by neighbours. We move to the cooperative MFC setting, where agents arguably have broader justification for communicating, in Ch. 6.

- In our experiments in this and the following chapter, we focus on *coordination games*, where agents can increase their individual rewards by following the same strategy as others and therefore inherently have an incentive to communicate policies, even if the MFG setting itself is technically non-cooperative. Nevertheless we find no need to make a distinction in our theoretical analysis, which holds across all types of non-cooperative MFG.

In the idealised theoretical setting, we prove that our networked algorithm's theoretical sample guarantees lie between those of earlier central-agent and independent algorithms [15]. However we show that these theoretical algorithms, although affording comparison between baseline sample guarantees, are not actually able to learn in practical time, and so we extend all three algorithms with experience replay buffers in order to compare the architectures experimentally. In our setting of learning from a continued, non-episodic run of the system, in which Yardim et al. [15] and Yongacoglu et al. [142] are the mostly closely related works, our experience replay buffer is itself a novel contribution. Much of the theoretical analysis in Yardim et al. [15] centres on ensuring the independence of samples that are collected along this single system run and are used once before being discarded. This makes the inclusion of a buffer that is cycled through repeatedly not obvious a priori.

We show empirically that when the agents' Q-functions can be only roughly estimated due to fewer samples/updates, possibly leading to high variance in policy updates, then using the communication network to propagate better-performing policies through the population leads to faster learning than that achieved by agents learning entirely independently, which still hardly appear to learn at all even with the buffer. 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 for the theoretical convergence guarantees will be infeasible in practice. As well as demonstrating our scheme's empirical benefits for learning speed, we conduct additional studies showing the advantages of communication for system robustness.

1.2.2 Chapter 5 - MFGs with function approximation

While our practical networked algorithm in Ch. 4 fulfilled all of our desiderata, it did so only in settings in which the state and action spaces are small enough that the Q-function can be represented by a table, limiting the algorithm's scalability. Moreover, in that work, as in many others on MFGs, agents only observe their local state as input to their Q-function (which defines their policy). This is sufficient when the solved MFG is expected to have a stationary distribution [8, 14, 15, 99, 100, 152]. However, in reality there are numerous reasons why agents may benefit from being able to respond to the current distribution, such as when:

- the solved MFG has a non-stationary equilibrium [20, 153];
- the distribution is subject to so-called 'common noise', i.e. noise that affects the local states of all agents in the same way, making the evolution of the distribution stochastic, such that even if the agent knows the policy used by all other agents, it cannot perfectly predict the evolution of the mean-field distribution, making population-independent policies suboptimal [38, 100, 144, 154, 155];
- agents may begin with any initial mean-field distribution [100, 153, 156]; or,
- the distribution deviates from the equilibrium for some other reason, such that agents need to be able to generalise their response to other (possibly previously unseen) distributions [100].

Recent work has thus increasingly focused on these more general settings where it is necessary for agents to have population-dependent policies (sometimes also called *master policies*) which depend on both the mean-field distribution and their local state [20, 100, 153–156]. The distribution is a large, high-dimensional observation object, taking a continuum of values. Therefore a population-dependent Q-function cannot be represented exactly in a table and must be approximated.

We therefore add to our list of desirable qualities for mean-field algorithms the following: they should permit function approximation to allow scalability to high-dimensional observations (including the option to include the mean field in the input to policies). Accordingly in this chapter we introduce function approximation to the MFG setting of decentralised agents learning online from a single, non-episodic run of the empirical system, allowing this setting to handle larger state spaces and to accept the mean-field distribution as an observation input.

We demonstrate that in this setting networked communication brings two specific benefits over the purely independent setting, while also removing the undesirable assumption of a central learner. Firstly, similarly to Ch. 4, when the Q-function is approximated rather than exact, some updates lead to better performing policies than others. Allowing networked agents to propagate better performing policies through the population leads to faster learning than in the purely independent case and now also very often *even than in the central-agent case*, as we show both theoretically and empirically. Secondly, we argue that in the real world it is unrealistic to assume that decentralised agents, endowed with local state observations and limited (if any) communication radius, would be able to observe the global mean-field distribution and use it as input to their Q-functions / policies. We therefore further contribute two setting-dependent algorithms by which decentralised agents can estimate the global distribution from local observations, and further improve their estimates by communication with neighbours.

We again focus on coordination games in our experiments, to pre-empt concerns about communication in a setting that is technically non-cooperative. Nevertheless we do show experimentally that self-interested communicating agents can obtain higher returns than independent agents even in an anti-coordination game, indicating that they may still have incentive to communicate.

1.2.3 Chapter 6 - MFC with function approximation

In this chapter we adapt our MFG algorithms to introduce networked communication to the cooperative MFC setting, where agents arguably have even broader justifi-

cation for communicating. In addition to the two uses of the network introduced in the previous two chapters (for proliferating policies and for estimating the empirical mean field), we now introduce a third use, namely for estimating the global average reward from local communication so as to optimise social welfare without needing to observe global information. As such our MFC algorithm again fulfils all of our proposed desiderata. Our theoretical analysis from previous chapters of networked communication in the non-cooperative MFG setting does not extend trivially to MFC, so in this chapter we contribute new theoretical proofs showing that decentralised policy exchange allows networked populations to learn faster than both the independent *and* the central-agent alternatives in the MFC setting, across different classes of cooperative game (coordination and anti-coordination). We also demonstrate this finding empirically in numerous games, as well as contributing an empirical study of the algorithms' robustness to communication failures, along with several ablation studies.

1.2.4 Additional comments

Remark 1.2.1. As we will see in Secs. 4.3 and 5.3, solving the theoretical MFG problem involves finding a single policy that, when given to all agents in the infinite population, best responds to the resulting mean-field distribution. Similarly, as we will see in Sec. 6.3, an optimal solution to the theoretical MFC problem is a single policy that, when followed by all agents in the infinite population, maximises the population's expected return. We give two ways to conceive of our work, illustrated in Fig. 1.2, which mirror and make more explicit the similar motivations underpinning many other works on MFGs and MFC [7, 13, 42, 43, 47, 53, 70, 80, 122, 142, 158–164].

1. Firstly, while previous works might make unrealistic assumptions about access to an oracle for the infinite population, we contribute algorithms that allow the solution to a MFG/MFC problem to be learnt using the empirical distribution of a decentralised finite population. Note that it is unnecessary (and may be

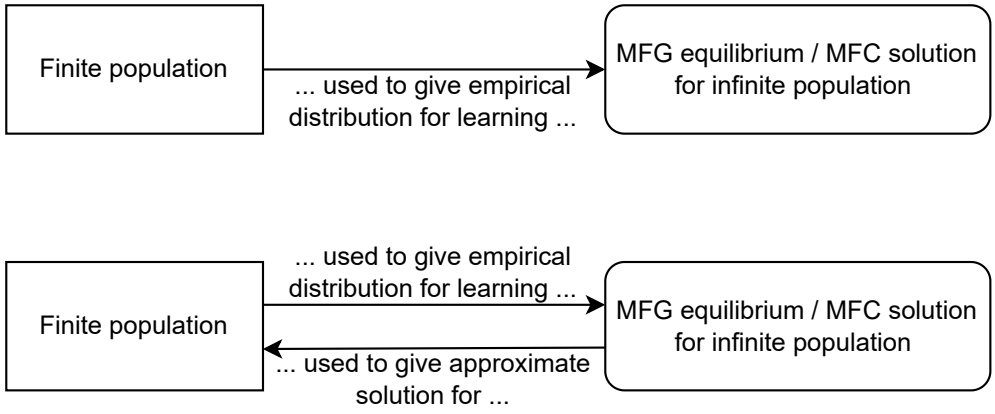


Figure 1.2: Two possible ways to conceive of our work regarding the relationship between the infinite- and finite-population games / control problems, described in Rem. 1.2.1. Note that in the MFG case, using the finite empirical population to try to learn a single MFG-NE policy $\boldsymbol{\pi} = (\pi^*, \dots, \pi^*)$ that is to be followed by the whole infinite population (Def. 4.3.5) is *not* the same as directly finding $\boldsymbol{\pi}^* = (\pi^1, \dots, \pi^N)$, i.e. the tuple of *individual* policies that gives the finite-population NE in Def. 4.3.3, a problem known to be hard [16–18, 21–23]. Similarly, in the MFC case, using the finite empirical population to learn the single-policy MFC social optimum $\boldsymbol{\pi} = (\pi^*, \dots, \pi^*)$ for the infinite population (Def. 6.3.6) is *not* the same as directly finding $\boldsymbol{\pi}^* = \arg\max_{\boldsymbol{\pi} \in \Pi^N} V^{pop}(\boldsymbol{\pi}, \mu_{\bar{t}}) = (\pi^1, \dots, \pi^N)$, i.e. the tuple of *individual* policies that maximises the expected finite population-average return in Def. 6.3.3, which is again known to be hard [7, 157].

impractical) to assume that the decentralised agents always follow a single identical policy throughout training, a logic also followed by earlier works [15].

2. Alternatively, we may have originally been interested in finding a NE or a cooperative solution for a large, finite population, but, due to the scalability issues of learning approaches like MARL, were forced to turn to the mean-field framework to find a policy that gives an approximate solution to the finite-population problem. We contribute algorithms that allow the deployed finite population to find the MFG/MFC solution that in turn approximately solves the original problem, without unrealistic assumptions about centralised training. Under this framing, it may matter less whether all agents follow a single policy in practice.

Remark 1.2.2. We further pre-empt 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 framework are unaffected, as they essentially apply at a different level of abstraction to the reality we may face of a deployed empirical population of N decentralised agents that interact within the same environment. Indeed, prior works have already combined networks with mean-field theory in different ways, such as using a mean field to describe adaptive dynamical networks [165].

1.3 Publications

All of the work in this thesis has been published in or submitted for publication to high-quality venues; these works are listed below. I am the sole author of all of these works, alongside my supervisor Professor Alessandro Abate.

Chapter 4 is based on the following paper, which at the time of writing is under review at TMLR:

- Patrick Benjamin and Alessandro Abate. *Networked Communication for Decentralised Agents in Mean-Field Games.* arXiv preprint arXiv:2306.02766 (2023).

Chapter 5 is based on the following paper, which has been accepted at AAMAS 2026.

- Patrick Benjamin and Alessandro Abate. *Networked Communication for Mean-Field Games with Function Approximation and Empirical Mean-Field Estimation.* arXiv preprint arXiv:2408.11607 (2024).

Versions of this paper have been accepted at many workshops, **winning the best paper award** at the first:

- Patrick Benjamin and Alessandro Abate. *Networked Communication for Mean-Field Games with Function Approximation and Empirical Mean-Field Estimation.* Adaptive and Learning Agents (ALA) Workshop at AAMAS 2025 (winner of best paper award).

- Patrick Benjamin and Alessandro Abate. *Improving Real-World Applicability of Networked Mean-Field Games using Function Approximation and Empirical Mean-Field Estimation.* Multi-Agent AI in the Real World (MARW) Workshop at AAAI 2025.

- Patrick Benjamin and Alessandro Abate. *Addressing the MARL Scalability Problem in Autonomous Transport using Practical Mean-Field Games.* Multi-Agent reinforcement Learning for Transportation Autonomy (MALTA) Workshop at AAAI 2025.

- Patrick Benjamin and Alessandro Abate. *Networked Communication in Mean-Field Games with Function Approximation and Empirical Mean-Field Estimation.* 2nd Workshop on Game AI Algorithms and Multi-Agent Learning (GAAMAL) at IJCAI 2025.

- Patrick Benjamin and Alessandro Abate. *Engineering Practical Mean-Field Games with Networked Communication, Function Approximation, and Population Estimation.* 13th International Workshop on Engineering Multi-Agent Systems (EMAS) at AAMAS 2025.

- Patrick Benjamin and Alessandro Abate. *Networked Communication for Mean-Field Games with Function Approximation and Empirical Population Estimation.* Games, Agents, and Incentives Workshop (GAIW) at AAMAS 2025.

- Patrick Benjamin and Alessandro Abate. *Networked Communication for Mean-Field Games with Function Approximation and Empirical Mean-Field Estimation.* 16th Workshop on Optimization and Learning in Multiagent Systems (OptLearnMAS-25) at AAMAS 2025.

- Patrick Benjamin and Alessandro Abate. *Deep Learning and Global-State Estimation for Arbitrarily Large Populations of Networked Agents*. 2025 Multi-disciplinary Conference on Reinforcement Learning and Decision Making (RLDM).

Chapter 6 is based on the following paper, which at the time of writing is under review at TMLR:

- Patrick Benjamin and Alessandro Abate. *Networked Communication for Decentralised Cooperative Agents in Mean-Field Control*. arXiv preprint arXiv:2503.09400 (2025).

Versions of this paper have been accepted at several workshops:

- Patrick Benjamin and Alessandro Abate. *Networked Communication for Decentralised Cooperative Agents in Mean-Field Control*. 2nd Workshop on Social Choice and Learning Algorithms (SCaLA-25) at IJCAI 2025.
- Patrick Benjamin and Alessandro Abate. *Networked Communication for Decentralised Cooperative Agents in Mean-Field Control*. 2nd Coordination and Cooperation in Multi-Agent Reinforcement Learning (CoCoMARL) Workshop at RLC 2025.

2
Related work

This chapter serves as a general related work for the thesis as a whole, and also for Ch. 4 in particular, where we first introduce networked communication into the the mean-field framework (specifically tabular MFGs). Since, as we noted in Sec. 1.1, MFGs have received more attention in the literature than MFC, there is a slight skew towards mentions of MFGs in this section, though the contentions we make generally refer to MFC as well unless we specify otherwise. Nevertheless please see also Secs. 5.2 and 6.2 for additional related work more specific to their respective chapters, i.e. the non-tabular MFG setting and MFC.

In Sec. 1.1 we gave several qualities that we argue are desirable for mean-field algorithms if they are eventually to be more applicable to complex, deployed scenarios in the real-world. Conversely, works on MFGs have traditionally been largely theoretical [1, 2] (often works do not present any empirical results [15, 150, 166–168]), and methods for finding equilibria have often relied on assumptions that are too strong for real-world applications. The MFG-NE is classically 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, as in Def. 4.3.5 below [42, 46–50, 57, 64, 80, 83, 87, 96, 97, 123, 127, 144, 151, 163, 167, 169–191]; crucially, these methods generally rely on the assumption of an infinite population [100]. Early

work solved the coupled equations using numerical methods that did not scale well for more complex state and action spaces [101–104]; 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) [8, 106, 192–198], and/or computed a best-response to the mean-field distribution [2, 19, 20, 100, 106, 107, 199, 200]. The latter approach is both computationally inefficient in non-trivial settings [15, 100], and in many cases is not convergent (as in general it does not induce a contractive operator) [20, 201]. Subsequent work, including our own, has therefore moved towards model-free and/or policy-improvement scenarios [75, 100, 105, 108, 109, 202–205], possibly with learning taking place by observing N-agent *empirical* population distributions [11, 15, 142].

Most prior works, including algorithms designed to solve MFGs 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 [8, 105, 106, 206, 207], or otherwise that the algorithm (rather than agents' policies) has direct control over the population distribution at each time step [208–210], such as cases where the agents' policies and distribution are updated on different timescales [24, 125], with the *fictitious play* method being particularly popular [14, 19, 20, 71, 99, 107–115, 211, 212]. In practice, many-agent problems may not admit such arbitrary generation or manipulation (for example, in the context of robotics or 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 [213]. Yardim et al. [15], Yongacoglu et al. [142] and our own work therefore present algorithms that seek the MFG-NE using only a single run of the empirical population.

Decentralised 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 settings able to simulate infinite populations) that we are most likely to be concerned with fault tolerance. Networked communication therefore naturally fits within our desired qualities for mean-field algorithms, and our focus on this setting means that our work is most closely related to Yardim et al. [15] and Yongacoglu et al. [142], which provide algorithms for centralised and independent learning from empirical distributions along non-episodic system runs. Yongacoglu et al. [142] 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. [15] 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 algorithms (where in Ch. 4 agents only need to observe 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.

More generally, a number of works refer to 'decentralisation' in MFGs, but often in a different sense to our understanding of it. In particular, many works that say they consider decentralisation actually learn/derive policies via a centralised method (often involving a representative player), and simply mean that agents' policies are *executed* independently based on local information, which we take as a given across our learning architectures [48, 53, 90, 163, 189, 214] - see Sec. 6.2 for a similar dynamic in MFC. He and Liu [54] use reinforcement learning (RL) to solve a two-level mean-field problem, where there is a MFG between 'aggregators', each of which is solving a local MFC problem. They solve the MFG via decentralised learning by the N aggregators, but each aggregator solves its MFC problem in a centralised manner via the assumption of a single agent that is representative of the heterogenous population. Moreover, they prove the existence of and convergence to a unique equilibrium, but do not provide sample guarantees or a convergence rate, as we do in Ch. 4. Other works involve decentralisation in learning but

under different MFG settings to our own: Li et al. [49], Ghosh [190], Xu et al. [191] derive controls in a decentralised way, but rely on a model of the environment, while Yardim et al. [161] uses independent learning but not via RL, as they focus on repeated play of static, stateless games.

Improving the training speed and sample efficiency of (deep) (multi-agent) RL is gaining increasing attention [215–218], though our own work is one of the only on the mean-field framework to be concerned with this. Huang and Lai [219] 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 changes in population size, where it is sometimes referred to as 'ad-hoc teaming', 'open-agent systems', 'scalability' or 'generalisation' [31], it has more commonly been addressed in MARL [220, 221] than in MFGs [153]. Wu et al. [153] 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, on the other hand, allows decentralised agents to join the empirical population during online learning and to have minimal impact on the learning process by adopting policies from existing members of the population through communication (Sec. 4.7.4.3).

An existing area of work called *robust mean-field games* studies the robustness of these games to uncertainty in the transition and reward functions [73, 222–228], but does not consider resilience to agent update failures, despite fault tolerance 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.

For the sake of defining the scope of terms, we do not consider what we refer to as the 'mean-field framework' to encompass the related but distinct research area called *mean-field RL* [229–231]. While drawing inspiration from similar sources, mean-field RL falls outside of the game-theoretic MFG/MFC paradigm. It is instead a type of MARL, and considers a mean over actions (originally by averaging pairwise interactions between agents [229]) rather than a distribution over states, as in our

case. This generally leads to lag and a chicken-and-egg problem, whereby agents respond to the other agents' previous actions, rather than their current states. The existence of this distinct area can be a source of confusion in nomenclature: while we use RL as a model-free learning approach in MFGs/MFC, this is not the same as doing mean-field RL, and MFGs/MFC can be, and classically were, solved without RL, as we discuss above. Some works have considered similar features to those we are interested in, such as decentralisation and estimation from local neighbourhoods, but in this distinct area of mean-field RL [230]. We draw attention in particular to the work by Subramanian et al. [232], who develop a framework they refer to as 'Decentralized Mean Field Games [sic]' but which they also emphasise is distinct from both MFGs and mean-field RL, and which removes the symmetry and anonymity of agents (agents may have different reward functions, and agent indices are retained). Therefore, despite its name, this setting is different from our own.

We note a similarity between 1. our method for deciding which policies to propagate through the population 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 [233, 234]. 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 [235, 236], 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 [237–239]. Indeed, some works on distributed embodied evolution specifically consider features or rewards relating to the joint behaviour of the whole population [240, 241], similar to the mean-field framework. The adjacent research area of cultural/language evolution for swarm robotics [242–244] 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 RL in the update of policies or the computation of the fitness functions.

Our work also shares parallels with *population-based training* [245], an approach that is likewise related to evolutionary algorithms. Population-based training involves optimising neural networks by performance-based transfer of parameters and hyperparameters among a population of concurrent processes. We are interested in the interactive behaviour of the population itself rather than simply using it for parallelising the optimisation.

3
General preliminaries

Contents

Please also see Secs. 4.3, 5.3 and 6.3 for more specific preliminaries for their respective chapters.

3.1 Mean-field games/control

Secs. 4.3, 5.3 and 6.3 contain very similar definitions for the MFG/MFC and tabular/non-tabular settings. Due to some slight setting-specific differences, we restate the relevant definitions in each chapter for clarity rather than giving them together here. Nevertheless, we introduce here the following notation that is common to all chapters.

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 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 of length $|\mathcal{S}|$ denoting the empirical categorical state distribution of the N agents at time t.

3.2 Networks

Our decentralised empirical population exhibits two time-varying graphs (with the second only used from Ch. 5 onwards), where the links between agents that make up the network may change at each time step t. The basic definition of such a network is as follows:

Definition 3.2.1 (Time-varying network)**.** The time-varying 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 $i \in \{1, \ldots, N\}$, and the edge set $\mathcal{E}_t \subseteq \{(i,j) : i,j \in \mathcal{N},\ i \neq j\}$ is the set of undirected links present at time t. A network's *diameter* $d_{\mathcal{G}_t}$ is the maximum of the shortest path length between any pair of nodes.

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 [246]. A collection is *jointly connected* if its members' union is connected.

One of these graphs $\mathcal{G}_t^{comm}$ is a communication network that defines which agents can communicate information to each other at time 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 is general to all settings, and can depend on other factors that may not depend on the agents' position in space or state s_t^i. For example, agents may be connected over long distances via satellites or the internet, and 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.

The communication network is used in all of Chs. 4, 5 and 6. From Ch. 5 we additionally become interested in agents being able to estimate the global mean field from local information. In principle, agents can use this same communication network $\mathcal{G}_t^{comm}$ to receive information about others' states in order to estimate the mean field. However for generality and modularity we say instead that agents exhibit a second network $\mathcal{G}_t^{obs}$. This is a graph defining which agents can observe each other's states, which we use in general settings for estimating the mean field from local information. The structure of the two networks may be identical (e.g. if embodied agents can both observe the position (state) of, and exchange information with, other agents within a certain physical distance from themselves), or different (e.g. if agents can observe the positions of nearby agents, but only exchange information with agents by which they are linked via satellite, which may connect agents over long distances).

We also define an alternative version of the observation graph that is useful in a specific subclass of environments, which can most intuitively be thought of as those where agents' states are positions in physical space. When this is the case, we usually think of agents' ability to observe each other as depending more abstractly on whether states are visible to each other. We define this visibility graph as follows:

Definition 3.2.2 (Time-varying state-visibility graph)**.** The time-varying state visibility graph $(\mathcal{G}_t^{vis})_{t\geq 0}$ is given by $\mathcal{G}_t^{vis} = (\mathcal{S}', \mathcal{E}_t^{vis})$, where $\mathcal{S}'$ is the set of vertices representing the environment states $\mathcal{S}$, and the edge set $\mathcal{E}_t^{vis} \subseteq \{(m,n) : m,n \in \mathcal{S}'\}$ is the set of undirected links present at time t, indicating which states are visible to each other.

We view an agent in s as able to obtain a count of the number of agents in s' if s' is visible to s. The benefit of this graph $\mathcal{G}_t^{vis}$ over $\mathcal{G}_t^{obs}$ is that there is mutual exclusivity: either an agent in state s is able to obtain a total count of all of the agents in state s' (if s' is visible to s), or it cannot obtain information about any agent in state s' (if those states are not visible to each other). Additionally, this graph permits an agent in state s to observe that there are *no* agents in state s'

as long as s' is visible to s. These benefits are not available if the observability graph is defined strictly between agents as in $\mathcal{G}_t^{obs}$, such that using $\mathcal{G}_t^{vis}$ facilitates more efficient estimation of the global mean-field distribution from local information in settings where $\mathcal{G}_t^{vis}$ is applicable (see Sec. 5.5).

In Ch. 5 we present algorithms that form an initial estimate of the global empirical mean field (to serve as an observation input for agents' Q-/policy-networks) via the observability graph $\mathcal{G}_t^{obs}$ or the visibility graph $\mathcal{G}_t^{vis}$, before refining this estimate via the communication graph $\mathcal{G}_t^{comm}$.

4

Networked Communication in Mean-Field Games

Contents

4.1 Introduction

We begin by recapping the content and contributions of this chapter, first stated in Sec. 1.2.1. We introduce networked communication to stationary MFGs, for which the solution concept is the 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 (see Fig. 4.1).

We prove that our networked algorithm's theoretical sample guarantees lie between those of earlier central-agent and independent algorithms. Next, to compare the architectures experimentally, we extend all three theoretical algorithms with experience replay buffers, without which we found them unable to learn in practical time. We show empirically that when the agents' Q-functions can be only roughly estimated due to fewer samples/updates, possibly leading to high variance in policy updates, then using the communication network to propagate better-performing policies through the population leads to faster learning than that achieved by agents learning entirely independently, which still hardly appear to learn at all even with the buffer. 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. As well as demonstrating our scheme's

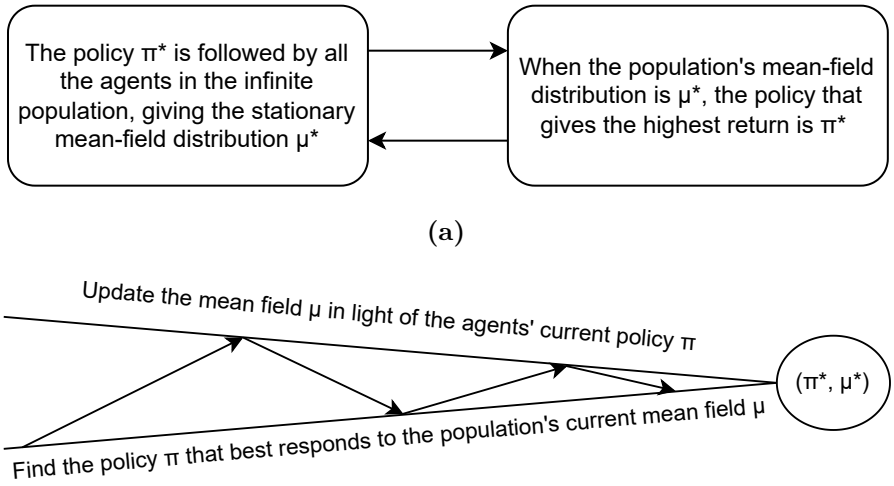


Figure 4.1: a) As formalised in Def. 4.3.5, an optimal solution to a MFG is a policy that best responds to the stationary mean-field distribution that arises when all agents follow that same policy. b) The solution can therefore be seen as the fixed point of two operators: finding the mean-field distribution that arises when agents follow a given policy and finding the policy that best responds to a given mean field.

empirical benefits for learning speed, we conduct additional studies showing the advantages of communication for system robustness.

To pre-empt conceptual concerns about whether selfish agents would have incentive to communicate in non-cooperative MFGs, our experiments focus on coordination games, where even selfish agents can increase their individual rewards by following the same strategy as others. Thus this work can be applied to real-world coordination games in e.g. traffic signal control, formation control in swarm robotics, and consensus and synchronisation e.g. for sensor networks [247]. Nevertheless we find no need to make a distinction in our theoretical analysis, which holds across all types of non-cooperative MFG.

In summary, our contributions include the following:

- We prove that a theoretical version of our networked algorithm (Alg. 1) has sample guarantees bounded between those of central-agent and independent algorithms for learning from a 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 contribute a policy-update stability guarantee (Sec. 4.5).

- We show experimentally that all three theoretical algorithms do not seem to learn at all in any practical time (Sec. 4.7.4.1). We therefore modify all three (Alg. 2, Sec. 4.6) to make learning feasible by including an experience replay buffer, allowing us to contribute the first empirical demonstrations of learning in all three architectures.

- Our experiments demonstrate that in practical settings our communication scheme can markedly benefit learning speed over the independent case, sometimes performing similarly to the centralised case while removing the restrictive assumption of the latter. We also show that via our practical modifications we can learn without enforcing several of the algorithms' other theoretical assumptions (a goal shared by other works on practical MFG algorithms [211]) (Sec. 4.7.4).

- We provide ablations and additional empirical studies showing that our decentralised communication architecture brings further benefits over both the central-agent and independent alternatives in terms of robustness to unexpected update failures and changes in population size. For further discussion of the relevance of these scenarios in large multi-agent systems, see Sec. 4.7.4.3.

The rest of this chapter is structured as follows. We give preliminaries in Sec. 4.3. We present our theoretical algorithms in Sec. 4.4 and theoretical results in Sec. 4.5. We give enhancements to the algorithms which are necessary for learning in practical time in Sec. 4.6, and provide experiments and discussion in Sec. 4.7. We conclude in Sec. 4.8.

4.2 Related work

Ch. 2 contains the related work relevant to this chapter.

4.3 Preliminaries

4.3.1 Core definitions

We use the notation introduced in Sec 3.1, as well as the following. The sets $\mathcal{S}$ and $\mathcal{A}$ are equipped with the discrete metric $d(x, y) = \mathbb{1}_{x \neq y}$. 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 in our experiments 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 many earlier works [8, 15, 54, 161, 167, 168, 200, 201, 248–250], 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 by regularisation can bias the NE [11, 15, 20, 250, 251]. We show in our experiments that we are able to reduce λ to 0 with no detriment to convergence.

We now 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 $\mathcal{U}_{\Delta h} := \{u \in \Delta_{\mathcal{A}} : h(u) \geq h_{\max} - \Delta h\}$.

Having given this notation, we now formalise the symmetric anonymous game involving N agents.

Definition 4.3.1 (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. $\forall i \in \{1, \ldots, N\}$

$$s^i_{t+1} \sim P(\cdot|s^i_t, a^i_t, \hat{\mu}_t) \quad \textit{and} \quad r^i_t = R(s^i_t, a^i_t, \hat{\mu}_t).$$

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

We now formalise the expected discounted returns of each agent in an N-player symmetric anonymous game.

Definition 4.3.2 (N-player discounted regularised return)**.** With joint policies $\boldsymbol{\pi} := (\pi^1, \ldots, \pi^N) \in \Pi^N$, 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, $\forall i, j \in \{1, \ldots, N\}$,

$$\Psi^i_h(\boldsymbol{\pi}, \upsilon_0) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t (R(s^i_t, a^i_t, \hat{\mu}_t) + h(\pi^i(s^i_t)))\Big|_{\substack{s^j_0 \sim \upsilon_0 \\ a^j_t \sim \pi^j(s^j_t) \\ s^j_{t+1} \sim P(\cdot|s^j_t, a^j_t, \hat{\mu}_t)}}\right].$$

This allows us to formalise the solution concept for the non-cooperative N-player symmetric anonymous game, namely the (approximate) NE.

Definition 4.3.3 (δ-NE)**.** Say $\delta > 0$ and $(\pi, \boldsymbol{\pi}^{-i}) := (\pi^1, \ldots \pi^{i-1}, \pi, \pi^{i+1}, \ldots, \pi^N) \in \Pi^N$. An initial distribution $\upsilon_0 \in \Delta_{\mathcal{S}}$ and an N-tuple of policies $\boldsymbol{\pi} := (\pi^1, \ldots, \pi^N) \in \Pi^N$ form a δ-NE ($\boldsymbol{\pi}$, υ_0) if

$$\Psi^i_h(\boldsymbol{\pi}, \upsilon_0) \geq \max_{\pi \in \Pi} \Psi^i_h((\pi, \boldsymbol{\pi}^{-i}), \upsilon_0) - \delta \quad \forall i \in \{1, \ldots, N\}.$$

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 a MFG - as V, rather than Ψ as in the finite N-agent case.

Definition 4.3.4 (Mean-field discounted regularised return)**.** For a policy-population pair $(\pi, \mu) \in \Pi \times \Delta_{\mathcal{S}}$,

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

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. We now introduce the solution concept of this stationary MFG.

Definition 4.3.5 (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:

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

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 [15, 20]:

Proposition 4.3.6 (N-player NE and MFG-NE (Thm. 1, [8]))**.** *If (π^*, μ^*) is a MFG-NE, then, under certain Lipschitz conditions [8], 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 4.3.7. We can show 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 [12, 15, 161].

4.3.2 Further technical conditions for algorithms and theorems

Our theoretical results, which compare our networked algorithm with the centralised and independent alternatives from Yardim et al. [15], rely on several further definitions and assumptions from their work. We give these now to allow us to introduce our learning operator for our algorithm in Sec. 4.4, in advance of the theoretical analysis in Sec. 4.5. These formalisations lay the groundwork

that allows the optimality and stability conditions, which define the MFG-NE in Def. 4.3.5, to hold.

4.3.2.1 Population update operators

The following characterisations ensure that the evolution of the population is convergent for a given policy.

Assumption 4.3.8 gives Lipschitz constants that provide smoothness conditions on the transition and reward functions P and R - this is a standard theoretical assumption in previous work [15]. These constants in turn ensure that the population-evolution and policy-update operators below are smooth and hence contractive, guaranteeing convergence.[1] If the Lipschitz condition failed - for example in environments with hard cliffs in the reward function across the reachable region, abrupt discontinuities in the transition kernel, or unbounded reward magnitudes - the operators Γ_{pop} and Γ_{η}^{md} would no longer be guaranteed to be Lipschitz, and the contraction-based convergence arguments underpinning Thms. 4.5.2 and 4.5.3 would no longer apply.

Assumption 4.3.8 (Lipschitz continuity of P and R)**.** There exist constants $K_\mu, K_s, K_a, L_\mu, L_s, L_a \in \mathbb{R}_{\geq 0}$ such that $\forall s, s' \in \mathcal{S}, \forall a, a' \in \mathcal{A}, \forall \mu, \mu' \in \Delta_{\mathcal{S}}$,

$$||P(\cdot|s,a,\mu) - P(\cdot|s',a',\mu')||_1 \;\leq\; K_\mu||\mu - \mu'||_1 \;+\; K_s d(s,s') \;+\; K_a d(a,a'),$$

$$|R(s,a,\mu) - R(s',a',\mu')| \;\leq\; L_\mu||\mu - \mu'||_1 \;+\; L_s d(s,s') \;+\; L_a d(a,a').$$

[1] In the games we introduce in our experiments (Sec. 4.7.1) the transition kernel P does not depend on $\hat{\mu}$ at all, so the K_μ-dependence of the Lipschitz condition on P holds trivially with $K_\mu = 0$ and only the Lipschitz conditions on P in s and a remain - and those are themselves trivial on a finite state-action space, since any bounded function on a finite set is Lipschitz with a constant given by the maximum range divided by the minimum nonzero distance between distinct points. The dependence of R on $\hat{\mu}$ in our games enters only through the value $\hat{\mu}_t(s_t^i)$ of the empirical distribution at the agent's own state, which is always at least $1/N$ by construction (agent i itself contributes one count to $\hat{\mu}_t(s_t^i)$). The relevant domain for $\hat{\mu}_t(s_t^i)$ is therefore the bounded-away-from-zero interval $[1/N, 1]$ rather than the full $[0, 1]$, which is what is needed to tame otherwise singular reward dependencies: the log-style 'cluster' reward used in Sec. 4.7.1 admits a finite Lipschitz constant of order N on $[1/N, 1]$ (even though it blows up as $\hat{\mu} \to 0$ and so is not Lipschitz on the unrestricted simplex), and the piecewise-defined 'target agreement' reward is Lipschitz away from its jump discontinuities, with Lipschitz constant 1 on each piece. We therefore implicitly require Assumption 4.3.8 only on the region of $\mathcal{S} \times \mathcal{A} \times \Delta_{\mathcal{S}}$ actually reached by the algorithm, which is sufficient since the contraction-based proofs only invoke the Lipschitz inequalities at points along the system run. We do not measure $L_s, L_a, L_\mu, K_s, K_a, K_\mu$ explicitly in any of our experiments.

The single-step operator tells us how the mean field evolves by one step when the whole population uses a certain policy, which allows us in turn to give the stable population operator as the fixed point of repeated updates. This is later plugged into the policy-improvement operator, allowing us to obtain the fixed-point consistency and hence the stationary MFG-NE.

Definition 4.3.9 (Population update operator)**.** The single-step population update operator $\Gamma_{pop} : \Delta_{\mathcal{S}} \times \Pi \to \Delta_{\mathcal{S}}$ is defined as, $\forall s \in \mathcal{S}$:

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

We will use the short hand notation

$$\Gamma^n_{pop}(\mu, \pi) := \underbrace{\Gamma_{pop}(\dots \Gamma_{pop}(\Gamma_{pop}(\mu, \pi), \pi), \dots, \pi)}_{n \text{ times}}.$$

We recall in the following lemma that Γ_{pop} is known to be Lipschitz [8, 118]. By ensuring that the population updates are smooth, we can in turn ensure that they are contractive, giving a unique and stable steady population via Assumption 4.3.11 below. This in turn ensures convergence.

Lemma 4.3.10 (Lipschitz population updates)**.** 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 $\forall \pi \in \Pi$. We therefore give the following assumption, which is common in previous works [8, 15, 99, 118]:

Assumption 4.3.11 (Stable population)**.** Population updates are stable, i.e. $L_{pop,\mu} < 1$.

If the assumption is violated - for instance if $\Gamma_{pop}(\cdot, \pi)$ has multiple fixed points or none for some π - then there is no well-defined map from policies to stable

distributions, and the MFG-NE associated with π^* may not exist or be unique.[2] While in our experiments in Sec. 4.7.1 we have not verified $L_{pop,\mu} < 1$ explicitly, some of the experimental runs do converge to a stable, high-reward population distribution (e.g. all agents clustered in one corner, or agreed on one target); this suggests that when other runs fail to reach such an outcome, the cause is not a violation of Assumption 4.3.11 but rather sub-optimal policy convergence for another reason. Assumption 4.3.11 allows us to give the following operator that maps policies to their stable mean-field distributions.

Definition 4.3.12 (Stable population operator Γ^{∞}_{pop})**.** Given Assumption 4.3.11, 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 $\Gamma_{pop}(\cdot, \pi) : \Delta_{\mathcal{S}} \to \Delta_{\mathcal{S}}$.

4.3.2.2 Policy improvement operators

We now introduce the policy improvement operators, which we use for policy improvement in place of a direct best-response operator (see Sec. 2). As with the population evolution operators, these must also be Lipschitz, to ensure smoothness and hence convergence.

We first define the regularised Q-functions.

Definition 4.3.13 (Q_h and q_h functions)**.** We define, for any pair $(s, a) \in \mathcal{S} \times \mathcal{A}$:

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

and

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

We can now give the operator that maps policy-population pairs to Q-functions, i.e. it gives the Q-function that pertains when an agent uses a certain policy when

[2]Since $L_{pop,\mu} = K_s/2 + K_a/2 + K_\mu$, the contraction condition is helped substantially in our experimental games by the fact that P does not depend on $\hat{\mu}$ at all, giving $K_\mu = 0$; the remaining contributions $K_s/2, K_a/2$ depend on the chosen metrics on $\mathcal{S}$ and $\mathcal{A}$ and we do not verify them explicitly.

the population has a certain mean-field distribution. We are able to approximate this operator via Def. 4.4.1 below, which learns online from samples taken along a trajectory of the current policy.

Definition 4.3.14 (Γ_q operator)**.** The operator $\Gamma_q : \Pi \times \Delta_{\mathcal{S}} \to \mathcal{Q}$, which maps 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 now define the policy mirror ascent (PMA) operator for policy improvement. Agents update a policy with respect to a given Q-function by selecting, for each state, a probability distribution over their actions that maximises the combination of three terms (Def. 4.3.15): 1. the value of the given state with respect to the 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 PMA operator:

Definition 4.3.15 (Policy mirror ascent operator (Def. 3.5, [15]))**.** 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 4.3.8), the PMA update operator $\Gamma_\eta^{md} : \mathcal{Q} \times \Pi \to \Pi$ is defined as, $\forall s \in \mathcal{S}, \forall Q \in \mathcal{Q}, \forall \pi \in \Pi$

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

Γ_q and Γ_η^{md} are both known to be Lipschitz continuous [15].

We can now define the theoretical learning operator Γ_η, which is used in the fixed-point iterations to find the MFG-NE. Γ_η takes a PMA step to update a policy with respect to the Q-function of that policy when the population has a stable mean-field distribution arising from following that policy.

Definition 4.3.16 (Nested learning operator)**.** For a learning rate $\eta > 0$, $\Gamma_\eta : \Pi \to \Pi$ is defined as

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

The following lemma demonstrates that the fixed points of Γ_η are MFG-NE policies.

Lemma 4.3.17 (Fixed points of Γ_η are MFG-NE)**.** *For arbitrary $\eta > 0$, a pair (π^*, μ^*) is a MFG-NE if and only if $\pi^* = \Gamma_\eta(\pi^*)$ and $\mu^* = \Gamma_{pop}^{\infty}(\pi^*)$.*

We now recall that Γ_η is Lipschitz continuous (Yardim et al. [15] establishes the conditions under which it is contractive, which we omit here for simplicity).

Lemma 4.3.18 (Lipschitz continuity of Γ_η)**.** *For any $\eta > 0$, the operator $\Gamma_\eta : \Pi \to \Pi$ is Lipschitz with constant L_{Γ_η} on $(\Pi, ||\cdot||_1)$.*

4.3.2.3 Conditions when learning online from samples collected along a single run with N agents

General theoretical guarantees on online learning from a single run require mixing conditions on the samples along the path. As per Yardim et al. [15], we decompose these into the two assumptions below.

The first of these assumptions presumes that throughout training along the single system run, the regulariser h ensures that policies continue taking each action in each state with probability bounded away from zero. Yardim et al. [15] provides conditions on h that are sufficient for Assumption 4.3.19 to hold; in practice it is achieved for a large class of strongly concave h, including those realised as entropy regularisation, which is how we implement h for our experiments. If persistence of excitation fails - for example if a learner collapses onto a deterministic policy - the collected samples no longer cover the state-action space sufficiently for $\hat{Q}^i$ to concentrate around Q^*, and the sample-complexity guarantees we prove in subsequent sections no longer hold. Experimentally, we do not measure p_{inf} explicitly, and in fact set the entropy scaling coefficient to 0 for the runs reported in this chapter; that our algorithms still converge in this setting suggests that persistence of excitation is sufficient but not necessary for the behaviour we observe.

Assumption 4.3.19 (Persistence of excitation)**.** 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 $\Gamma_{\eta}^{md}(q,\pi)(a|s) \geq p_{inf}$, $\forall (s,a) \in \mathcal{S} \times \mathcal{A}$.

The next assumption complements Assumption 4.3.19 by requiring that, given a persistently excited policy, the induced state chain reaches every $s \in \mathcal{S}$ with a uniformly positive probability δ_{mix} within a finite time T_{mix}. It is automatically satisfied whenever the induced Markov chain on $\mathcal{S}$ is ergodic with strictly positive transition probabilities - a condition met by all the grid worlds we consider experimentally, where every square is reachable from every other under any policy that takes each cardinal direction with non-zero probability. If sufficient mixing fails - for example if the environment contains absorbing states or non-communicating sub-MDPs - the TD samples collected along the system run no longer give uniform convergence, and the sample-complexity guarantees we prove in subsequent sections no longer hold. Experimentally, we do not estimate T_{mix} or δ_{mix} explicitly, but the connectedness and finite size of our grid worlds appear to make sufficient mixing unproblematic in practice.

Assumption 4.3.20 (Sufficient mixing)**.** 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 $\{s_0^i\}_i \in \mathcal{S}^N$, there exist $T_{mix} > 0, \delta_{mix} > 0$ such that $\mathbb{P}(s_{T_{mix}}^j = s' | \{s_0^i\}_i) \geq \delta_{mix}$, $\forall s' \in \mathcal{S}, j \in \{1, \ldots, N\}$.

4.4 Learning with networked, decentralised agents

Roadmap We first introduce theoretical versions of our operators and algorithm (Secs. 4.4.1, 4.4.2), in order to show that our networked framework has sample guarantees bounded between those of the centralised- and independent-learning cases (Sec. 4.5). We show experimentally in Sec. 4.7.4.1 that these sample guarantees do not lead to practical learning times, whereas our novel incorporation of an

experience replay buffer (Sec. 4.6.1), 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 such cases we demonstrate experimentally that our modified networked algorithm still respects the theoretical guarantees: it can significantly outperform the independent algorithm, often performing similarly to the central-agent one (Sec. 4.7).

4.4.1 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, non-episodic system run (Lines 1-10 of Alg. 1). The two underlying learning 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 representative agent (the agent with arbitrary index $i = 1$) estimates the Q-function and computes an updated policy that is pushed to all the other agents.

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 (Line 8), i.e. to approximate the operator Γ_q (Def. 4.3.14, Sec. 4.3.2):

Definition 4.4.1 (Stochastic TD-learning operator, simplified from Def. 4.1 in Yardim et al. [15])**.** 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

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

Having estimated the Q-function of their current policy, agents use the Q-function to update this policy via the PMA operator from Def. 4.3.15 (Line 10).

The theoretical learning algorithm has three nested loops (see Lines 2, 4 and 5 of Alg. 1). The policy update is applied K times (Line 10). Before the policy

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 $\{s_0^i\}_{i=1}^N$
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 $(\forall i)$: $\hat{Q}_{m+1}^i = \tilde{F}_{\beta_m}^{\pi_k^i}(\hat{Q}_m^i, \zeta_{t-2}^i)$ (Def. 4.4.1)
9: $\quad$ **end for**
10: $\quad$ PMA step $\forall i : \pi_{k+1}^i = \Gamma_\eta^{md}(\hat{Q}_{M_{pg}}^i, \pi_k^i)$ (Def. 4.3.15)
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: $\quad\quad \forall i : J_t^i = i \cup \{j \in \mathcal{N} : (i,j) \in \mathcal{G}_t^{comm}\}$
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 $\{\pi_K^i\}_{i=1}^N$

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 (Line 8). Prior to the TD update in each of the M_{pg} loops, agents take M_{td} steps in the environment without updating (Line 6). 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 non-episodic system run [252]. However, we find in our experiments that we are able to essentially remove the inner M_{td} loops (Sec. 4.7.4).

4.4.2 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

in Lines 11-18 by the following method, which allows policies to spread through the population.[3] Coupled to their updated policy π^i_{k+1}, agents generate a scalar value σ^i_{k+1} (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^i_{k+1} at random leads to policies being exchanged at random (required in Thm. 4.5.2), whereas generating σ^i_{k+1} as an approximation of the return of π^i_{k+1} and then selecting the highest received value of σ^j_{k+1} leads to better performing policies spreading through the population. The latter is the approach we use for accelerating learning empirically (described in Sec. 4.6.2 on the practical running of our algorithm), albeit we use a softmax rather than a max function for selecting between received values. However, for generality in our theoretical results, we do not focus on a specific method for generating σ^i_{k+1}, such that it can be arbitrary for Thms. 4.5.2 and 4.5.10 below, and with few restrictions for Thms. 4.5.3 and 4.5.6.

Agents broadcast their policy π^i_{k+1} and the associated σ^i_{k+1} 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^j_{k+1} with a probability defined by a softmax function over the received values, and *adopts* the policy associated with this σ^i_{k+1}, i.e. it sets its own current π^i_{k+1} and σ^i_{k+1} to the ones it has selected (Lines 15, 16). This process repeats for C communication rounds, before the Q-function estimation steps begin again. After each communication 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

[3] As discussed in Sec. 4.2, our communication method is reminiscent of the use of fitness functions in distributed evolutionary algorithms [233, 237].

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^i_{k+1}, in subsequent sections.

Remark 4.4.2. Our networked architecture is effectively a generalisation of both the central-agent and independent settings (Algs. 2, 3, Yardim et al. [15]). The independent setting is the special case where there is no communication, i.e. $C = 0$ - this serves as an implicit ablation of our communication scheme. The central-agent setting is the special case when σ^i_{k+1} 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 if we set $\tau_k \to 0$ (such that the softmax becomes a max function), and assume that the communication network becomes jointly connected repeatedly during each set of communication rounds, results on max-consensus tell us that the central learner's policy will always be 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 [253, 254].

Remark 4.4.3. In practice, when referring to a central-agent version of the networked Alg. 1, for simplicity we assume there is no networked communication and instead that the updated policy π^1_{k+1} of the representative learner $i = 1$ is pushed to all agents after Line 10, as in Alg. 2 of [15].

4.5 Theoretical results

4.5.1 Introduction

In this section 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 (Sec. 4.5.2) or not (Sec. 4.5.3). We then provide the order of the difference in these bounds in the non-random case in terms of the network structure and number of communication rounds (Sec. 4.5.4). We finally give a policy-update stability guarantee, which applies in all scenarios (Sec. 4.5.5).

All expectations $\mathbb{E}[\,\cdot\,]$ in this section and its proofs are taken jointly over:

- the initial joint state $\{s_0^i\}_{i=1}^N$;
- the stochastic transitions and action samples drawn during the M_{td} and M_{pg} inner loops of Alg. 1;
- the stochasticity of the PMA updates inherited from the sampled $\hat{Q}^i$;
- where relevant (Thm. 4.5.2 only), the random adoptions performed under the softmax in Line 15.

The communication network $\mathcal{G}_t^{comm}$ is supplied as an exogenous input at each round rather than as a random variable in $\mathbb{E}[\,\cdot\,]$, and may or may not depend on the agents' states. In our experiments it is defined by a broadcast radius applied to agent positions, in which case it is deterministic conditional on the joint state and inherits randomness only implicitly through the stochastic state evolution, but the theory itself accepts any sequence of graphs - deterministic or otherwise - satisfying the relevant structural conditions. In Thms. 4.5.2-4.5.3 the sequence is left arbitrary and its only role is to determine which policies are exchanged in Line 15, while Thm. 4.5.6 additionally fixes $\mathcal{G}_t^{comm}$ as static and connected with constant diameter $d_{\mathcal{G}}$, and Rems. 4.5.8-4.5.9 discuss loosening these conditions to dynamic / jointly connected sequences. Our bounds therefore hold for any realisation of $\mathcal{G}_t^{comm}$ consistent with the relevant theorem's structural conditions; explicit modelling of stochastic graph generation (e.g. link drops, Erdős-Rényi sampling) is left to future work.

4.5.2 Networked learning with random policy adoption

We begin by recalling the sample guarantees of the architecture where agents learn entirely independently [15].

Lemma 4.5.1 (Independent learning, from Thm. 4.5, Yardim et al. [15])**.** *For p_{inf} and δ_{mix} defined in Assumptions 4.3.19 and 4.3.20 respectively, define $t_0 := \frac{16(1+\gamma)^2}{((1-\gamma)\delta_{mix}p_{inf})^2}$. Assume that Assumptions 4.3.8, 4.3.11, 4.3.19 and 4.3.20 hold, and that π^* is the unique MFG-NE policy. For L_{Γ_η} defined in Lem. 4.3.18, 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. 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 $i \in \{1, \ldots, N\}$,

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

We first give a result for the trivial situation of random policy adoption to provide an intuition that networked communication preserves the sample guarantees of independent learning, before showing the conditions under which the latter can be outperformed.

Theorem 4.5.2 (Networked learning with random policy adoption)**.** *For p_{inf} and δ_{mix} defined in Assumptions 4.3.19 and 4.3.20 respectively, define $t_0 := \frac{16(1+\gamma)^2}{((1-\gamma)\delta_{mix}p_{inf})^2}$. Assume that Assumptions 4.3.8, 4.3.11, 4.3.19 and 4.3.20 hold, and that π^* is the unique MFG-NE policy. For L_{Γ_η} defined in Lem. 4.3.18, 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. Let us set $C > 0$ and $\tau_k \to \infty$. 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. 4.5.1, i.e. the output satisfies, for all agents $i \in \{1, \ldots, N\}$,*

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

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 for an arbitrary $C > 0$ rounds, which does not affect the random output of the algorithm, such that the random output satisfies the same expectation as if $C = 0$. □

4.5.3 Networked learning with non-random policy adoption

If instead 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 algorithm is bounded between that of the centralised and independent algorithms, formalised in Thm. 4.5.3 below.

More formally, in Thm. 4.5.3, we 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 arbitrary index i (so as not to bias the spread of any particular policy). Let the random output of this algorithm be denoted as $\{\pi_K^{i,net}\}_i$. Also consider an independent-learning version of the algorithm (i.e. with the same parameters except $C = 0$) and denote its random output $\{\pi_K^{i,ind}\}_i$; and a central-agent version of the algorithm with the same parameters (see Rem. 4.4.3) and denote its random output as π_K^{cent}. Then for all agents $i \in \{1, \dots, N\}$, the random outputs of these algorithms $\{\pi_K^{i,net}\}_i$, $\{\pi_K^{i,ind}\}_i$ and π_K^{cent} satisfy the following relations, where ub_{net}, ub_{ind} and ub_{cent} are respective upper bounds for each case:

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

Thm. 4.5.3 compares these upper bounds:

Theorem 4.5.3 (Networked learning with non-random policy adoption)**.** *Assume that Assumptions 4.3.8, 4.3.11, 4.3.19 and 4.3.20 hold, and that Alg. 1 is run with learning rates and constants as defined in Thm. 4.5.2, except now let us set $\tau_k \in \mathbb{R}_{>0}$. Then for all agents $i \in \{1, \dots, N\}$, the random outputs $\{\pi_K^{i,net}\}_i$, $\{\pi_K^{i,ind}\}_i$ and π_K^{cent} satisfy*

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

Proof. We build off the proof of our Lem. 4.5.1, given in Thm. D.9 of Yardim et al. [15]. There 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 $k \in \{0, 1, \dots, K\}$ define the policy divergence as the random variable $\Delta_k := \sum_{i=1}^{N} ||\pi_k^i - \bar{\pi}_k||_1$. We can say that $\Delta_{k,cent} = 0 \ \forall k$ is the divergence in the central-agent 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 $\Pr\left(\text{adopted}^i = \sigma_{k+1}^j\right) = \frac{\exp\left(\sigma_{k+1}^j/\tau_k\right)}{\sum_{x=1}^{[J_t^i]} \exp\left(\sigma_{k+1}^x/\tau_k\right)}$ (as in Line 15 of Alg. 1) is higher for some $j \in J_t^i$ 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 $\pi_{k+1}^i = \pi_{k+1}^j$ for some $i, j \in \{1, \dots, N\}$. As such, $\Delta_{k+1,cent} \leq \mathbb{E}\left[\Delta_{k+1,C}\right] \leq \mathbb{E}\left[\Delta_{k+1,0}\right]$, 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. 4.5.1 given in Thm. D.9 of Yardim et al. [15] ends with, for constants χ and ξ,

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

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

Thus, for all agents $i \in \{1, \dots, N\}$, the random outputs $\{\pi_K^{i,ind}\}_i$, $\{\pi_K^{i,net}\}_i$ and π_K^{cent} satisfy respectively:

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

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

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

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.

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

□

We recall the following lemma from Yardim et al. [15]; we use it in Rem. 4.5.5 to aid intuitive understanding of the result just given in Thm. 4.5.3.

Lemma 4.5.4 (Conditional TD learning from a single continuous run of the empirical distribution of N agents, from Thm. 4.2, Yardim et al. [15])**.** *Define* $t_0 := \frac{16(1+\gamma)^2}{((1-\gamma)\delta_{mix}p_{inf})^2}$. *Assume that Assumption 4.3.20 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* $\hat{Q}^i_{M_{pg}}$ *of Lines 3-9 satisfies*

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

Remark 4.5.5. It may help to see that our Thm. 4.5.3 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 central-agent cases respectively. In Lem. 4.5.4, 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 central-agent case requires.

4.5.4 Effect of amount of communication on relative architecture performance

We now provide a result that shows how the sample guarantees of our networked architecture vary along the spectrum between those of the central-agent and independent cases depending on the amount of communication that occurs.

Theorem 4.5.6 (Relation between communication network structure and order of difference between the architectures' bounds)**.** *In addition to the assumptions in Thm. 4.5.3, now also assume that the communication network $\mathcal{G}_t^{comm}$ remains static and connected during the C communication rounds. Assume also the diameter $d_{\mathcal{G}}$ of the network is equal for all k. Let us set $\tau_k \; \forall k$ as a small positive constant chosen to be sufficiently close to zero that the softmax essentially becomes a max function. Then, for the tight bound big Theta (Θ), we can say that the difference in the upper bounds ub_{net}, ub_{ind} and ub_{cent} from Thm. 4.5.3 depends on C and the network diameter $d_{\mathcal{G}}$ as follows (where the '$\approx$' relation comes from the approximate spread of policies through the network as explained in the proof):*

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

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

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

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. From the proof of Thm. 4.5.3 we have:

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

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

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

Say that $\sigma_{k+1}^{\max}$ is the highest σ^i value in the population before the communication rounds at $k+1$. With a static, connected network and τ_k close to 0 for all k, max-consensus will always be reached on $\sigma_{k+1}^{\max}$ after $C = d_{\mathcal{G}}$ communication rounds, such that $\Delta_{k,cent} = \Delta_{k,d_{\mathcal{G}}} = 0$ [255]. The convergence rate of the max-consensus algorithm is $\frac{1}{d_{\mathcal{G}}}$ [255], 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:

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

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:

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

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

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

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 4.5.7. If it is always σ_{k+1}^1 and π_{k+1}^1 that is adopted by the whole population (i.e. $i = 1$), then this is exactly the same as the central-agent case. If the σ_{k+1}^j and π_{k+1}^j that gets adopted has different j for each k, then this is akin to a version of the central-agent setting where the index of the representative learning agent may differ for each k.

Remark 4.5.8. Thm. 4.5.6 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 instead only assume that the network repeatedly becomes jointly connected during each set of communication rounds, we can replace $d_{\mathcal{G}}$ in the results in Thm. 4.5.6 with $d_{avg} \cdot \omega$, namely the average diameter of the union of each jointly connected collection of graphs in the sequence, multiplied by the average number ω of graphs in each jointly connected collection. As noted in Rem. 4.4.2, max-consensus is reached if C is large enough that the number of sequential jointly connected collections of graphs occurring within C is equal to the largest diameter of the union of any collection. This is equivalent to the central-agent case; there is no added benefit to higher values of C than this.

Remark 4.5.9. Thm. 4.5.6 sets τ_k as a small positive value close to 0 such that the softmax function becomes a max function. If we instead set $\tau_k \in \mathbb{R}_{>0}$ not close to 0 such that the softmax function is less peaked, 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_{k+1}^j. 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.

4.5.5 Stability guarantee

For completeness, we finally give a stability guarantee that follows from the earlier theorems.

Theorem 4.5.10 (Policy-update stability guarantee)**.** *Let Alg. 1 run as per Thm. 4.5.2 or Thms. 4.5.3/4.5.6, 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 a stability guarantee 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. 4.5.2, 4.5.3 and 4.5.6 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. □

4.6 Practical modifications to theoretical algorithms for empirical use

The theoretical analysis in Sec. 4.5 requires algorithmic hyperparameters (see Thm. 4.5.2) that render convergence impractically slow in all of the centralised, independent and networked cases. In particular, the values of δ_{mix} and p_{inf} give rise to very large t_0, causing very small learning rates $\{\beta_m\}_{m\in\{0,\ldots,M_{pg}-1\}}$, and necessitating very large values for M_{td} and M_{pg}. Indeed Yardim et al. [15] do not provide empirical demonstrations of their algorithms for the centralised and

independent cases. Our experiments in Sec. 4.7.4.1 show that with these algorithms, none of the architectures appear to improve their returns at all without extremely high numbers of inner loops that would take impractically long to run on standard computers, taking many days or even many weeks.

For convergence of the algorithms in practical time, we seek to drastically increase $\{\beta_m\}_m$ and reduce M_{td} and M_{pg}. We found empirically that the two algorithmic enhancements below helped achieve feasible learning times with significantly reduced numbers of loops. The first involves recycling transitions using a buffer, and the second gives a principled way of selecting σ^i_{k+1} in Line 11 in Alg. 1.

4.6.1 Algorithm acceleration by use of experience-replay buffer

We modify our Alg. 1 (and accordingly the algorithms of the two non-networked architectures) as follows, shown in *blue* in Alg. 2. 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 [256–258]. 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 $\{\beta_m\}_m$, 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 continued, non-episodic system run, which may explain why we are able to achieve adequate stable learning even when reducing the number of M_{td} waiting steps within each M_{pg} loop (Sec. 4.7.4).

Our replay buffer allows the first practical demonstrations of all three architectures for learning from a single continued system run - all of our experiments after those in Sec. 4.7.4.1 use the buffer, with which learning can now occur. The

Algorithm 2 Networked learning with experience replay and performance-related generation of σ^i_{k+1}

Require: loop parameters K, M_{pg}, M_{td}, C, L, E, learning parameters η, β, λ, γ, $\{\tau_k\}_{k\in\{0,\dots,K-1\}}$
Require: initial states $\{s^i_0\}_{i=1}^N$
Set $\pi^i_0 = \pi_{\max}, \forall i$ and $t \leftarrow 0$
for $k = 0, \dots, K-1$ **do**
  $\forall s, a, i : \hat{Q}^i_0(s,a) = Q_{\max}$
  $\forall i$: Empty i's buffer
  for $m = 0, \dots, M_{pg} - 1$ **do**
    for M_{td} iterations **do**
      Take step $\forall i : a^i_t \sim \pi^i_k(\cdot|s^i_t), r^i_t = R(s^i_t, a^i_t, \hat{\mu}_t), s^i_{t+1} \sim P(\cdot|s^i_t, a^i_t, \hat{\mu}_t)$; $t \leftarrow t+1$
    end for
    $\forall i$: Add ζ^i_{t-2} to i's buffer
  end for
  for $l = 0, \dots, L-1$ **do**
    $\forall i$: Shuffle buffer
    for transition ζ^i_b in i's buffer $(\forall i)$ **do**
      Compute TD update $(\forall i)$: $\hat{Q}^i_{m+1} = \tilde{F}^{\pi^i_k}_\beta(\hat{Q}^i_m, \zeta^i_{t-2})$ (see Def. 4.4.1)
    end for
  end for
  PMA step $\forall i : \pi^i_{k+1} = \Gamma^{md}_\eta(\hat{Q}^i_{M_{pg}}, \pi^i_k)$ (see Def. 4.3.15)
  $\forall i : \sigma^i_{k+1} \leftarrow 0$
  for $e = 0, \dots, E-1$ evaluation steps **do**
    Take step $\forall i : a^i_t \sim \pi^i_{k+1}(\cdot|s^i_t), r^i_t = R(s^i_t, a^i_t, \hat{\mu}_t), s^i_{t+1} \sim P(\cdot|s^i_t, a^i_t, \hat{\mu}_t)$
    $\forall i : \sigma^i_{k+1} \leftarrow \sigma^i_{k+1} + \gamma^e(r^i_t + h(\pi^i_{k+1}(s^i_t)))$
    $t \leftarrow t+1$
  end for
  for C rounds **do**
    $\forall i$: Broadcast $\sigma^i_{k+1}, \pi^i_{k+1}$
    $\forall i : J^i_t = i \cup \{j \in \mathcal{N} : (i,j) \in \mathcal{G}^{comm}_t\}$
    $\forall i$: Select adopted$^i \sim \Pr\left(\text{adopted}^i = j\right) = \frac{\exp(\sigma^j_{k+1}/\tau_k)}{\sum_{x\in J^i_t}\exp(\sigma^x_{k+1}/\tau_k)}\ \forall j \in J^i_t$
    $\forall i : \sigma^i_{k+1} \leftarrow \sigma^{\text{adopted}^i}_{k+1}, \pi^i_{k+1} \leftarrow \pi^{\text{adopted}^i}_{k+1}$
    Take step $\forall i : a^i_t \sim \pi^i_{k+1}(\cdot|s^i_t), r^i_t = R(s^i_t, a^i_t, \hat{\mu}_t), s^i_{t+1} \sim P(\cdot|s^i_t, a^i_t, \hat{\mu}_t)$; $t \leftarrow t+1$
  end for
end for
return policies $\{\pi^i_K\}_{i=1}^N$

intuition behind the better learning efficiency resulting from the buffer 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.

Remark 4.6.1. Our introduction of the replay buffer means that the *specific* sample guarantees given in our theoretical results no longer apply. This is because these assume that learning is conducted from a single stream of samples that are discarded straight after their first and only use, and these results do not account for temporary storage and shuffled reuse of samples. We do not directly update the sample guarantees in light of this algorithmic modification, and leave such additional proofs to future work. However we emphasise here that *we expect the ranking of the performances of the architectures to be preserved*, i.e. the central-agent architecture still learns as fast or faster than the networked one, which in turn learns as fast or faster than the independent one. This is because, although the use of samples in learning has changed, the underlying machinery that drives the difference in performance between the architectures has not. The independent architecture will still have worse sample guarantees than the central-agent one due to bias caused by policy divergence, and our networked communication and adoption can still reduce this divergence depending on network structure and the number of communication rounds, as before. In summary, *the conceptual gap between our theoretical and empirical algorithms is minimal, and our theoretical results still give heuristic insight to explain our experimental results that use the buffer*, which show networked populations outperforming independent ones while underperforming or performing similarly to the central-agent populations, as predicted by our original theory.

We leave β 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.

4.6.2 Generation of σ^i_{k+1}

Reducing the number of loops in the hope of achieving practical convergence times can lead to 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}. This problem will increase with the size of the state and action spaces. 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 markedly outperform the independent case 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. 4.5.

We do so via the steps added in *orange* in Alg. 2, which replace Line 11 in Alg. 1: for $\boldsymbol{\pi}_{k+1} := (\pi^1_{k+1}, \dots, \pi^N_{k+1})$, we set σ^i_{k+1} to a finite-step estimation $\widehat{\Psi}^i_{h,k+1}(\boldsymbol{\pi}_{k+1}, \upsilon_0; E)$ of the discounted return $\Psi^i_{h,k+1}(\boldsymbol{\pi}_{k+1}, \upsilon_0)$ (Def. 4.3.2). The estimation is given by, $\forall i, j \in \{1, \dots, N\}$

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

This is calculated by tracking each agent's discounted return for E evaluation steps (Lines 19-23).

Generating σ^i_{k+1} in this way means policies that are more likely to spread through the network are those estimated to receive a higher return in reality, despite possibly being generated from poorly estimated Q-functions; this biases the population towards faster learning. Naturally the quality of the finite-step 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.

4.7 Experiments

Our technical contribution of the replay buffer to MFG algorithms for online learning from non-episodic system runs allows us also to contribute the first empirical demonstrations of learning with these algorithms, not just in the networked case but also in the central-agent and independent cases. The latter two serve as baselines to show the advantages of the networked architecture. 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. 4.3.15, and the JAX framework to accelerate and vectorise some elements of our code.

4.7.1 Games

We follow the gold standard in prior works on stationary MFGs regarding the types of game demonstrated: we focus on grid-world environments where agents can move in the four cardinal directions or remain in place [20, 99, 110, 126, 153, 200]. While this type of experiment is characteristic of similar MFG works, we recognise that these are simple games. They nevertheless serve as useful preliminary demonstrations of the validity of our algorithms and the considerations necessary for achieving practical learning; we leave experiments in more complex environments to future work. Moreover, grid-world environments naturally reflect the deployed, spatial applications in which we are interested in our setting, where agents learn online and communicate with neighbours on a network (which is likely to be defined spatially, though is not restricted to such a case).

We conduct numerical tests with two tasks (defined by the agents' reward functions), chosen for being particularly amenable to intuitive 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.

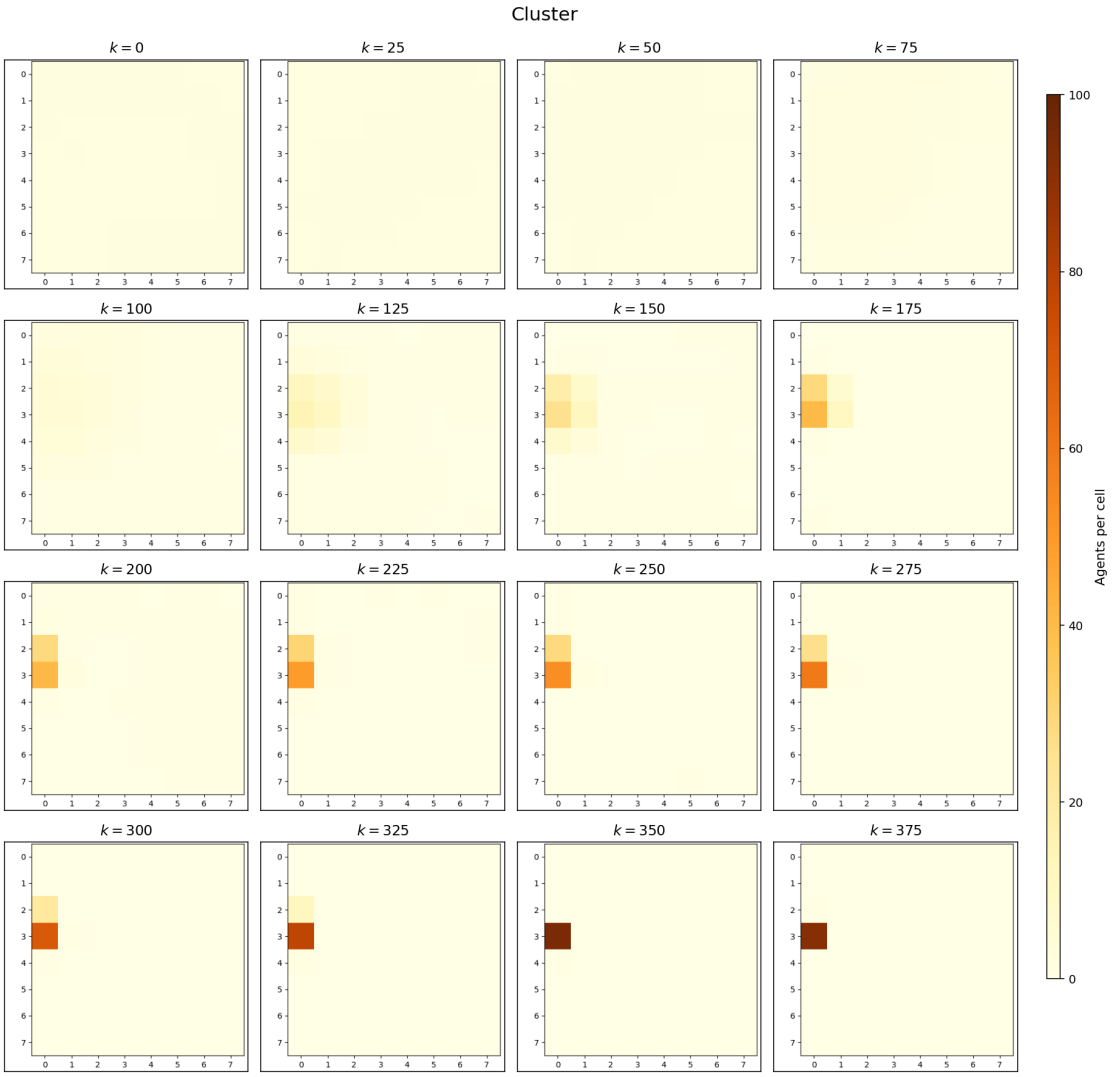


Figure 4.2: An example evolution of the population's distribution in the 'cluster' game. Note that it is more common for the population to gather in a corner cell, as discussed in the text.

Cluster. This is the inverse of the 'exploration' game in [20], 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.

Per-state reward $\log(\hat{\mu}(s))$ is concave in $\hat{\mu}(s)$ (normalised so that $\hat{\mu} \in [1/N, 1]$ maps to $[0, 1]$), so in this game any Pareto-dominant MFG-NE has all agents at a single state. There are $|\mathcal{S}|$ such MFG-NE, one per choice of clustering state; each

gives per-agent reward 1 (the maximum), so they are all Pareto-optimal. They jointly Pareto-dominate a continuum of sub-optimal NE in which the mean-field distribution is stationary on a k-subset of states with mass fractions $(\alpha_1, \ldots, \alpha_k)$, $\sum_i \alpha_i = 1$, $k \geq 2$ - individual agents need not be stationary (any joint policy preserving the mass fractions is admissible, including ones in which agents swap between populated states), but moving to a non-populated state gives normalised reward 0 (the moving agent's own mass gives $\hat{\mu} = 1/N$ at the new state), so confining agents to the populated support is a best response. The uniform-wandering NE is the limit of this family at $k = |\mathcal{S}|$, with per-agent normalised reward $1 - \log(|\mathcal{S}|)/\log(N)$. The location of the Pareto-dominant clustering state is unspecified by the reward and is selected stochastically by the early dynamics. In practice we usually find agents cluster at one of the four corners (though not in the case of the example trajectory in Fig. 4.2) despite all states being Pareto-equivalent: three of the five available actions leave an agent in any corner cell (the 'stay' action plus the two cardinal moves that bounce off a wall), two leave them in any non-corner edge cell, and only 'stay' keeps them at any interior cell - so exploratory policies drift the population toward the corners, and once a cluster begins to form there the $\log \hat{\mu}$ reward reinforces this bias.

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 $R(s_t^i, a_t^i, \hat{\mu}_t) = r_{targ}(r_{collab}(\hat{\mu}_t(s_t^i)))$, where

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

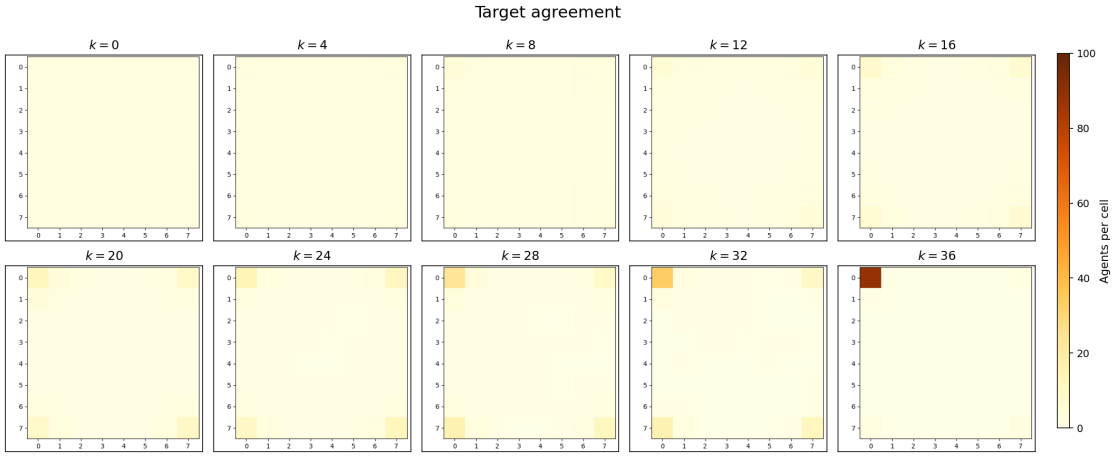


Figure 4.3: An example evolution of the population's distribution in the 'target agreement' game.

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

In this game the reward function jointly penalises being away from a target (r_{targ}) and being at a target without a sufficient fraction of the population (r_{collab}), so any Pareto-dominant strategy must concentrate the population on one of the $|\Phi|$ specified locations, giving $|\Phi| = 4$ Pareto-optimal MFG-NE. These jointly Pareto-dominate a continuum of sub-optimal NE in which the mean-field distribution is partitioned across two or more targets with insufficient mass at each (paying the r_{collab} penalty), parametrised by the chosen subset of targets and the mass split among them; as in the cluster game, individual agents may swap between populated targets as long as the mass split is preserved. Networked communication is expected to break this symmetry faster than the independent baseline by spreading whichever policy is empirically performing best - typically a policy that directs agents toward the target with the largest accumulated population fraction.

Both of these two games are coordination games, where selfish agents can increase their individual rewards by following the same strategy as others and therefore inherently have an incentive to communicate policies. Moreover, they require more sophisticated solutions than the dispersal/exploration games often considered in similar MFG works [20, 99, 153], where a trivial starting policy that

encourages agents to move across the grid at random may already be close to the equilibrium policy.

4.7.2 Experimental metrics

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

4.7.2.1 Exploitability

Works on MFGs most commonly use the *exploitability* metric to evaluate how close a given policy π is to a NE policy π^* [19, 20, 100, 153, 200, 259]. The metric usually assumes that all agents are following the same policy π, and quantifies how much an agent could benefit by deviating from π, by measuring the difference between the return V_h (Def. 4.3.4) gained by π and that gained by a policy that best responds to the population distribution generated by π. Let us denote by μ^π the distribution generated when π is the policy followed by all of the population aside from the deviating agent; then the exploitability of policy π is defined as follows:

Definition 4.7.1 (Exploitability of π)**.** The exploitability $\mathcal{E}_{\text{expl}}$ of policy π is given by:

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

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 $\pi = \pi^*$. Thus lower exploitability is considered better.

Since we do not have access to the exact best response policy $\arg\max_{\pi'} V_h(\pi', \mu^\pi)$ as in some related works [20, 153], we instead approximate the exploitability metric, similarly to [107], 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 'deviation' k loops of policy improvement. To approximate the expectations in Def. 4.7.1, 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. Due to the expensive computations required for this metric, we evaluate it only on alternate k iterations of the actual system evolution (for our experiments without the experience replay buffer in Sec. 4.7.4.1, we evaluate only every 20 k).

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, i.e. when $\pi_k^i \neq \pi_k^j$ for some $i, j \in \{1, \ldots, N\}$. 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.

The exploitability metric has a number of limitations in our setting. In coordination games (the setting for our tasks), agents benefit by following the same behaviour as others, and so a deviating agent generally stands to gain less from a 'best-responding' policy than it might in the non-coordination games on which many other works focus. For example, the return of a best-responding agent in the 'cluster' game still depends on the extent to which other agents coordinate on where to cluster, meaning it cannot significantly increase its return by deviating from a badly clustering policy. This means that the downward trajectory of the exploitability metric is less clear in our plots than in other works.

Moreover, our approximation of exploitability takes place via policy improvement steps (as in the main algorithm) for an independent, deviating agent while the policies of the rest of the population are frozen. As such, the quality of our approximation is limited by the number of policy-improvement/expectation-estimation rounds, which must be restricted for the sake of the running speed of the experiments. Furthermore, since one of the findings of our paper is that independent-learning

agents increase their returns significantly slower (if at all) than networked or central-agent populations, it is arguably unsurprising that approximating the best response by an independently deviating agent sometimes gives an unclear and noisy metric.

Given the limitations presented by approximating exploitability, we also provide the second metric to indicate the progress of learning.

4.7.2.2 Average discounted return

We record the average finite-step discounted return of the agents' policies π_k^i during the M_{pg} steps of each outer k loop. This allows us to observe that settings that converge to similar exploitability values may not have similar average agent returns, suggesting that some algorithms are better than others not just at reaching equilibria, but also at finding 'preferable' (i.e. Pareto-dominant) equilibria (when the assumption of a unique MFG-NE is removed by reducing regularisation; see Sec. 4.7.4) - cf. Li et al. [61], Graber [160]. See, for example, Fig. 4.12, where the networked agents converge to similar exploitability as the independent agents, but receive higher average reward.

4.7.2.3 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 $\bar{\pi} = \pi^1$. Many of our theoretical results and proofs relate to the policy divergence, and in Sec. 4.5 we show extensively how the comparatively worsening sample complexities between the centralised, networked and independent cases are the result of their range of policy divergences. We therefore include this metric to show how this relationship affects learning in practice.

Furthermore, the theoretical guarantees assume that the population is trying to learn the unique equilibrium policy π^*, with the implication that all agents should end up with this identical policy, regardless of the learning architecture (Sec. 4.5). However, we find in practice that populations may be converging (in terms of exploitability/return) while having non-diminishing policy divergence, particularly in the independent setting. We therefore also include this metric to indicate the difference between theoretical and empirical convergence.

4.7.3 Hyperparameters

See Table 4.1 for our hyperparameter choices. In general, we seek to show that our networked algorithm is robust to 'poor' choices of hyperparameters, such as low numbers of iterations, as may be required when aiming for practical convergence times in complex real-world problems. By contrast, the independent algorithm exhibits minimal learning 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.

We can broadly 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 $(\beta, \eta, \lambda, \tau, \gamma)$.

Table 4.1: Hyperparameters

Hyper-param.	Value	Comment
Gridsize	8x8 / 16x16	Most experiments are run on the smaller grid, while Figs. 4.12 and 4.13 demonstrate 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}, 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 [105, 108, 126, 142, 201, 229–232]. In experiments on 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.

Continued on next page

Table 4.1: Hyperparameters (continued)

Hyper-param.	Value	Comment
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 {1,5,10}. We choose 1 to show the convergence benefits brought by even a single communication round, even in networks that may have limited connectivity.
L	100	As with M_{pg}, we 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 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 adequate 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. 4.14 and 4.15); this yields learning, but does not perform as well as if we step 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 inverse annealing process may lead to better results.

4.7.4 Results and discussion

We first provide results for learning without a replay buffer as in the theoretical algorithms. We then give the rest of our experiments with the replay buffer, beginning with our standard experimental setting and then robustness studies and

ablations (note that the independent setting serves as an implicit ablation of our communication scheme). We summarise findings in the body of each sub-section, while the specific results are discussed fully in each figure's caption. In each plot the decimals refer to each agent's broadcast radius as a fraction of the maximum possible distance in the grid (i.e. the diagonal). Note that the networked population with the largest radius is always fully connected.

We pre-empt possible concerns regarding the wide confidence intervals in many of our plots by saying that many works with similar experiments do not report error bars at all, and if they do they usually only give 1-sigma intervals, whereas we give 2-sigma [20, 99, 110, 126, 200]. Moreover, the central-agent architecture usually has similar or higher variance compared to the networked agents in the plots, indicating that the wide confidence intervals are not an issue introduced by our communication algorithm; they are instead likely to be due to poor estimation of the Q-function when using the small numbers of loops required for practical runtimes. The independent agents have very low variance, but this is because they hardly appear to increase their returns at all in most cases.

We also give the following remark regarding the exploitability metric in some of our experimental plots, relating to the issues with this metric in coordination games, as discussed in Sec. 4.7.2.1:

Remark 4.7.2. The reward structure of our coordination games is such that exploitability sometimes increases from its initial value before it decreases down to 0 (e.g. Fig. 4.6). 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. In fact,

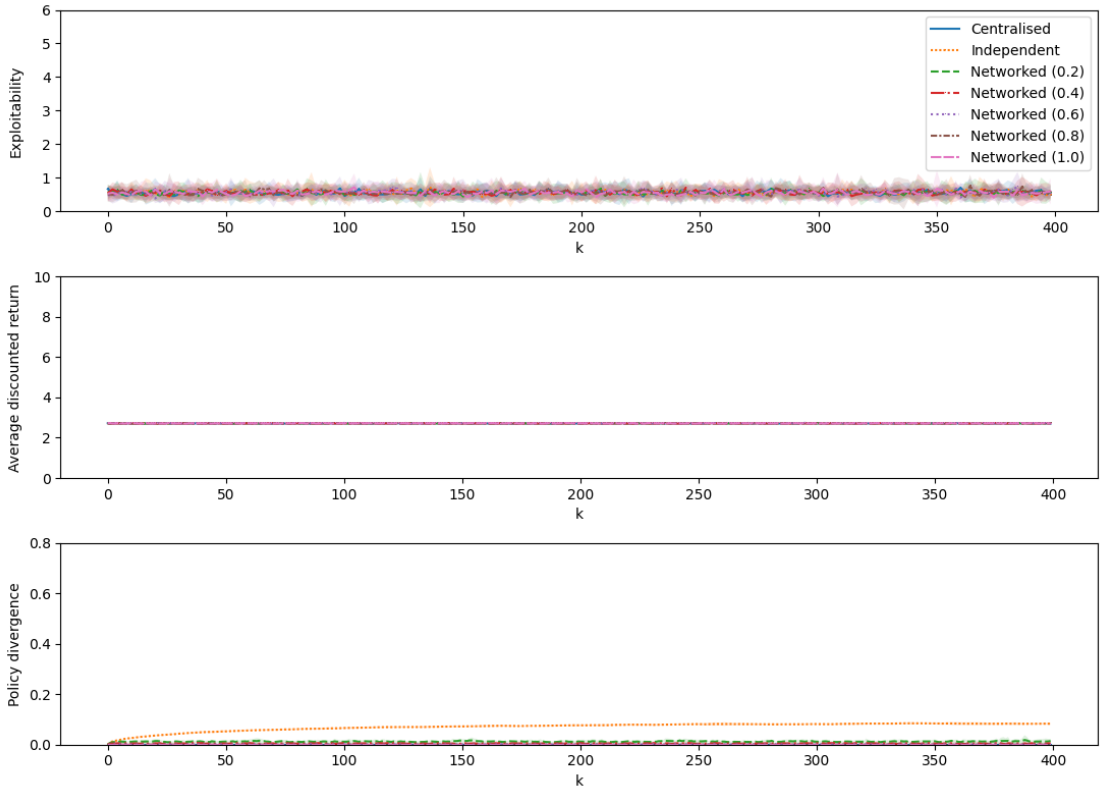


Figure 4.4: 'Cluster' game without our experience replay buffer. There is no noticeable improvement in any of the agents' returns, i.e. no noticeable learning, even after $K = 400$ iterations.

in such cases we can often see that the independent agents are hardly learning at all, with the independent agents' average return not increasing and the exploitability staying level rather than ultimately decreasing (see, for example, Figs. 4.6, 4.8, 4.10 and 4.12).

4.7.4.1 Learning with no experience replay buffer

Figs. 4.4 and 4.5 illustrate the importance of our incorporation of the experience replay buffer. Without it, as in the original theoretical 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. In these experiments without a replay buffer 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 Table 4.1). The theoretical results in fact require that these

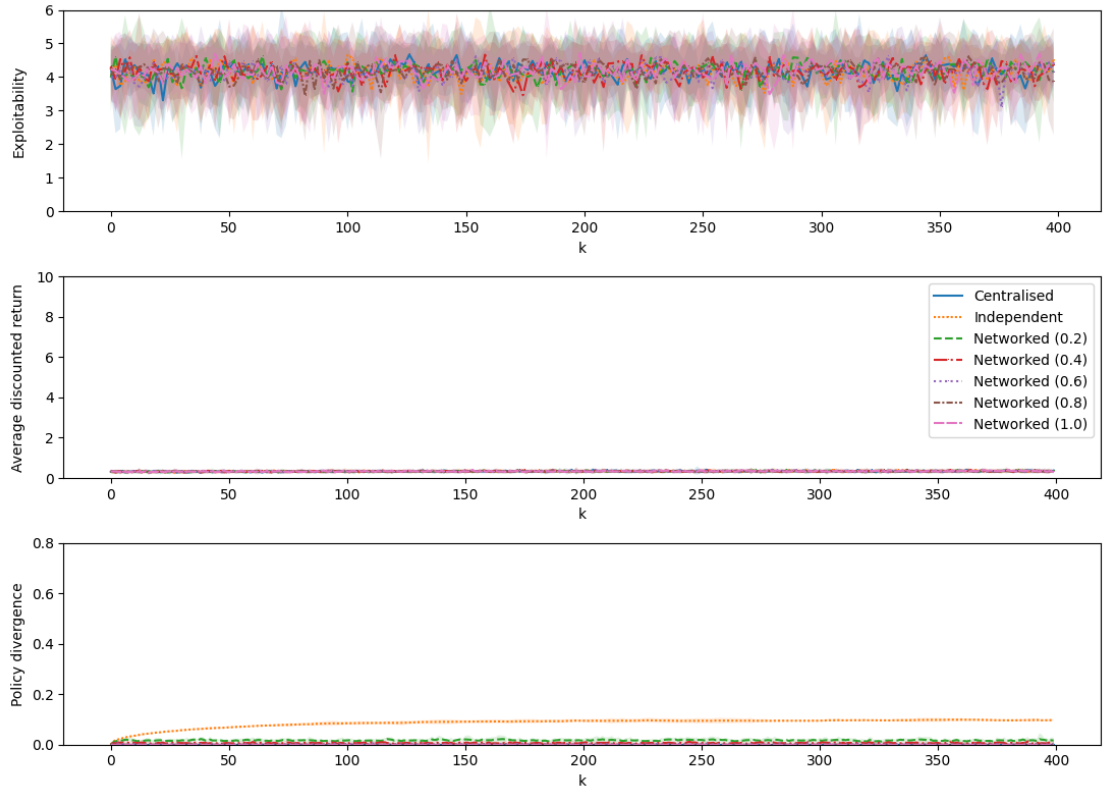


Figure 4.5: 'Target agreement' game without our experience replay buffer. There is no noticeable improvement in any of the agents' returns, i.e. no noticeable learning, even after $K = 400$ iterations.

values are many orders of magnitude higher. While setting them as such might mean that empirically we do see some learning occurring within $K < 400$, such experiments would take impractically long to run on standard computers, taking many days or even many weeks.

These experiments are run for 5 trials rather than 10 as in all other cases, and with exploitability evaluated every 20 k instead of every 2 k for computational efficiency.

The remainder of our experiments all include our replay buffer, and therefore do permit learning in practical time, albeit at different rates for the different architectures.

4.7.4.2 Standard experimental setting with replay buffer

Even with only a single communication round in each of the K loops, networked agents learn faster and reach higher returns than independent agents, which still

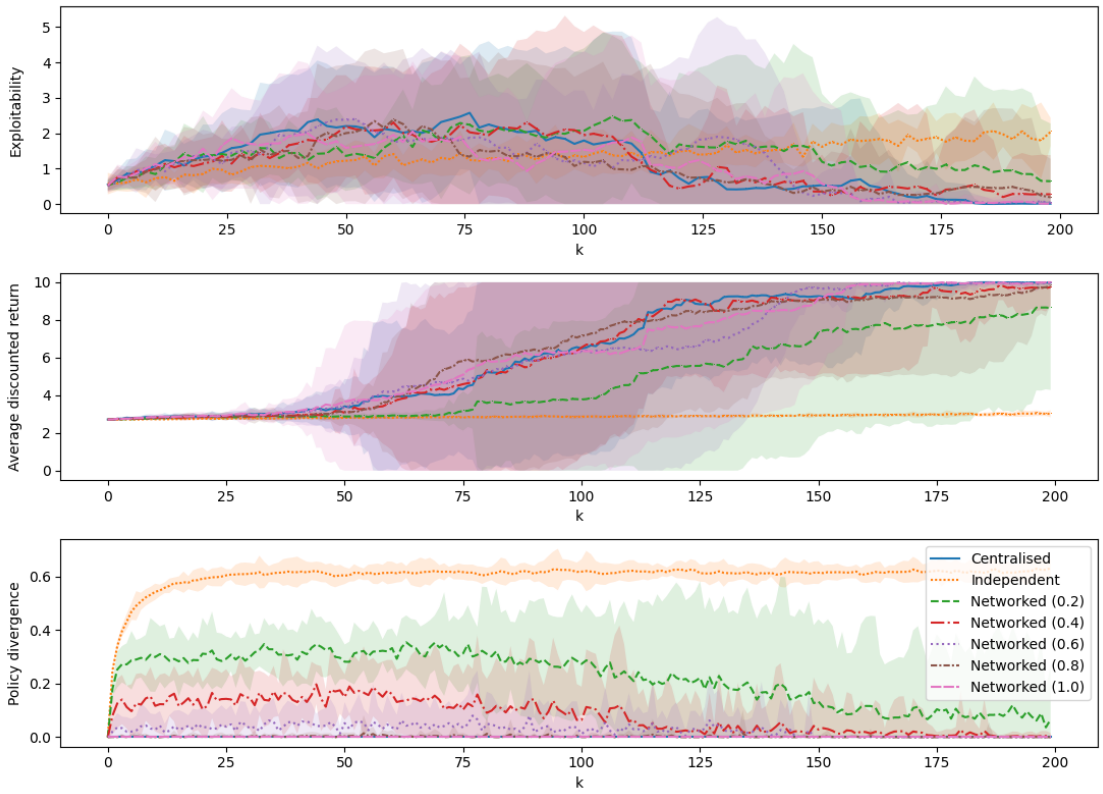


Figure 4.6: 'Cluster' game. Even with only a single communication round, our networked architecture significantly outperforms the independent case, which hardly appears to be learning at all. All broadcast radii except the smallest (0.2, green) have similar mean exploitability and return to the centralised case.

hardly appear to learn at all. Moreover networked agents appear to match the central-agent population in the 'cluster' game (Fig. 4.6). Our experiments show that our practical algorithmic enhancements enable convergence within a practical number of iterations even when we remove a number of the assumptions required for the theoretical algorithms:

- We reduce M_{pg} by many orders of magnitude from its theoretically required value (see Sec. 4.6), while still converging within a reasonable K. We keep the learning rate β fixed, removing the annealing scheme for $\{\beta_m\}_{m\in\{0,\dots,M_{pg}-1\}}$ required in the theorems, and use a much higher value.

- In our experiments we do not ensure that the communication network $\mathcal{G}_t^{comm}$ remains static and connected, nor that the diameter $d_{\mathcal{G}}$ of the network is

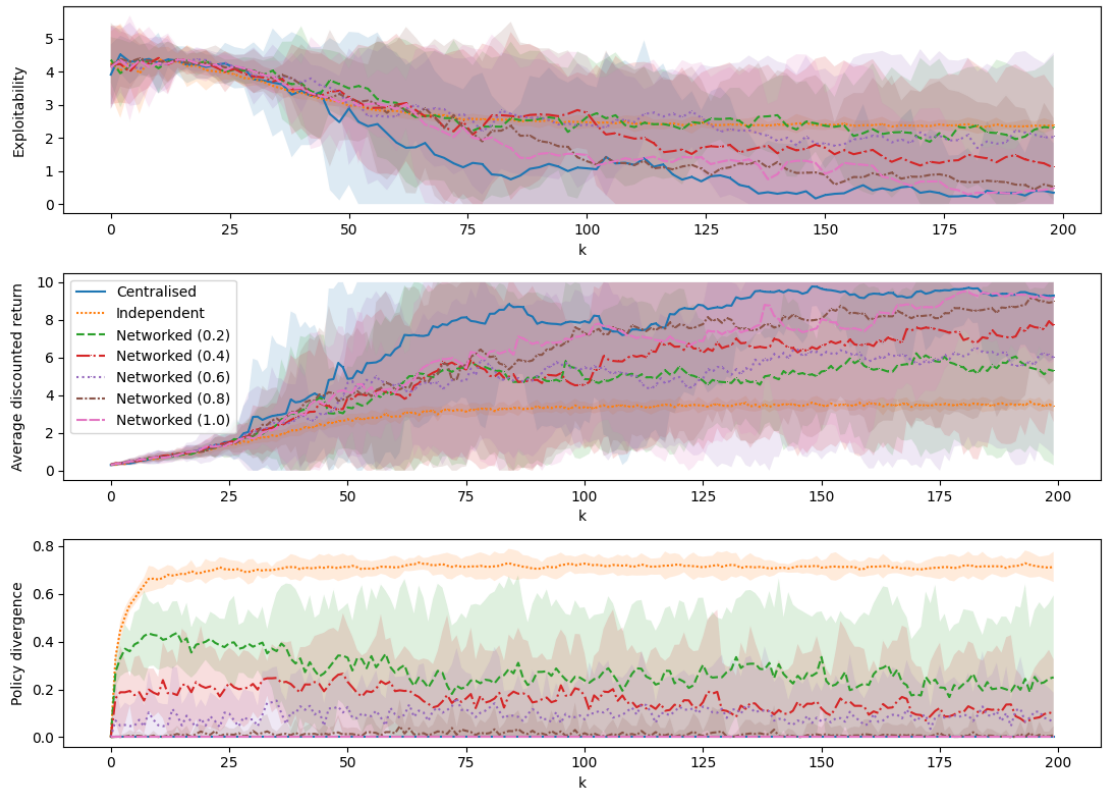


Figure 4.7: 'Target agreement' game. Even with only a single communication round, our networked case outperforms the independent case with respect to exploitability and return. The fact that the lowest broadcast radius (0.2, green) ends with similar exploitability to the independent case yet higher return suggests our networked algorithm might help agents find 'preferable' (i.e. Pareto-dominant) equilibria.

equal for all k. Nevertheless, even with a single communication round the networked agents learn faster than independent ones (which hardly learn at all), sometimes performing similarly to the centralised case.

- The M_{td} parameter is theoretically required for the learner to wait between collecting samples when learning from the empirical distribution in a continued, non-episodic system run. However, our replay buffer allows us to reduce it 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 MFG-NE unbiased and also removing Assumption 4.3.19. In general an unregularised

MFG-NE is not unique [15]; the ability of centralised and networked agents to coordinate on one of the multiple possible solutions (and in so doing also to reduce policy divergence) may help to explain why they outperform the independent case, as discussed further below (cf. Li et al. [61], Graber [160]).

- For the PMA operator (Def. 4.3.15), 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 distributions over actions instead of needing to identify the Lipschitz constants given in Assumption 4.3.8.

We now give further intuition into the benefits of our communication scheme in our empirical settings where multiple equilibria are possible. For sufficiently high λ the MFG-NE is unique, and involves all the agents constantly moving about with high entropy, at the cost of biasing the problem. However, when λ is 0, the 'target agreement' and 'cluster' tasks both explicitly admit multiple Pareto-optimal Nash equilibria. In a given trial of the 'target agreement' task, all the agents could converge to remaining stationary at any one of the four corners, and any one of these four situations would lead to the highest possible returns. We found in our experiments that with the different random seeds for each trial, agents did end up converging to a different corner at random each time. Similarly in the 'cluster' task: for a given trial all the agents could converge to remaining stationary in any one of the grid points, and any one of these $height \times width$ situations would lead to the highest possible returns. (In practice, empirically we found that the agents usually converged at random to one of the corners in the 'cluster' task as well, rather than to anywhere on the grid. This is because in the early stages of the trial, when agents start with random policies, they already spend more time visiting corners, because at any corner three actions will keep them in place, since they cannot move off the edge of the grid).

The discussion so far applies to *Pareto-optimal* Nash equilibria, i.e. the situations where agents end up with the highest possible returns (equivalent to a normalised average return of 10 in the plots). Population distributions can also be at an

equilibrium that does not receive particularly high returns or is not Nash: we can broadly characterise three situations here:

1. Agents, which begin the trial with random policies, never manage to reach any critical mass that breaks the ties between the possible coordination points, so continue moving about the grid with a high degree of entropy forever, even if λ is 0. This is most likely what is happening for the independent agents across the experiments, and is why they usually converge to low returns.

2. The population gets segregated into two or more isolated parts of the grid, each of which would otherwise give a (Pareto-optimal) Nash equilibrium if the whole population were present e.g. half the population learns a policy that remains in the top left corner while the other half learns to stay at the bottom right. If the policies do not retain enough exploration, the agents will never discover the other isolated groups with which they could combine for mutual benefit (whilst if there is too much exploration, we revert to one of the other suboptimal situations, depending on the value of λ).

3. The population is not segregated, but oscillates between two or more locations that would otherwise represent Pareto-optimal Nash equilibria, without ever being able to settle on stable policies that agree on one location. This is similar to Case 1, but with the number of meeting points that are visited having been narrowed down.

Case 1 is likely to receive the worst returns. How much worse Case 2 and 3 are than the Pareto-optimal Nash equilibria depends on the size of the segregated populations and/or the frequency of the visitations caused by the oscillations. The ability of learning architectures to align the behaviour of the *whole* population on a *single* choice of Nash equilibrium location determines how close to the maximum return the population will receive. The independent case has no way to align policies outside of the signal from the returns themselves; if no critical mass ever forms to show differentiation in the returns, then the independent population

will always remain at a Pareto-inefficient equilibrium. The central-agent case has an inherent method for aligning the policies of the whole population, but these policies may still oscillate between locations that would otherwise be Pareto-optimal Nash equilibria, which is why central-agent populations do not always reach the maximum returns in our plots.

Our communication algorithm provides a method both for 1) aligning agents' policies, and for 2) choosing better performing policies on which to align (where both of these elements contribute to the selection of better equilibria). This is why we see our decentralised, networked populations receiving higher returns than the independent ones, as our algorithm helps agents to get out of the worse performing equilibria. (In principle, under the right conditions, our communication paradigm could even outperform the central-agent case as we see in the subsequent chapters: the latter aligns the population on a policy update of arbitrary quality, generated by arbitrary agent $i = 1$, rather than aligning on better performing policies.) The degree to which our communication algorithm leads to policy consensus depends upon the network connectedness and the number of communication rounds. Since in our experiments we use $C = 1$, it is the network connectedness - determined by the size of the broadcast radius - that has the greatest effect (for greater numbers of communication rounds, this may matter less). This is why we see the populations with higher broadcast radii converging to higher returns faster than populations with lower broadcast radii, which are in turn more capable than entirely independent agents - they are better able to align the population so as to converge to equilibria that are closer to Pareto-optimal Nash equilibria.

In summary, the fact that different populations in our experiments do not just improve their returns at different speeds, but actually appear to converge to different final returns, is reflective of them settling at different equilibria that give different returns. Our communication algorithm actively helps populations to settle at equilibria that are closer to Pareto-optimal, i.e. 'preferable' (and in so doing, to choose between multiple possible Nash equilibria).

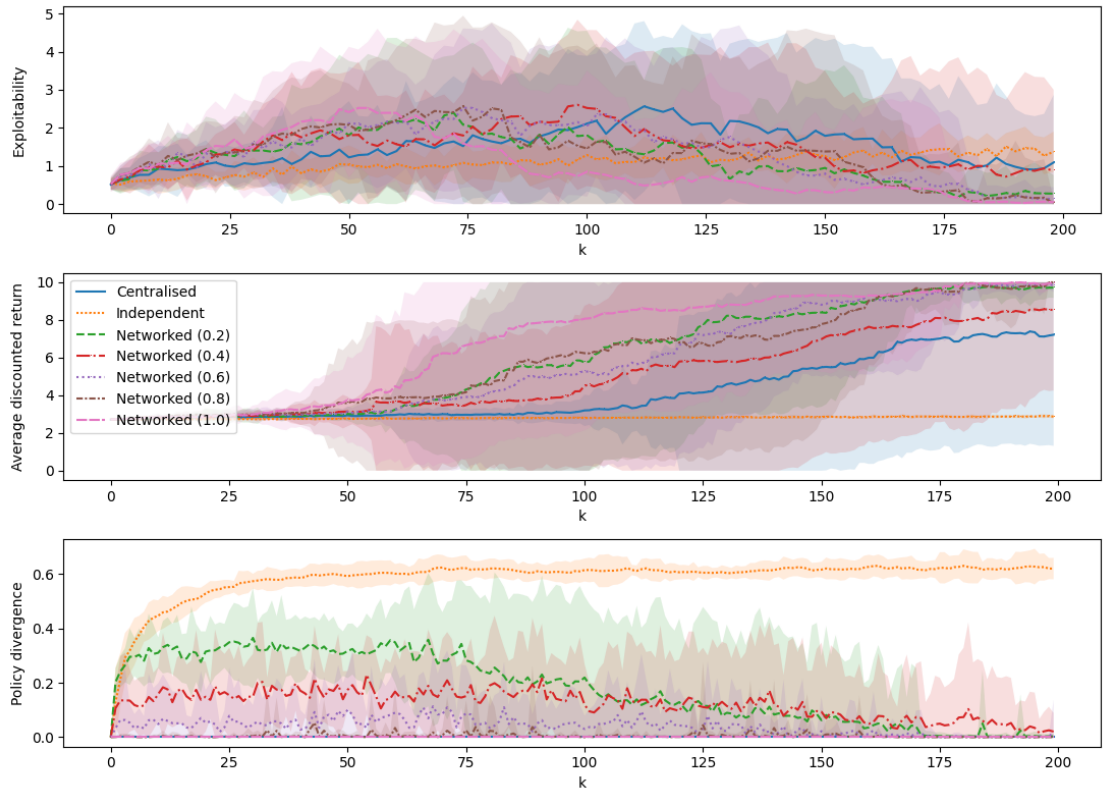


Figure 4.8: 'Cluster' game, testing robustness to 50% probability of policy update failure. The communication network allows agents that have successfully updated their policies to spread this information to those that have not, providing redundancy. Independent learners cannot do this and hardly appear to learn at all (no increase in return); likewise the centralised population is susceptible to its single point of failure and learns slower than before. Thus our networked architecture outperforms both the centralised and independent cases.

4.7.4.3 Robustness experiments

We consider two scenarios to which we desire real-world many-agent systems (e.g. robotic swarms, autonomous vehicle traffic, etc.) to be robust. The networked setup affords population **fault-tolerance** and **online scalability**, which are motivating qualities of many-agent systems.

Fault-tolerance We consider a scenario in which the learning/updating procedure of agents fails with a certain probability within each iteration, in which cases $\pi^i_{k+1} = \pi^i_k$ (see Figs. 4.8 and 4.9 for our experimental results in this scenario). In real-life decentralised settings, this might be particularly liable to occur since the

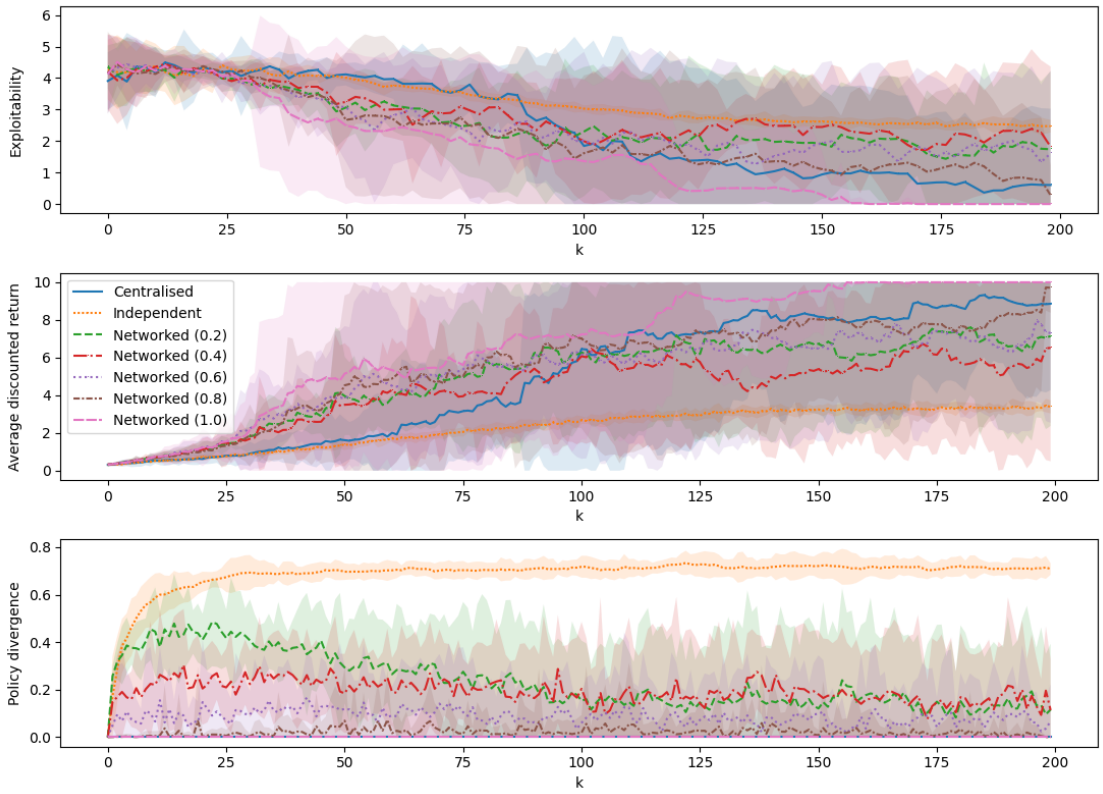


Figure 4.9: 'Target agreement' game, testing robustness to 50% probability of policy update failure. All the networked cases outperform the independent case and also learn faster than the centralised case for long periods. The communication network allows agents that have successfully updated their policies to spread this information to those that have not, providing redundancy. Independent learners cannot do this so have even slower convergence than normal in this task; likewise the centralised architecture is susceptible to its single point of failure, hence learning can be slower than in the networked case.

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. Regardless of their cause, such failures slow the improvement of the population in the independent case, and in the central-agent population 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 update failures, with the updated policies of any agents that have managed to learn spreading through the population to those that have not (cf. Horyna et al. [137]). This feature thus ensures that improvement can continue for potentially the whole population even

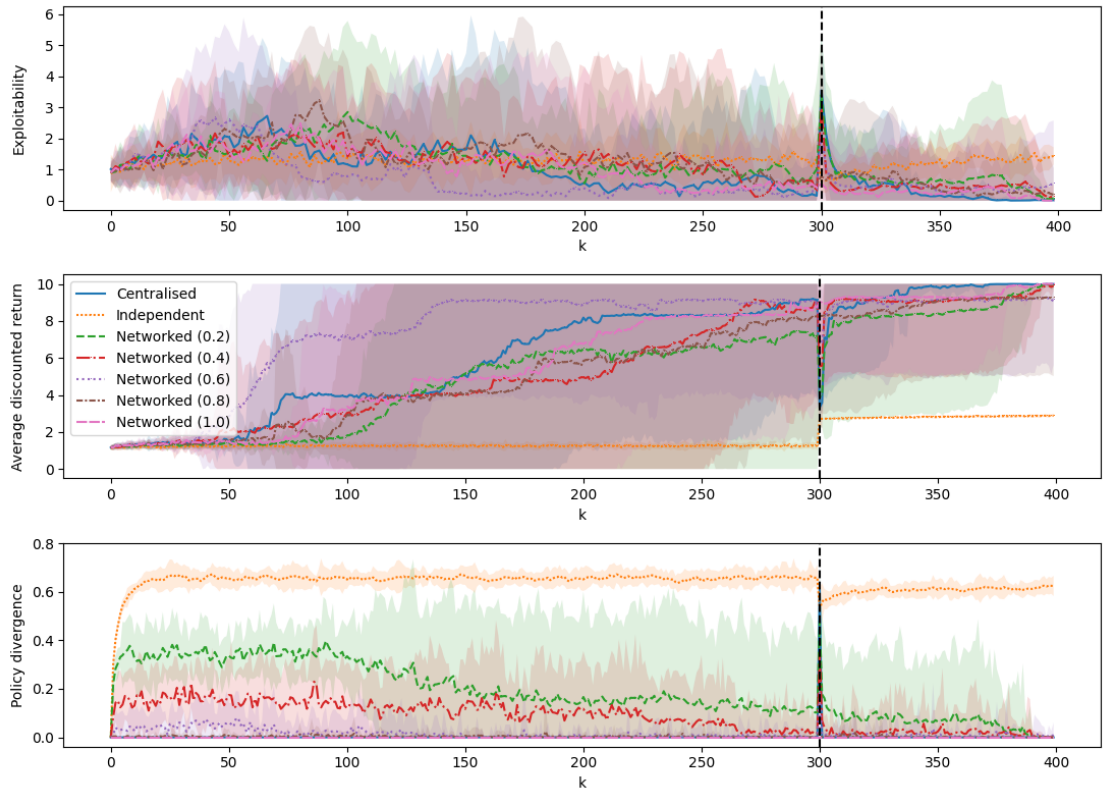


Figure 4.10: 'Cluster' game, testing robustness to a five-times increase in population. While the independent algorithm appears to enjoy similar exploitability to the other cases (see Rem. 4.7.2), we can see from its average return that it is not in fact learning at all; while the return rises after 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. 4.11, 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 populations perform similarly to or outperform the centralised case, and all markedly 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 at $k = 300$ is lower than in Fig. 4.6, is reflective of the fact that the error in the solution reduces as N tends to infinity.

if a high number of agents do not manage to learn at a given iteration.

Our experimental setup for this scenario is as follows: at every k iteration 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. See Figs. 4.8 and 4.9.

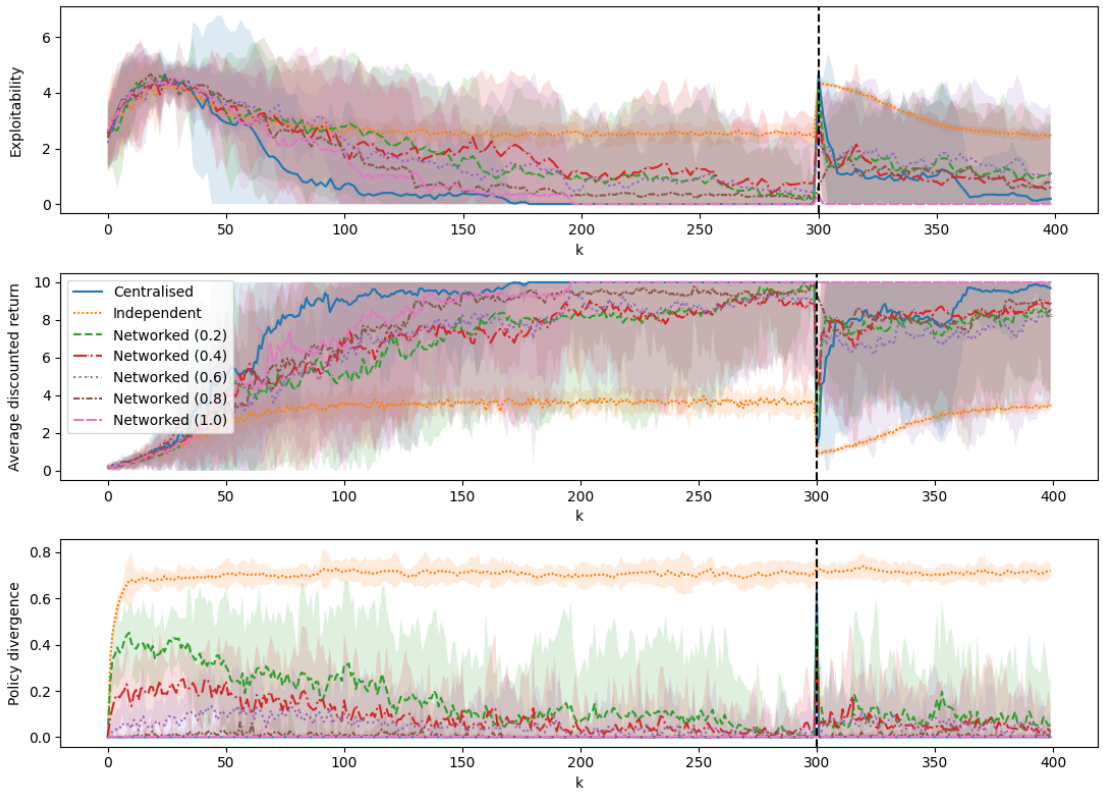


Figure 4.11: 'Target agreement' game, testing robustness to a five-times increase in population. The networked architectures are quickly able to spread the learnt policies to the newly arrived agents such that learning progress is minimally disturbed, whereas convergence is significantly impacted in the independent case. The largest broadcast radius (1.0, pink), in particular, suffers no disturbance at all, being more robust than the centralised case, which takes a significant amount of time to return to equilibrium.

Online scalability 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 Ch. 2 for comparison with other works considering this type of robustness [31, 153, 220, 221]. 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 MFG-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 MFG-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, without needing to rely on a central node.

Our experimental setup for this scenario is as follows: 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. 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. See Figs. 4.10 and 4.11.

Note that, for simplicity of notation of the communication network, we presume changes in population size occur outside of communication rounds, but this is not required by our algorithm. If agents are both leaving and (re)joining the population, we assume vertices/agents are indexed uniquely, rather than strictly from $1, \ldots, N$.

The remainder of our experiments provide further studies and ablations in the standard settings (i.e. not the robustness scenarios):

4.7.4.4 Experiments on larger grid

Figs. 4.12 and 4.13 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 outperform the independent case for most broadcast radii.

4.7.4.5 Ablation study of softmax temperature annealing scheme

Figs. 4.14 and 4.15 illustrate the effect of fixed $\{\tau_k\}_{k\in\{0,\ldots,K-1\}} = 100$, where the networked architecture does not perform as well as if we use the stepped inverse annealing scheme employed in all the other experiments and detailed in Table 4.1. The intuition behind the better performance achieved with the inverse annealing scheme is as follows. If we begin with small τ_k (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

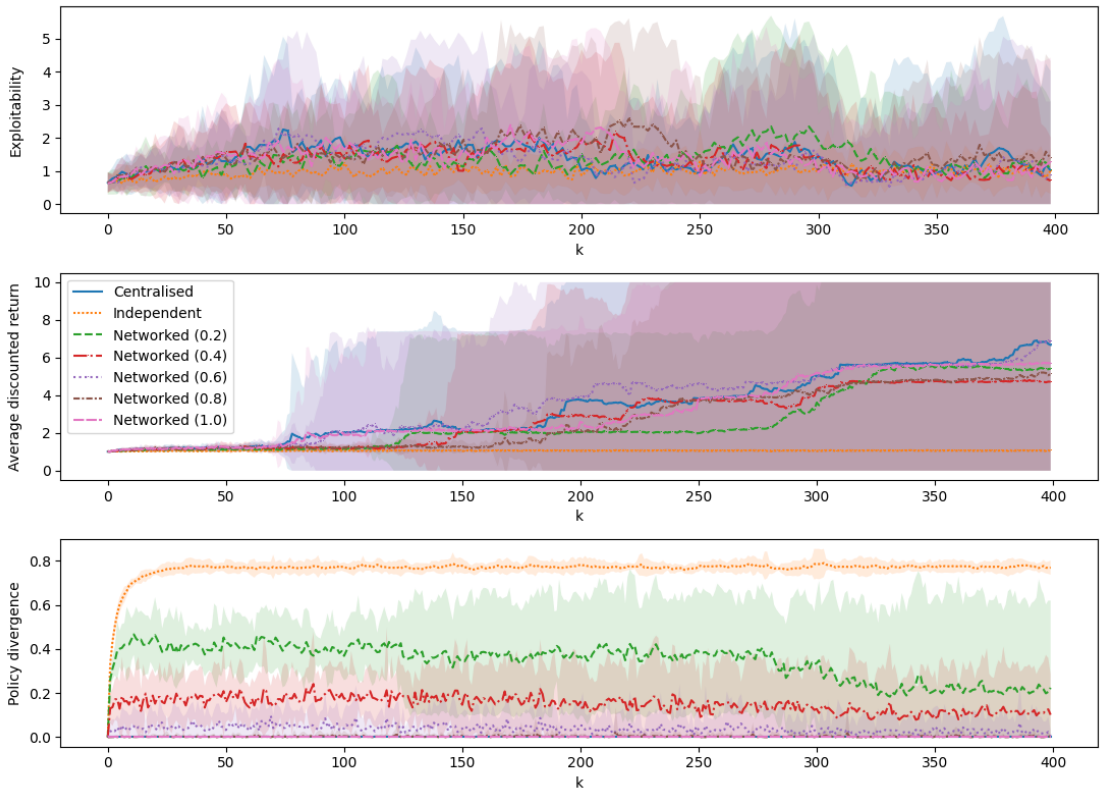


Figure 4.12: 'Cluster' game on the larger 16x16 grid. While the independent-learning case has similar exploitability to the other settings, we can see that it is not actually learning to increase its return at all, making this an undesirable equilibrium. (I.e. agents are moving about randomly so there 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. 4.7.2 and Sec. 4.7.4.2.) All the networked settings perform similarly to the centralised case and outperform the return of the independent agents.

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.

4.8 Conclusion

We contributed a networked communication scheme as a novel architecture for learning MFGs from the empirical distribution, and provided accompanying theoretical and practical algorithms. We showed theoretically and experimentally that networked agents can considerably outperform independent ones, often performing

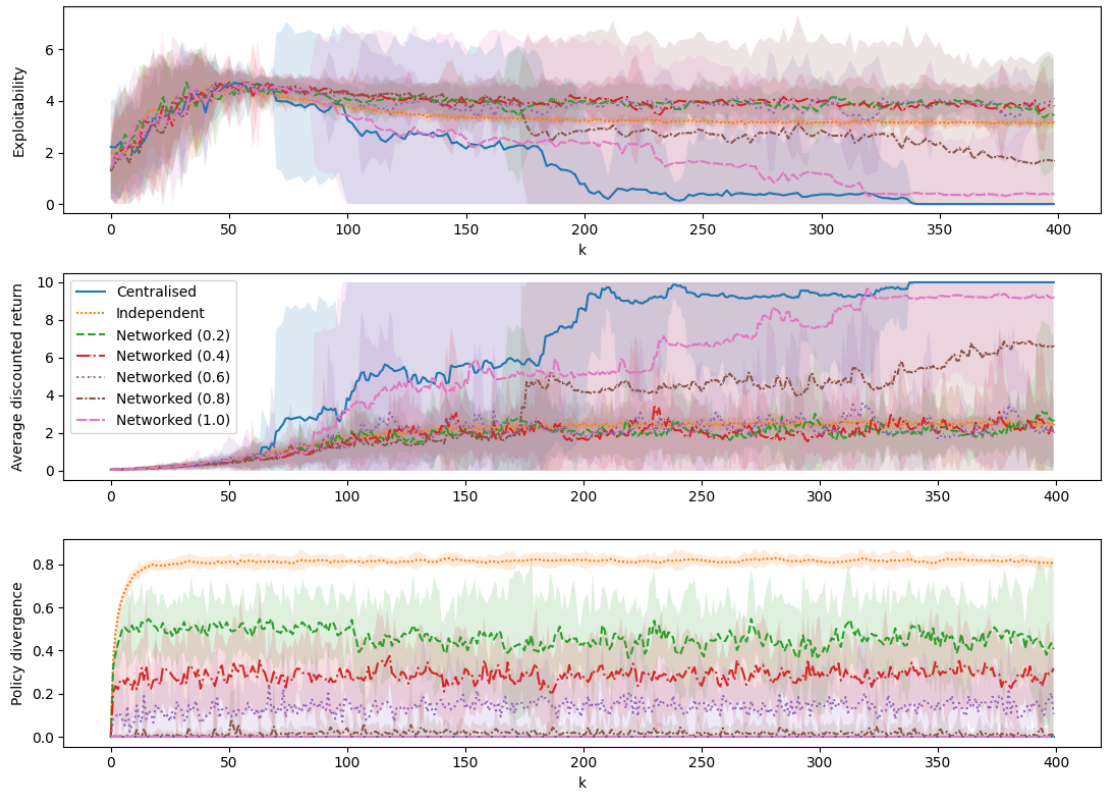


Figure 4.13: 'Target agreement' game on the larger 16x16 grid. There is greater differentiation in this setting than in the 8x8 grid (Fig. 4.7) between the different broadcast radii in the networked cases, as might be expected in a less densely populated environment. The two largest broadcast radii (1.0, pink, and 0.8, brown), which have the most connected networks, outperform the independent case in terms of both exploitability and return. However, the other broadcast radii perform similarly to the independent case.

similarly to the central-agent architecture while avoiding the restrictive assumption of the latter and its single point of failure. For discussion of potential avenues for future work, please see Ch. 7.

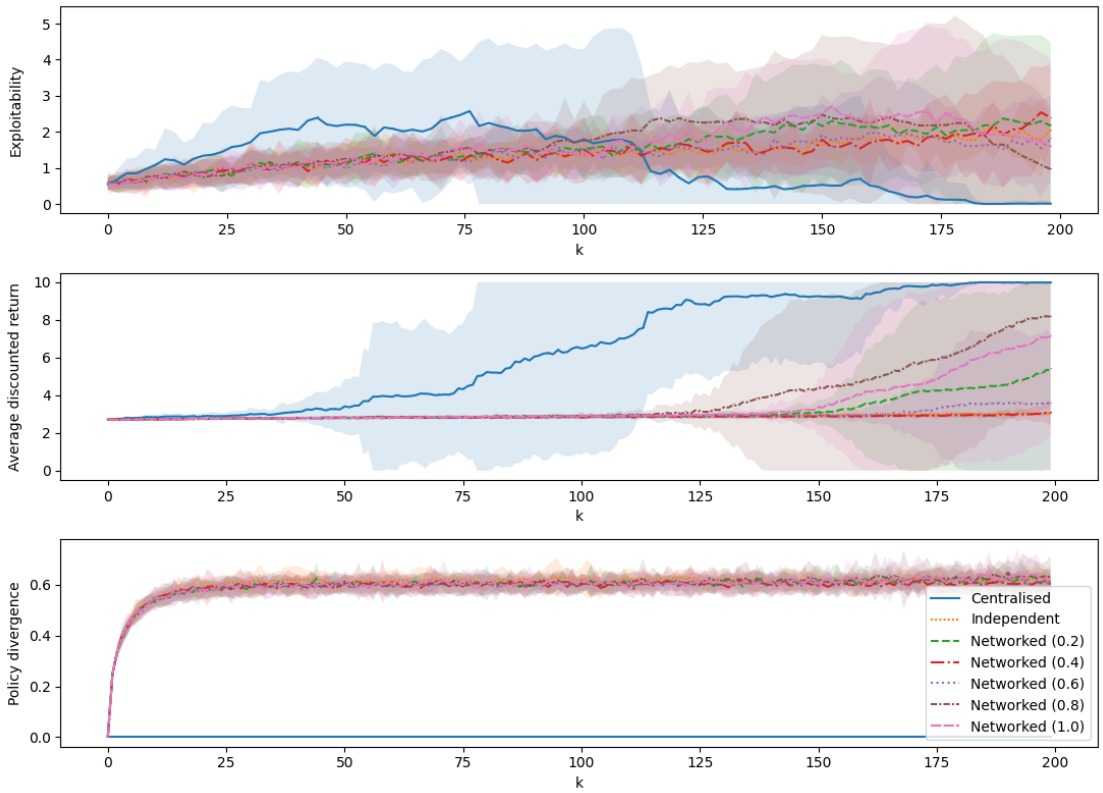


Figure 4.14: 'Cluster' game with τ_k fixed as 100 for all k; compare this to Fig. 4.6 where τ_k is stepped. Without the inverse annealing scheme, the networked architecture appears to perform similarly to the independent case in terms of exploitability, but several broadcast radii outperform the independent case in terms of return, demonstrating that our networked algorithm can still help agents find 'preferable' (i.e. Pareto-dominant) equilibria. However, whereas with annealing the networked architecture converges similarly to the centralised case, here it performs less well.

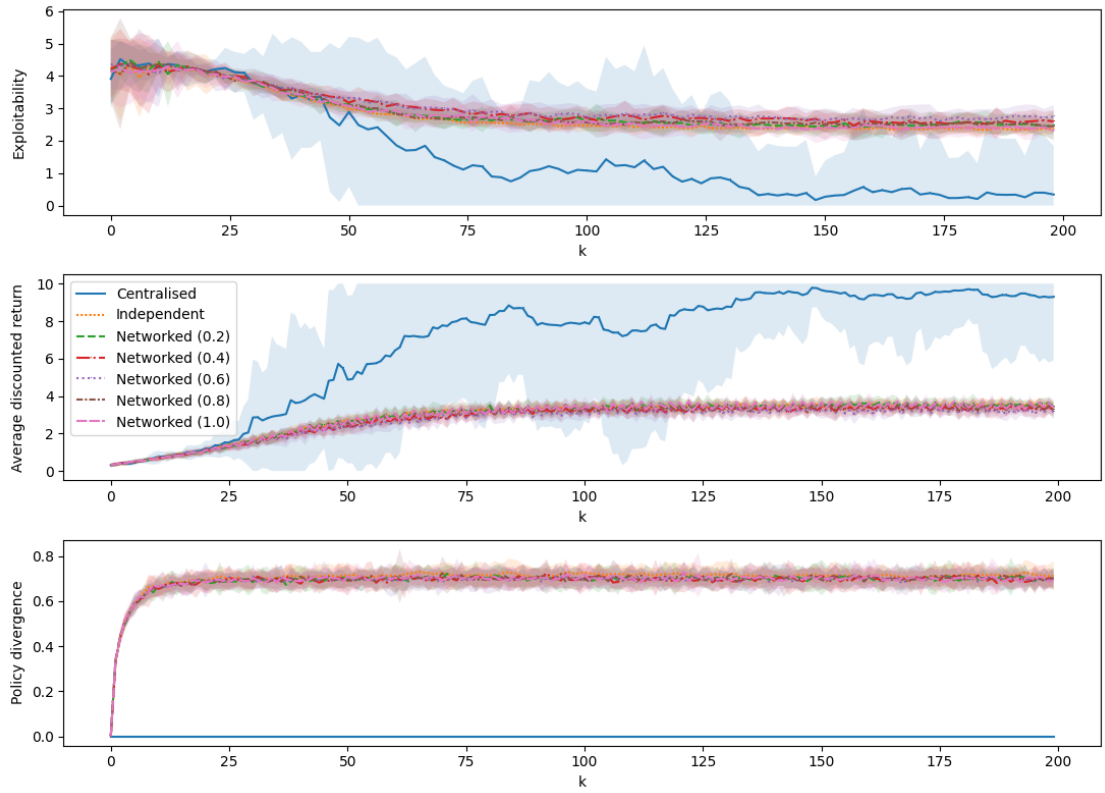


Figure 4.15: 'Target agreement' game with τ_k fixed as 100 for all k. Without our inverse annealing scheme for the softmax temperature, the networked architecture does not outperform the independent case. Compare this to Fig. 4.7 which shows the benefit of annealing τ_k.

5

MFGs with Function Approximation and Empirical Mean-Field Estimation

Contents

5.1 Introduction

We begin by recapping the content and contributions of this chapter, first stated in Sec. 1.2.2.

In Ch. 4 we introduced networked communication to MFGs where the state and action spaces are small enough that the Q-function can be represented by a table, limiting scalability. In Ch. 4 agents only needed to observe their local state as input to their Q-function (which defines their policy), since we focused on stationary MFGs.

We now wish to consider *population-dependent policies*, which depend on both the mean-field distribution and agents' local state [20, 100, 153–156]. The distribution is a large, high-dimensional observation object, taking a continuum of values. Therefore a population-dependent Q-function cannot be represented exactly in a table and must be approximated. To address these limitations while maintaining our desiderata for practical MFG algorithms, we now introduce function approximation to the MFG setting of decentralised agents learning online from a single, non-episodic run of the empirical system, allowing this setting to handle larger state spaces and to accept the mean-field distribution as an observation input. To overcome the difficulties of training non-linear approximators in this context, we use the so-called 'Munchausen' trick, introduced by Vieillard et al. [260] for single-agent RL, and extended to MFGs by Laurière et al. [20], and to MFGs with population-dependent policies by Wu et al. [153].

We demonstrate that in this setting, our decentralised, networked communication scheme brings two specific benefits over purely independent learning, while also removing the undesirable assumption of a central learner. Firstly, when the Q-function is approximated rather than exact, some updates lead to better performing policies than others. As before, allowing networked agents to propagate better performing policies through the population leads to faster learning and better avoidance of Pareto-inefficient equilibria than in the purely independent case.

However in this chapter we also now demonstrate that *networked agents can learn faster even than the central-agent case*, as we show both theoretically and empirically.

Secondly, we argue that in the real world it is unrealistic to assume that decentralised agents, endowed with local state observations and limited (if any) communication radius, would be able to observe the global mean-field distribution and use it as input to their Q-functions / policy. We therefore further contribute two setting-dependent algorithms by which decentralised agents can estimate the global distribution from local observations, and further improve their estimates by communication with neighbours.

Again, to pre-empt conceptual concerns about whether selfish agents would have incentive to communicate in non-cooperative MFGs, our experiments focus on coordination games. However, we show empirically that self-interested communicating agents can obtain higher returns than independent agents even in an anti-coordination game (Fig. 5.5), indicating that they may in fact have broader incentive to communicate across non-cooperative settings. Again we find no need to make a distinction between coordination and non-coordination games in our theoretical analysis, which holds across all types of non-cooperative MFG.

In summary, our contributions in this chapter are:

- We introduce, for the first time, function approximation to MFG settings with decentralised agents. To do this:
  - We use Munchausen RL for the first time in an infinite-horizon MFG context (for finite-horizon see Laurière et al. [20], Wu et al. [153]).
  - This constitutes the first use of function approximation for solving MFGs from a single, non-episodic run of the empirical system (for tabular settings see Yardim et al. [15] and our work in Ch. 4).
- Function approximation allows us to explore larger state spaces, and also settings where agents' policies depend on the mean-field distribution as well as their local state.

- Rather than assuming that agents have access to this global knowledge as in prior works, we present two additional novel algorithms allowing decentralised agents to locally estimate the empirical distribution and to improve these estimates by inter-agent communication.
- We prove theoretically that networked agents can learn faster even than central-agent populations in the function-approximation setting.
- We support this with extensive experiments, where our results demonstrate the two benefits of the decentralised communication scheme, which significantly outperforms both the independent and central-agent settings.

The rest of this chapter is structured as follows. Related work is given in Sec. 5.2. We give preliminaries in Sec. 5.3 and our core learning and policy-improvement algorithm in Sec. 5.4. We present our mean-field estimation and communication algorithms in Sec. 5.5, theoretical results in Sec. 5.6 and experiments in Sec. 5.7.

5.2 Related work

This section discusses the work most closely related to this chapter; please see Ch. 2 for work more generally related to networked communication in MFGs.

Laurière et al. [20] uses Munchausen Online Mirror Descent (MOMD), similar to our method for learning with neural networks, but their work has numerous differences to our setting: most relevantly, they study a finite-horizon episodic setting, where the mean-field distribution is updated in an exact way and an oracle supplies a central learner with rewards and transitions for it to learn a population-independent policy. Wu et al. [153] uses MOMD to learn population-dependent policies, albeit also with a central-agent method that exactly updates the mean-field distribution in a finite-horizon episodic setting. Perrin et al. [156] learns population-dependent policies with function approximation in infinite-horizon settings like our own, but does so in a central-agent, two-timescale manner without using the empirical mean-field distribution. Zhang et al. [208] also uses function

approximation along a non-episodic path, but involves a generic central agent learning an estimate of the mean field rather than using an empirical population. Approaches that directly update an estimate of the mean field must be able to generate rewards from this arbitrary mean field, even if they otherwise claim to be oracle-free. They are thus inherently centralised algorithms and rely on strong assumptions that may not apply in real-world problems. Conversely, we are interested in practical convergence in online, deployed settings, where the reward is computed from the empirical finite population.

Yongacoglu et al. [142] addresses decentralised learning from a continuous, non-episodic run of the empirical system using either full or compressed information about the mean field, but agents are assumed to receive this information directly, rather than estimating it locally as in the algorithm we now present. They also do not consider function approximation or inter-agent communication in their algorithms. In the distinct area of mean-field RL, which we disambiguate from MFGs in Ch. 2, Subramanian et al. [230] estimates the empirical mean-field distribution from the local neighbourhood. However, in that work agents are estimating the previous mean action rather than the current distribution over states as in our MFG setting. Their agents also do not have access to a communication network by which they can improve their estimates.

5.3 Preliminaries

5.3.1 Mean-field games

We use the notation introduced in Sec 3.1, as well as the following. For agent $i \in \{1 \dots N\}$, we now say that i's policy at time t depends on its observation o_t^i. We explore three different forms that this observation object can take:

- In the conventional setting, as in Ch. 4, the observation is simply i's current local state s_t^i, such that $\pi^i(a|o_t^i) = \pi^i(a|s_t^i)$.
- When the policy is population-dependent, if we assume perfect observability of the global mean-field distribution then we have $o_t^i = (s_t^i, \hat{\mu}_t)$.

- It is unrealistic to assume that decentralised agents with a possibly limited communication radius can observe the global mean field, so we allow agents to form a local estimate $\tilde{\hat{\mu}}_t^i$ that can be improved by communication with neighbours. Here we have $o_t^i = (s_t^i, \tilde{\hat{\mu}}_t^i)$.

In the following definitions we focus on the population-dependent case when $o_t^i = (s_t^i, \hat{\mu}_t)$, and clarify afterwards the connection to the other observation cases. Thus the set of policies is $\Pi = \{\pi : \mathcal{S} \times \Delta_{\mathcal{S}} \to \Delta_{\mathcal{A}}\}$, and the set of Q-functions is denoted $\mathcal{Q} = \{q : \mathcal{S} \times \Delta_{\mathcal{S}} \times \mathcal{A} \to \mathbb{R}\}$.

The following definition of symmetric anonymous games is equivalent to Def. 4.3.1 from the previous chapter.

Definition 5.3.1 (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. $\forall i = 1, \dots, N$:

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

As in Ch. 4, at the limit as $N \to \infty$, the infinite population of agents can be characterised as a limit distribution $\mu \in \Delta_{\mathcal{S}}$; the infinite-agent game is termed an MFG. The *mean-field flow* $\boldsymbol{\mu}$ is given by the infinite sequence of mean-field distributions s.t. $\boldsymbol{\mu} = (\mu_t)_{t \geq 0}$.

Definition 5.3.2 (Induced mean-field flow)**.** We denote by $I(\pi)$ the mean-field flow $\boldsymbol{\mu}$ induced when all the agents follow π, where this is generated from π as follows:

$$\mu_{t+1}(s') = \sum_{s,a} \mu_t(s) \pi(a | s, \mu_t) P(s' | s, a, \mu_t).$$

When the mean-field flow is stationary such that the distribution is the same for all t, i.e. $\mu_t = \mu_{t+1}\ \forall t \geq 0$, the policy $\pi^i(a|s_t^i, \mu_t)$ need not depend on the distribution, such that $\pi^i(a|s_t^i, \mu_t) = \pi^i(a|s_t^i)$, i.e. we recover the classical population-independent policy. However, for such a population-independent policy the initial distribution μ_0 must be known and fixed in advance, whereas otherwise it need not be.

The next definition is essentially an unregularised version of Def. 4.3.4 from the previous chapter.

Definition 5.3.3 (Mean-field discounted return)**.** In a MFG where all agents follow policy π giving a mean-field flow $\boldsymbol{\mu} = (\mu_t)_{t\geq 0}$, the expected discounted return of the representative agent is given by

$$V(\pi, \boldsymbol{\mu}) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t (R(s_t, a_t, \mu_t)) \middle| \begin{smallmatrix} s_0 \sim \mu_0 \\ a_t \sim \pi(\cdot|s_t, \mu_t) \\ s_{t+1} \sim P(\cdot|s_t, a_t, \mu_t) \end{smallmatrix}\right].$$

We add the following definition, which is not made explicit in the previous chapter.

Definition 5.3.4 (Best-response (BR) policy)**.** A policy π^* is a *best response (BR)* against the mean-field flow $\boldsymbol{\mu}$ if it maximises the discounted return $V(\cdot, \boldsymbol{\mu})$; the set of these policies is denoted $BR(\boldsymbol{\mu})$:

$$\pi^* \in BR(\boldsymbol{\mu}) := \arg\max_{\pi} V(\pi, \boldsymbol{\mu}).$$

The following definition is essentially a generalisation of Def. 4.3.5 to non-stationary equilibria.

Definition 5.3.5 (MFG Nash equilibrium (MFG-NE))**.** A pair $(\pi^*, \boldsymbol{\mu}^*)$ is a mean-field game Nash equilibrium if the following two conditions hold:

- π^* is a best response to $\boldsymbol{\mu}^*$, i.e. $\pi^* \in BR(\boldsymbol{\mu}^*)$;
- $\boldsymbol{\mu}^*$ is induced by π^*, i.e. $\boldsymbol{\mu}^* = I(\pi^*)$.

π^* is thus a fixed point of the map $BR \circ I$, i.e. $\pi^* \in BR(I(\pi^*))$. If a population-dependent policy is a MFG-NE policy for any initial distribution μ_0, it is a 'master policy'.

In the previous chapter we showed that, in tabular settings, it is possible for a finite population of decentralised agents (each of which is permitted to have a distinct population-independent policy π^i) to learn the MFG-NE using only the empirical distribution $\hat{\mu}_t$, rather than the exactly calculated infinite flow $\boldsymbol{\mu}$ [15]. This MFG-NE may be the goal in itself, or it can in turn serve as an approximate NE for the harder-to-solve game involving the finite population (discussed further in Rem. 1.2.1). In this chapter we provide algorithms to perform this process in non-tabular and population-dependent settings, and demonstrate their benefits theoretically and empirically.

5.3.2 (Munchausen) Online Mirror Descent

Instead of finding a BR at each iteration, which is computationally expensive, we can use a form of policy iteration for MFGs called Online Mirror Descent (OMD). This begins with an initial policy π_0, and then at each iteration k, evaluates the current policy π_k with respect to its induced mean-field flow $\boldsymbol{\mu} = I(\pi_k)$ to compute its Q-function Q_{k+1}. To stabilise the learning process, we then use a weighted sum over this and past Q-functions, and set π_{k+1} to be the softmax over this weighted sum, i.e. $\pi_{k+1}(\cdot|s,\mu) = softmax\left(\frac{1}{\tau_q}\sum_{\kappa=0}^{k+1} Q_\kappa(s,\mu,\cdot)\right)$. τ_q is a temperature parameter that scales the entropy in Munchausen RL [260] (see Sec. 5.4.1); note that this is a different temperature to the one agents use when selecting which communicated parameters to adopt, which we now denote τ_k^{comm} for disambiguation purposes (Sec. 5.4.2).

If the Q-function is approximated non-linearly using neural networks, it is difficult to compute this weighted sum. The *Munchausen trick* addresses this by computing a single Q-function that mimics the weighted sum using implicit regularisation based on the Kullback-Leibler (KL) divergence between π_k and π_{k+1} [260]. Using this reparametrisation gives Munchausen OMD (MOMD), detailed further in Sec. 5.4.1 [20, 153]. MOMD does not bias policies or the MFG-NE, and has the same convergence guarantees as OMD [153, 204, 261].

Algorithm 3 Networked learning with non-linear function approximation

Require: loop parameters K, M, L, E, C_p, learning parameters $\gamma, \tau_q, |B|, cl, \nu$, $\{\tau_k^{comm}\}_{k\in\{0,\dots,K-1\}}$

Require: initial states $\{s_0^i\}_{i=1}^N$; $t \leftarrow 0$

1: $\forall i$: Randomly initialise parameters θ_0^i of Q-networks $\check{Q}_{\theta_0^i}(o,\cdot)$, and set $\pi_0^i(a|o) = \text{softmax}\Big(\frac{1}{\tau_q}\check{Q}_{\theta_0^i}(o,\cdot)\Big)(a)$ and $\check{Q}_{\theta_{0,0}^{i,\prime}} \leftarrow \check{Q}_{\theta_0^i}(o,\cdot)$

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

3: $\quad\forall i$: Empty i's buffer

4: $\quad$**for** $m = 0, \dots, M-1$ **do**

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

6: $\quad\quad\forall i$: Add ζ_t^i to i's buffer

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

8: $\quad$**for** $l = 0, \dots, L-1$ **do**

9: $\quad\quad\forall i$: Sample batch $B_{k,l}^i$ from i's buffer

10: $\quad\quad$Update θ to minimise $\hat{\mathcal{L}}(\theta, \theta')$ as in Def. 5.4.1

11: $\quad\quad$If $l \mod \nu = 0$, set $\theta' \leftarrow \theta$

12: $\quad$**end for**

13: $\quad\check{Q}_{\theta_{k+1}^i}(o,\cdot) \leftarrow \check{Q}_{\theta_{k,L}^i}(o,\cdot)$

14: $\quad\forall i$: $\pi_{k+1}^i(a|o) \leftarrow \text{softmax}\Big(\frac{1}{\tau_q}\check{Q}_{\theta_{k+1}^i}(o,\cdot)\Big)(a)$

15: $\quad\forall i : \sigma_{k+1}^i \leftarrow 0$

16: $\quad$**for** $e = 0, \dots, E-1$ evaluation steps **do**

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

18: $\quad\quad\forall i : \sigma_{k+1}^i \leftarrow \sigma_{k+1}^i + \gamma^e \cdot r_t^i$

19: $\quad\quad t \leftarrow t+1$

20: $\quad$**end for**

21: $\quad$**for** C_p rounds **do**

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

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

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

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

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

27: $\quad$**end for**

28: **end for**

29: **return** policies $\{\pi_K^i\}_{i=1}^N$

5.4 Learning and policy improvement

5.4.1 Q-network and update

Lines 1-14 of our novel Alg. 3 contain the core Q-function/policy update method. Agent i has a neural network parametrised by θ_k^i to approximate its Q-function:

$\check{Q}_{\theta_k^i}(o,\cdot)$. The agent's policy is given by $\pi_{\theta_k^i}(a|o) = \text{softmax}\Big(\frac{1}{\tau_q}\check{Q}_{\theta_k^i}(o,\cdot)\Big)(a)$. We denote the policy $\pi_k^i(a|o)$ for simplicity when appropriate. Each agent maintains a buffer (of size M) of collected transitions of the form $\Big(o_t^i, a_t^i, r_t^i, o_{t+1}^i\Big)$. At each iteration k, they empty their buffer (Line 3) before collecting M new transitions (Lines 4-7); each decentralised agent i then trains its Q-network $\check{Q}_{\theta_k^i}$ via L training updates as follows (Lines 8-12).

For training purposes, i also maintains a target network $\check{Q}_{\theta_{k,l}^{i,'}}$ with the same architecture but parameters $\theta_{k,l}^{i,'}$ copied from $\theta_{k,l}^i$ less regularly than $\theta_{k,l}^i$ themselves are updated, i.e. only every ν learning iterations (Line 11). At each iteration l, the agent samples a random batch $B_{k,l}^i$ of $|B|$ transitions from its buffer (Line 9), and trains its neural network via stochastic gradient descent to minimise the following empirical loss (Def. 5.4.1, Line 10). For $cl < 0$, $[\cdot]_{cl}^0$ is a clipping function used to prevent numerical issues if the policy is too close to deterministic, as the log-policy term is otherwise unbounded [153, 260]:

Definition 5.4.1 (Empirical loss for Q-network)**.** This is given by:

$$\hat{\mathcal{L}}(\theta,\theta') = \frac{1}{|B|}\sum_{transition\in B_{k,l}^i}\Big|\check{Q}_{\theta_{k,l}^i}(o_t,a_t) - T\Big|^2,$$

where the target T is

$$\begin{aligned} T = r_t &+ [\tau_q \ln \pi_{\theta_{k,l}^{i,'}}(a_t|o_t)]_{cl}^0 \\ &+ \gamma\sum_{a\in\mathcal{A}}\pi_{\theta_{k,l}^{i,'}}(a|o_{t+1})\left(\check{Q}_{\theta_{k,l}^{i,'}}(o_{t+1},a) - \tau_q \ln \pi_{\theta_{k,l}^{i,'}}(a|o_{t+1})\right). \end{aligned}$$

5.4.2 Communication and adoption of parameters

We use the communication network $\mathcal{G}_t^{comm}$ to share two types of information at different points in Alg 3. One is used to improve local estimates of the mean field (Sec. 5.5). The other, described here, adapts our work in Ch. 4 for the function-approximation case, where now agents broadcast the parameters of the Q-network that defines their policy, rather than the Q-function table. It is similar to an unregularised version of the method introduced in Sec. 4.6.2, and is used to privilege the spread of better performing policy updates through the population,

allowing faster learning in this networked case than in the independent and even central-agent cases.

At each iteration k, after independently updating their Q-network and policy (Lines 3-14), agents *estimate* the infinite discounted return (Def. 5.3.3) of their new policies by collecting rewards for E steps (not added to the training buffer), and assign the finite-step discounted sum to σ^i_{k+1} (Lines 15-20). They then broadcast their Q-network parameters along with σ^i_{k+1} (Line 22). Receiving these from neighbours on the network, agents select which set of parameters to adopt by taking a softmax over their own and the received estimate values σ^j_{k+1} (Lines 23-25). They repeat the process for C_p rounds. This allows decentralised agents to adopt policy parameters estimated to perform better than their own, accelerating learning as shown in Sec. 5.6.

5.5 Mean-field estimation and communication

We now give our algorithms for decentralised estimation of the empirical categorical mean-field distribution. We first describe the general version, assuming the more general setting where $\mathcal{G}^{obs}_t$ applies (see discussion in Sec. 3.2). We subsequently detail how the algorithm can be made more efficient in environments where the more abstract visibility graph $\mathcal{G}^{vis}_t$ applies, as in our experimental settings. In both cases, the algorithm runs to generate the observation object when a step is taken in the main Alg. 3, i.e. to produce $o^i_t = (s^i_t, \tilde{\hat{\mu}}^i_t)$ for the steps $a^i_t \sim \pi^i_k(\cdot|o^i_t)$ in Lines 5, 17 and 26. Note that if $\mathcal{G}^{obs}_t/\mathcal{G}^{vis}_t$ are *fully* connected, all agents' estimated mean-field observations will be equivalent to the true categorical distribution, even without communicating with neighbours to improve their initial estimates. Both versions of the algorithm are subject to implicit assumptions, which we highlight and suggest methods for addressing in our discussion of limitations and future work in Sec. 7.2.2.

5.5.1 Algorithm for the general setting

In this setting, our method (Alg. 4) assumes each agent is associated with a unique ID to avoid the same agents being counted multiple times. Each agent maintains

Algorithm 4 Mean-field estimation and communication in general settings

Require: Time-dependent observation graph $\mathcal{G}_t^{obs}$, time-dependent communication graph $\mathcal{G}_t^{comm}$, states $\{s_t^i\}_{i=1}^N$, number of communication rounds C_e

1: $\forall i, s$: Initialise count vector $\hat{v}_t^i[s]$ with $\emptyset$
2: $\forall i : \hat{v}_t^i[s_t^j] \leftarrow \{ID^j\}_{j \in i \cup \{j' \in \mathcal{N} : (i,j') \in \mathcal{E}_t^{obs}\}}$
3: **for** c_e in $1, \ldots, C_e$ **do**
4: $\quad \forall i$: Broadcast $\hat{v}_{t,c_e}^i$
5: $\quad \forall i : J_t^i \leftarrow \{j \in \mathcal{N} : (i,j) \in \mathcal{E}_t^{comm}\}$
6: $\quad \forall i, s : \hat{v}_{t,(c_e+1)}^i[s] \leftarrow \hat{v}_{t,c_e}^i[s] \cup \{\hat{v}_{t,c_e}^j[s]\}_{j \in J_t^i}$
7: **end for**
8: $\forall i : counted\_agents_t^i \leftarrow \sum_{s \in \mathcal{S} : \hat{v}_t^i[s] \neq \emptyset} |\hat{v}_t^i[s]|$
9: $\forall i : uncounted\_agents_t^i \leftarrow N - counted\_agents_t^i$
10: $\forall i, s : \tilde{\hat{\mu}}_t^i[s] \leftarrow \frac{uncounted\_agents_t^i}{N \times |\mathcal{S}|}$
11: $\forall i, s$ where $\hat{v}_t^i[s]$ is not $\emptyset$: $\tilde{\hat{\mu}}_t^i[s] \leftarrow \tilde{\hat{\mu}}_t^i[s] + \frac{|\hat{v}_t^i[s]|}{N}$
12: **return** mean-field estimates $\{\tilde{\hat{\mu}}_t^i\}_{i=1}^N$

a 'count' vector $\hat{v}_t^i$ of length $|\mathcal{S}|$ i.e. of the same shape as the vector denoting the true empirical categorical distribution of agents. Each state position in the vector can hold a list of IDs. Before any actions are taken at each time step t, each agent's count vector $\hat{v}_t^i$ is initialised as full of $\emptyset$ ('no count') markers for each state (Line 1). Then, for each agent j with which agent i is connected via the observation graph, i places j's unique ID in its count vector in the correct state position (Line 2). Next, for $C_e \geq 0$ communication rounds, agents exchange their local counts with neighbours on the communication network (Line 4), and merge these counts with their own count vector, filtering out the unique IDs of those that have already been counted (Line 6). If $C_e = 0$ then the local count will remain purely independent. By exchanging these partially filled vectors, agents are able to improve their local counts by adding the states of agents that they have not been able to observe directly themselves.

After the C_e communication rounds, each state position $\hat{v}_t^i[s]$ either still maintains the $\emptyset$ marker if no agents have been counted in this state, or contains $x_s > 0$ unique IDs. The local mean-field estimate $\tilde{\hat{\mu}}_t^i$ is then obtained from $\hat{v}_t^i$ as follows. All states that have a count x_s have this count converted into the categorical probability x_s/N (we assume that agents know the total number of agents in the finite population,

even if they cannot observe them all at each t) (Line 11). The total number of agents counted in $\hat{v}_t^i$ is given by $counted\_agents = \sum_{s \in \mathcal{S}} x_s$, and the agents that have not been observed are $uncounted\_agents = N$ - $counted\_agents$. In this general setting, the unobserved agents are assumed to be uniformly distributed across all the states, so $uncounted\_agents/(N \times |\mathcal{S}|)$ is added to all the values in $\tilde{\hat{\mu}}_t^i$, replacing the $\emptyset$ marker for states for which no agents have been observed (Line 10).

5.5.2 Algorithm for visibility-based environments

We explain now the differences in our estimation algorithm (Alg. 5) for the subclass of environments where $\mathcal{G}_t^{vis}$ applies in place of $\mathcal{G}_t^{obs}$, i.e. the mutual observability of agents depends in turn on the mutual visibility of states. The benefit of $\mathcal{G}_t^{vis}$ over $\mathcal{G}_t^{obs}$ is that the former allows an agent in state s to obtain a correct, complete count $x_{s'} \geq 0$ of all the agents in state s', for any state s' that is visible to s (note the count may be zero). Unique IDs are thus not required as there is no risk of counting the same agent twice when receiving communicated counts: either *all* agents in s' have been counted, or no count has yet been obtained for s'. This simplifies the algorithm and helps preserve agent anonymity and privacy.

Secondly, uncounted agents cannot be in states for which a count has already been obtained, since the count is complete and correct, even if the count is $x_{s'} = 0$. Therefore after the C_e communication rounds, the $uncounted\_agents$ proportion needs to be uniformly distributed only across the positions in the vector that still have the $\emptyset$ marker (Line 13), and not across all states as in the general setting. This makes the estimation more accurate in this special setting.

We now describe the flow of Alg. 5. It begins with agents using the visibility graph $\mathcal{G}_t^{vis}$ to count the number of agents in locations that fall within the visibility radius (Line 2). For C_e communication rounds, agents can supplement this local count with those received from neighbours over the communication network $\mathcal{G}_t^{comm}$, in order to count agents that do not fall within the visibility radius (Lines 3-8). We assume agents know the population's total size N, and therefore can distribute the uncounted agents uniformly over the states that remain unaccounted for after

Algorithm 5 Mean-field estimation and communication for environments with $\mathcal{G}_t^{vis}$

Require: Time-dependent visibility graph $\mathcal{G}_t^{vis}$, time-dependent communication graph $\mathcal{G}_t^{comm}$, states $\{s_t^i\}_{i=1}^N$, number of communication rounds C_e

1: $\forall i, s$: Initialise count vector $\hat{v}_t^i[s]$ with $\emptyset$
2: $\forall i$ and $\forall s' \in \mathcal{S}' : (s_t^i, s') \in \mathcal{E}_t^{vis} : \hat{v}_t^i[s'] \leftarrow \sum_{j \in \{1,\dots,N\}: s_t^j = s'} 1$
3: **for** c_e in $1, \dots, C_e$ **do**
4: $\quad \forall i$: Broadcast $\hat{v}_{t,c_e}^i$
5: $\quad \forall i : J_t^i = i \cup \{j \in \mathcal{N} : (i,j) \in \mathcal{E}_t^{comm}\}$
6: $\quad \forall i, s$: Initialise new count vector $\hat{v}_{t,(c_e+1)}^i[s]$ with $\emptyset$
7: $\quad \forall i, s$ and $\forall j \in J_t^i : \hat{v}_{t,(c_e+1)}^i[s] \leftarrow \hat{v}_{t,c_e}^j[s]$ if $\hat{v}_{t,c_e}^j[s] \neq \emptyset$
8: **end for**
9: $\forall i : counted\_agents_t^i \leftarrow \sum_{s \in \mathcal{S}: \hat{v}_t^i[s] \neq \emptyset} \hat{v}_t^i[s]$
10: $\forall i : uncounted\_agents_t^i \leftarrow N - counted\_agents_t^i$
11: $\forall i : unseen\_states_t^i \leftarrow \sum_{s \in \mathcal{S}: \hat{v}_t^i[s] = \emptyset} 1$
12: $\forall i, s$ where $\hat{v}_t^i[s]$ is not $\emptyset$: $\tilde{\hat{\mu}}_t^i[s] \leftarrow \frac{\hat{v}_t^i[s]}{N}$
13: $\forall i, s$ where $\hat{v}_t^i[s]$ is $\emptyset$: $\tilde{\hat{\mu}}_t^i[s] \leftarrow \frac{uncounted\_agents_t^i}{N \times unseen\_states_t^i}$
14: **return** mean-field estimates $\{\tilde{\hat{\mu}}_t^i\}_{i=1}^N$

the communication rounds (Lines 9-11). Agents now have a vector containing a true or estimated count for every state; this is converted to an estimated empirical mean field by dividing all counts by N (Lines 12-13).

Remark 5.5.1. In our Algs. 4 and 5, agents share their local *counts* with neighbours on the communication network $\mathcal{G}_t^{comm}$, and only after the C_e communication rounds do they complete their estimated distribution by distributing the uncounted agents along their vectors. An alternative would be for each agent to immediately form a local *estimate* from their local count obtained via $\mathcal{G}_t^{obs}$ or $\mathcal{G}_t^{vis}$, which only then would be communicated and updated via the communication network. However, we take the former approach to avoid poor local estimates spreading through the network and leading to widespread inaccuracies. Information that is certain (the count) is spread as widely as possible, before being locally converted into an estimate of the total mean field. The same would be the case in our extension proposed in Sec. 7.2.2 for averaging noisy counts, i.e. only the counts would be averaged, with the estimates completed by distributing the remaining agents after the C_e communication rounds.

5.6 Theoretical results

5.6.1 Introduction

To demonstrate the benefits of the networked architecture by comparison, we also consider the results of baseline central-agent and independent architectures given by alternative versions of our algorithm. As in Ch. 4 and Yardim et al. [15]:

- In the **central-agent** setting, the Q-network updates of arbitrary agent $i = 1$ are automatically pushed to all other agents. For our new sub-routine the true global mean-field distribution is always used in place of the local estimate i.e. $\tilde{\hat{\mu}}_t^i = \hat{\mu}_t$.
- In the **independent** case, there are no links in $\mathcal{G}_t^{comm}$ or $\mathcal{G}_t^{obs}/\mathcal{G}_t^{vis}$, i.e. $\mathcal{E}_t^{comm} = \mathcal{E}_t^{obs} = \mathcal{E}_t^{vis} = \emptyset$.

Networked populations generally learn faster than both central-agent and independent ones in our experiments. To indicate how this is possible while allowing simplicity of the theory, we consider a special case involving three assumptions that give conditions under which networked populations provably do outperform central-agent ones. We explore when these assumptions apply in practice, and discuss how even when loosening them, the intuition provided by Thm. 5.6.4's proof still offers useful heuristic insight as to why our networked agents can learn faster than the alternative architectures. We do not enforce the assumptions in our experiments, and our empirical results nevertheless usually follow our theoretical result.

All expectations in this section are taken jointly over:

- the initial joint state $\{s_0^i\}_{i=1}^N$ sampled from μ_0;
- the stochastic transitions, actions, and rewards collected by each agent into its individual replay buffer;
- the stochastic mini-batch samples $B_{k,l}^i$ drawn from those buffers and used in the Q-network loss in Def. 5.4.1;
- any random initialisation of the Q-network parameters θ_0^i;

- the softmax draws over neighbours' policies in Line 24 of Alg. 3.

The communication network $\mathcal{G}_t^{comm}$ and visibility graph $\mathcal{G}_t^{vis}$ are supplied as exogenous inputs at each round rather than as random variables in $\mathbb{E}[\,\cdot\,]$. They may or may not be functions of the agents' positions. In our experiments they are determined by communication and visibility radii; this makes them deterministic conditional on positions, so they inherit randomness only implicitly through the stochastic state evolution. The theory itself, however, permits any sequence of graphs satisfying the relevant structural conditions. We do not analyse explicitly random graph models (e.g. Erdős-Rényi link sampling). Our bounds hold for any realisation of $(\mathcal{G}_t^{comm}, \mathcal{G}_t^{vis})$ consistent with those conditions.

5.6.2 Analysis

The first assumption simplifies the theory by presuming that it is only the decentralised policy communication scheme that creates a difference in learning between the networked and central-agent cases, by assuming that the estimated mean fields are equivalent to the true ones used in the central-agent case (this is only relevant for population-dependent policies). Note that populations with fully connected observation/visibility graphs will in any case always be able to accurately estimate $\hat{\mu}_t$ by Algs. 4 and 5, even for $C_e = 0$. This is may apply reasonably commonly in practice depending on the scenario; for example, if the network is defined by a broadcast radius (as in our experiments), then the network will be fully connected whenever that radius is at least large enough to cover the area that all the agents fall within. In our experiments we set $C_e = 1$ to show the benefit of even just one communication round, but we do not enforce this assumption, and still generally see our theoretical result holding empirically. We leave analysis of the theoretical impact of worsening mean-field estimates for future work.

Assumption 5.6.1. Assume that Algs. 4 and 5 allow networked agents to obtain accurate estimates of the global empirical mean field, i.e. $\forall i\ \tilde{\hat{\mu}}_t^i = \hat{\mu}_t$.

Now recall that at each iteration k of our networked Alg. 3, after independently updating their policies in Line 14, the population has the policies $\{\pi^i_{k+1}\}^N_{i=1}$. There is randomness in these independent policy updates, stemming from the random sampling of each agent's independently collected buffer. In Lines 15-20, agents estimate the infinite discounted returns $\{V(\pi^i_{k+1}, I(\pi^i_{k+1}))\}^N_{i=1}$ (Def. 5.3.3) of their updated policies by computing $\{\sigma^i_{k+1}\}^N_{i=1}$: the E-step discounted return with respect to the *empirical* mean field generated when agents follow the individual policies $\{\pi^i_{k+1}\}^N_{i=1}$. We can characterise the approximation as $\{\sigma^i_{k+1}\}^N_{i=1}$ $= \{\widehat{V}(\pi^i_{k+1}, I(\pi^i_{k+1}); E)\}^N_{i=1}$.

Our second assumption presumes that the networked population reaches consensus on a single policy within the policy communication rounds of each k iteration. We assume this to give general and intuitive comparison with the central-agent population which always shares a single policy. Incomplete consensus would by definition give different levels of strategy alignment/diversity, such that the relative performance of the central-agent and networked architectures might then depend on the specific reward function of the (coordination) task, and the weight placed on alignment in that reward function compared to other characteristics of the policy.

Assumption 5.6.2. After the C_p communication rounds in Lines 21-27 in which agents exchange and adopt policies from neighbours, the networked population is left with a single policy such that $\forall i, j \in \{1, \dots, N\}$ $\pi^i_{k+1} = \pi^j_{k+1}$.

While this may sound like a strong assumption, we phrase it like this so as not to make overly strong restrictions on the communication network instead - we intentionally leave it so that Assumption 5.6.2 can be fulfilled in numerous ways. Most simply we can think of Assumption 5.6.2 holding if:

1. we set τ^{comm}_k close to 0 for all k, such that the softmax essentially becomes a max function; and

2. the communication network $\mathcal{G}^{comm}_t$ is static and connected during the C_p communication rounds, where C_p is at least as large as the network diameter $d_{\mathcal{G}^{comm}_t}$.

Under these conditions, as discussed in Ch. 4, previous results on max-consensus algorithms show that all agents in the network will converge on the highest value σ_{k+1}^{max} (and hence the associated π_{k+1}^{max}) within a number of rounds equal to the diameter $d_{\mathcal{G}_t^{comm}}$ [255]. If we assumed more strongly that the network was always *fully* connected, policy consensus would be achieved within a single communication round.

As in Ch. 4, policy consensus can be achieved even outside of these conditions, including if the network is dynamic and not connected at every step. Recall from Sec. 3.2 that a collection of graphs is *jointly connected* if its members' union is connected. Now, instead of assuming that the communication network is static and connected, we assume instead only that the sequence of networks contains one or more sequential jointly connected collections. Then max-consensus is reached within C_p if C_p is large enough that the number of sequential jointly connected collections occurring within C_p is equal to the largest diameter of the union of any such collection.

Thus Assumption 5.6.2 may not hold if C_p is not large enough or if parts of the population remain isolated. However, we do not enforce this assumption in our experiments, where we use $C_p = 1$ to show the benefit of even just one communication round, yet we still generally see networked populations significantly outperforming central-agent populations. However networked populations that are less connected (due to having smaller broadcast radii) usually outperform central-agent populations by a smaller margin. This is probably due to Assumptions 5.6.1 and 5.6.2 being empirically more likely to be violated in less connected populations. Nevertheless the intuition provided by Thm. 5.6.4's proof below indicates why networked populations are still able to perform better than central-agent populations even if these assumptions are loosened.

We also add the following assumption that the finite-step estimates of the returns give sufficiently accurate comparisons between policies, so that better policies are indeed the ones that get adopted in expectation:

Assumption 5.6.3. Assume that $\{\sigma_{k+1}^i\}_{i=1}^N$ are sufficiently good approximations so as to respect the ordering of the true values $\{V(\pi_{k+1}^i, I(\pi_{k+1}^i))\}_{i=1}^N$, i.e. $\forall i, j \in$

$\{1, \ldots, N\}$:

$$\sigma^i_{k+1} > \sigma^j_{k+1} \iff V(\pi^i_{k+1}, I(\pi^i_{k+1})) > V(\pi^j_{k+1}, I(\pi^j_{k+1})).$$

In practice, even if Assumption 5.6.3 does not strictly hold, the softmax parameter τ^{comm}_k allows a smooth degradation as the ordering of the estimates worsens with respect to the ordering of the true values. That is, if instead of the exact correct policy ordering we have that better policies are simply *more likely* to be given higher estimated evaluations, then the softmax means that these policies remain *more likely* to spread, and a better policy may still be adopted even if it is not evaluated as being better.

We are now nearly ready to give our theoretical result. Call the network consensus policy π^{net}_{k+1}, and its associated finitely estimated return $\sigma^{\text{net}}_{k+1}$. Recall that the central-agent case is where the Q-network update of arbitrary agent $i = 1$ is automatically pushed to all the others instead of the policy exchange in Lines 15-27; this is equivalent to a networked case where policy consensus is reached on a *random* one of the policies $\{\pi^i_{k+1}\}^N_{i=1}$. Call this policy *arbitrarily* given to the whole population π^{cent}_{k+1}, and its associated finitely estimated return $\sigma^{\text{cent}}_{k+1}$. Now we can say:

Theorem 5.6.4. *Given Assumptions 5.6.1, 5.6.2 and 5.6.3,*

$$\mathbb{E}[V(\pi^{\text{net}}_{k+1}, I(\pi^{\text{net}}_{k+1}))] > \mathbb{E}[V(\pi^{\text{cent}}_{k+1}, I(\pi^{\text{cent}}_{k+1}))].$$

Thus in expectation networked populations will increase their returns faster than central-agent ones.

Proof. Before the communication rounds in Line 21 (Alg. 3), the randomly updated policies $\{\pi^i_{k+1}\}^N_{i=1}$ have associated estimated returns $\{\sigma^i_{k+1}\}^N_{i=1}$. Call the mean and maximum of this set $\sigma^{\text{mean}}_{k+1}$ and $\sigma^{\text{max}}_{k+1}$ respectively. Since π^{cent}_{k+1} is chosen arbitrarily from $\{\pi^i_{k+1}\}^N_{i=1}$, it will obey $\mathbb{E}[\sigma^{\text{cent}}_{k+1}] = \sigma^{\text{mean}}_{k+1}\ \forall k$, though there will be high variance. Conversely, the softmax adoption probability (Line 24, Alg. 3) for the networked case means by definition that policies with higher σ^i_{k+1} are more likely to be adopted at each communication round. Thus the π^{net}_{k+1} that is adopted by the whole networked

population will obey $\mathbb{E}[\sigma^{\text{net}}_{k+1}] > \sigma^{\text{mean}}_{k+1}$ (if τ^{comm}_{k+1} is a positive value near zero, it will obey $\mathbb{E}[\sigma^{\text{net}}_{k+1}] = \sigma^{\text{max}}_{k+1}\ \forall k$). So $\mathbb{E}[\sigma^{\text{net}}_{k+1}] > \mathbb{E}[\sigma^{\text{cent}}_{k+1}]$, which by Assumption 5.6.3 implies the result. □

The adoption scheme in Line 24 biases the spread of policies towards those estimated to be better, which, given sufficiently good approximations (Assumption 5.6.3), results in higher discounted returns in practice. By choosing updates in a more principled way, networked agents learn faster than the central-agent case that pushes updates regardless of quality. Similar logic can also be applied to understand why networked agents outperform entirely independent ones, combined with the fact that greater policy diversity in the independent case worsens sample complexity over the networked and central-agent cases by biasing approximations of the Q-function, as discussed in Ch. 4 [15].

Significantly, the communication scheme not only allows us to avoid the undesirable assumption of a central learner, but even to outperform it. Moreover, we will see in the next section that empirically the benefit of networked communication over central-agent learning is greater in this function approximation setting than in the tabular case of Ch. 4, perhaps due to greater variance in the quality of Q-function estimates in the present case. This shows that networked communication facilitates greater scalability than the central-agent paradigm.

5.7 Experiments

We provide two sets of experiments. The first set demonstrates that our function-approximation algorithm (Alg. 3) can scale to large state spaces for population-independent policies, and that in such settings networked, communicating populations can outperform purely independent agents (by an even greater margin than in the tabular settings from Ch. 4) and even central-agent populations. The second set demonstrates that Alg. 3 can handle population-dependent policies, as well as the ability of Alg. 5 to practically estimate the mean-field distribution locally.

5.7.1 Experimental set-up

As in Ch. 4, for the types of game used in our experiments we follow the gold standard in prior MFG works, i.e. grid-world environments where agents can move in the four cardinal directions or remain in place [20, 99, 110, 126, 153, 200]. We present results from five tasks defined by the agents' reward/transition functions, four of which are *coordination* tasks, while the fifth is a non-coordination task. In all cases, rewards are normalised in [0,1] after they are computed. The first two tasks are the same as those used with population-independent policies in Ch. 4, but while there we showed results for an 8×8 and a 'larger' 16×16 grid, our results here are for 100×100 and 50×50 grids. We restate those tasks here for ease of reference:

- **Cluster.** This is the inverse of the 'exploration' game in Laurière et al. [20], 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.

- **Target agreement.** 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 to maximise their individual rewards. Define the magnitude of the distances between x, y at t as $dist_t(x, y)$. The reward function is given by $R(s_t^i, a_t^i, \hat{\mu}_t) = r_{targ}(r_{collab}(\hat{\mu}_t(s_t^i)))$, where

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

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

As in Ch. 4, in these respective games the Pareto-dominant MFG-NE have all agents at a single state ('cluster', $|\mathcal{S}|$ Pareto-optimal MFG-NE) or at a single target corner ('target agreement', $|\Phi| = 4$ Pareto-optimal MFG-NE), each with a continuum of partition-based Pareto-dominated NE. With the larger 100×100 / 50×50 grids in this chapter, the time required for a randomly initialised population to first encounter such a configuration is much longer, so we expect the gap between networked and independent learners to widen relative to the 8×8 / 16×16 tabular results. In the 'cluster' task we again observe agents settling at one of the four corners despite the reward being symmetric across states, driven by the same action-asymmetry described in Ch. 4; in the 'target agreement' task the four targets are themselves placed at corner cells, so the corner outcome is reward-driven rather than only geometric.

We also demonstrate the ability of our algorithms to handle two more complex tasks, using population-dependent policies and estimated mean-field observations:

- **Evade shark in shoal.** This is a similar idea to the task found in Liu et al. [262]. Define the magnitude of the horizontal and vertical distances between x, y at t as $dist_t^h(x, y)$ and $dist_t^v(x, y)$ respectively. The state s_t^i now consists of agent i's position x_t^i and a 'shark's' position ϕ_t. At each time step, the shark steps towards the most populated grid point according to the empirical mean-field distribution i.e. $x_t^* = \arg\max_{x \in \mathcal{S}} \hat{\mu}_t(x)$. A horizontal step is taken if $dist_t^h(\phi_t, x_t^*) \geq dist_t^v(\phi_t, x_t^*)$, otherwise a vertical step is taken. As well as featuring a non-stationary distribution, we add 'common noise' to the environment, with the shark moving in a random direction with probability 0.01. As noted in Sec. 1.2.2, such noise that affects the local states of all agents in the same way, making the evolution of the distribution stochastic, makes population-independent policies sub-optimal [38, 100, 144, 154, 155].

Agents are rewarded more for being further from the shark, and also for clustering with other agents. The reward function is given by

$$R(s_t^i, a_t^i, \hat{\mu}_t) = \ dist_t^h(\phi_t, x_t^i) + dist_t^v(\phi_t, x_t^i) + \text{norm}_{dist}(\log(\hat{\mu}_t(x_t^i))),$$

where $\text{norm}_{dist}(\cdot)$ indicates that the final term is normalised to have the same maximum and minimum values as the total combined vertical and horizontal distance.

Since the shark steps toward the most populated grid point, any persistent cluster pulls the shark with it, so the population must flee while remaining co-located. We expect the algorithm to drive the population toward a moving cluster that maintains near-maximal distance from the shark. The induced mean-field flow is itself non-stationary, so unlike the previous tasks the Pareto-dominant MFG-NE here are not stationary.

- **Push object to edge.** This is similar to the task presented in Cunha Queiroz and MacRae [263]. As before, define the magnitude of the horizontal and vertical distances between x, y at t as $dist_t^h(x, y)$ and $dist_t^v(x, y)$ respectively. The state s_t^i consists of agent i's position x_t^i and an 'object's' position ϕ_t. The number of agents in the positions surrounding the object at time t generates a probability field around the object, such that the object is most likely to move in the direction away from the side with the most agents. As such, if agents are equally distributed around the object, it will be equally likely to move in any direction, but if they coordinate on choosing the same side, they can 'push' it in a certain direction. If $\text{Edges} = \{\text{edge}^1, \ldots, \text{edge}^4\}$ are the grid edges, the closest edge to the object at time t is given by $\text{edge}_t^* = \arg\min_{\text{edge} \in \text{Edges}} \Big(\min(dist_t^h(\phi_t, \text{edge}), dist_t^v(\phi_t, \text{edge}))\Big)$. Agents are rewarded for how close they are to the object, and for how close the object is to the boundary of the grid, i.e. they must coordinate on which side of the object from which to 'push' it, to ensure it moves to the grid's boundary. The

reward function is given by

$$R(s_t^i, a_t^i, \hat{\mu}_t) = \Big(D_{\max}^{ag} - [dist_t^h(\phi_t, x_t^i) + dist_t^v(\phi_t, x_t^i)]\Big) + \Big(D_{\max}^{box} - [dist_t^h(\phi_t, \text{edge}_t^*) + dist_t^v(\phi_t, \text{edge}_t^*)]\Big),$$

where $D_{\max}^{ag} = 2(L-1)$ and $D_{\max}^{box} = \lfloor (L-1)/2 \rfloor$ are the maximum possible agent-to-box and box-to-closest-edge distances on the $L \times L$ grid, and R is then normalised by $D_{\max}^{ag} - 1 + D_{\max}^{box}$ to lie in $[0, 1]$.

In this game the Pareto-dominant joint policies actively push the box toward one of the four edges (by symmetry these are equivalent): agents gather on one side of the box and follow it as it moves, keeping orientation so that the probability field continues to favour motion toward the chosen edge. Under any discount factor $\gamma < 1$ this strictly improves cumulative reward over passive policies that let the box random-walk to a wall, since it gets the wall-distance term in R to its maximum sooner. Once the box reaches a wall it is trapped on the perimeter thereafter, so the wall-distance term remains at its maximum. The Pareto-dominant MFG-NE thus form a continuum: the box can be at any perimeter cell, and the agents at any configuration of cells next to it, with all such configurations achieving the same per-agent maximum reward. Under stochastic dynamics the box typically drifts along the edge into one of the four corners, so the practical long-run resting positions are the corners. These perimeter NE Pareto-dominate sub-optimal NE in which agents are distributed symmetrically around the box (no net push): the box then performs a symmetric random walk and reaches the wall only by chance, giving strictly lower per-agent discounted reward than the active-push policies.

The above tasks are all coordination tasks, in that agents receive higher returns by aligning their policies and hence have an inherent incentive to communicate their policy parameters, even though the MFG framework is non-cooperative. Our fifth game is a non-coordination task: the reward function is not designed to give higher returns for more aligned policies. We include this game to demonstrate that even in

such non-cooperative scenarios our networked architecture receives higher returns than both independent and central-agent alternatives, such that decentralised selfish agents may still have incentive to communicate. The fifth game is:

Disperse. This is similar to the 'exploration' tasks in Laurière et al. [20], Wu et al. [153] and other MFG works. In our version agents are rewarded for being located in more sparsely populated areas *but only if they are stationary*, to avoid trivial random policies. The reward function is given by $R(s_t^i, a_t^i, \hat{\mu}_t) = r_{stationary}(-\hat{\mu}_t(s_t^i))$, where

$$r_{stationary}(x) = \begin{cases} x & \text{if } a_t^i \text{ is 'remain stationary'} \\ -1 & \text{otherwise.} \end{cases}$$

In this game we expect the algorithm to drive the population toward a uniform stationary distribution. The per-state reward $-\log \hat{\mu}(s)$ is concave in $\hat{\mu}(s)$ and so penalises any concentration, giving a unique Pareto-dominant MFG-NE: the uniform stationary distribution. Sub-optimal behaviour corresponds to agents continuing to move (paying the -1 penalty from $r_{stationary}$) or clumping in dense regions.

Following from Ch. 4, in these spatial environments, both the communication network $\mathcal{G}_t^{comm}$ and the visibility graph $\mathcal{G}_t^{vis}$ are determined by the physical distance from agent i. We show plots for various radii, expressed as fractions of the maximum possible distance (the grid's diagonal length). We set $C_p = C_e = 1$ to show the benefit to learning speed brought by even a *single* communication round. Note that the networked population with the largest radius is always fully connected, and therefore these agents are always able to accurately estimate $\hat{\mu}_t$ even for $C_e = 0$. That is, their observations are equivalent to those that the central-agent population would receive, albeit that policies are updated and spread differently.

Experiments were conducted on a Linux-based machine with 2 x Intel Xeon Gold 6248 CPUs (40 physical cores, 80 threads total, 55 MiB L3 cache). We use the JAX framework to accelerate and vectorise our code. Random seeds are set in our code in a fixed way dependent on the trial number to allow easy replication of experiments.

5.7.2 Experimental metrics

To give as informative results as possible about both performance and proximity to a MFG-NE, we provide two metrics for each experiment (the same as the first two metrics in Ch. 4). Both metrics are plotted with mean and standard deviation, computed over ten trials (each with a random seed) of the system run in each setting.

5.7.2.1 Exploitability

In Ch. 4 we described the exploitability metric, which is the one MFGs works most commonly use to evaluate how close a given policy π is to a NE policy π^* [19, 20, 100, 153, 200, 259]. Our discussion of exploitability here is very similar to that in Sec. 4.7.2.1, but we give it again since some of the definitions and details differ slightly in the context of the present chapter.

Exploitability usually assumes that all agents are following the same policy π, and quantifies how much an agent can benefit by deviating from π, by measuring the difference between the return given by π and that of a BR policy with respect to the distribution generated by π:

Definition 5.7.1 (Exploitability of π)**.** The exploitability $\mathcal{E}_{\text{expl}}$ of policy π is given by:

$$\mathcal{E}_{\text{expl}}(\pi) = V(BR(I(\pi)), I(\pi)) - V(\pi, I(\pi)).$$

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 $\pi = \pi^*$. Prior works conducting empirical testing have generally focused on the central-agent setting, so this classical definition, as well as most evaluations, only consider exploitability when all agents are following a single policy π_k. However, as we noted in Ch. 4, both purely independent populations and networked populations may have divergent policies $\pi_k^i \neq \pi_k^j$ for some $i, j \in \{1, \ldots, N\}$. We are therefore interested in the 'exploitability' of the population's joint policy $\boldsymbol{\pi} := (\pi^1, \ldots, \pi^N) \in \Pi^N$.

Since we do not have access to the exact BR policy as in some related works [20, 153], we must instead approximate the exploitability, similarly to Ch. 4 and Perrin et al. [107]. We freeze the policy of all agents apart from a deviating agent, for which we store its current policy and then conduct 50 k loops of policy improvement. To approximate the expectations in Def. 5.7.1, we take the best return of the deviating agent across 10 additional k loops, as well as the mean of all the other agents' returns across these same 10 loops. (While the policies of all non-deviating agents is π_k in the central-agent case, if the non-deviating agents do not share a single policy, then this method is approximating the exploitability of their joint policy $\boldsymbol{\pi}_k^{-d}$, where d is the deviating agent.) We then revert the deviating agent back to its stored policy, before learning continues for all agents as per the main algorithm. Due to the expensive computations required for this metric, we evaluate it every second k iteration of the main algorithm for Figs. 5.1, 5.3, 5.2, 5.4 and 5.5, and every fourth iteration for the population-dependent experiments.

As in Ch. 4, *the exploitability metric has a number of limitations in our setting.* In coordination games (the setting for all tasks apart from the 'disperse' game), agents benefit by following the same behaviour as others, and so a deviating agent generally stands to gain less from switching to a BR policy than it might in the non-coordination games on which many other works focus. For example, the return of a best-responding agent in the 'cluster' task still depends on the extent to which other agents coordinate on where to cluster, meaning it cannot significantly increase its return by deviating from a badly clustering policy. This means that the downward trajectory of the exploitability metric is less clear in our plots than in other works that do not focus on coordination games. This is likely why the difference between the approximated exploitability of the independent agents and the other populations is clearer in our non-coordination 'disperse' task in Fig. 5.5 than in the other tasks.

Moreover, our approximation takes place via MOMD policy improvement steps (as in the main algorithm) for an independent, deviating agent while the policies of the rest of the population are frozen. As such, the quality of our approximation is limited by the number of policy-improvement/expectation-estimation rounds,

which must be restricted for the sake of the running speed of the experiments. Furthermore, since one of the findings of our paper is that independent-learning agents increase their returns significantly slower (if at all) than networked or central-agent populations, it is arguably unsurprising that approximating the BR by an independently deviating agent sometimes gives an unclear and noisy metric. This includes the exploitability going below zero, which should not be possible if the policies and distributions are computed exactly. Given the limitations presented by approximating exploitability, we also provide a second metric to indicate the progress of learning, as in Ch. 4.

5.7.2.2 Average discounted return

Similarly to Ch. 4, we record the average discounted return of the agents' policies π_k^i during the M iterations. This allows us to compare how quickly agents are learning to increase their returns, even when exploitability gives us limited ability to distinguish between the desirability of the MFG-NEs to which populations converge. We can observe that settings that converge to similar exploitability values may not have similar average agent returns, suggesting that some algorithms are better than others not just at finding NE, but more Pareto-efficient NE (for reasons related to why networked populations can increase their returns faster, as per Sec. 5.6, similar to those discussed in Sec. 4.7.4.2). See for example Figs. 5.1 and 5.2, where the networked populations converge to similar exploitability as the independent and central-agent populations, but receive higher average returns.

5.7.3 Hyperparameters

See Table 5.1 for our hyperparameter choices. We can broadly group our hyperparameters into those controlling the size of the experiment, those controlling the size of the Q-network, those controlling the number of iterations of each loop in the algorithms and those affecting the learning/policy updates or policy adoption.

As in Ch. 4, in our experiments we generally want to demonstrate that our communication-based algorithms outperform the central-agent and independent

architectures by allowing policies that are estimated to be better performing to proliferate through the population, such that convergence occurs within fewer iterations and computationally faster, even when the Q-function is poorly approximated and/or the mean-field is poorly estimated, as is likely to be the case in real-world scenarios. As such, we generally select hyperparameters at the lower end of those we tested during development, to show that our algorithms are particularly successful given what might otherwise be considered 'undesirable' hyperparameter choices. Moreover we want to show that there is a benefit even to a small amount of communication, so that communication rounds themselves do not excessively add to time complexity. Thus we set $C_p = C_e = 1$ to show the benefits to convergence brought by even a *single* communication round.

Table 5.1: Hyperparameters

Hyper-param.	Value	Comment
Trials	10	We run 10 trials with different random seeds for each experiment. We plot the mean and standard deviation for each metric across the trials.
Gridsize	10×10/ 50×50/ 100×100	Experiments on large state spaces are run on 50×50 and 100×100 grids (Figs. 5.1-5.5), while experiments with population-dependent policies are run on the 10×10 grid (Figs. 5.6a, 5.7a, 5.6b and 5.7b).
Population size	500	We chose 500 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, while also being feasible to simulate wrt. time and computation constraints [105, 108, 126, 142, 153, 201, 229–232].
Number of neurons in input layer	cf. comment	The agent's position is represented by two concatenated one-hot vectors indicating the agent's row and column. An additional two such vectors are added for the shark's/object's position in the 'evade' and 'push object' tasks. For population-dependent policies, the mean-field distribution is a flattened vector of the same size as the grid. As such, the input size in the 'evade' and 'push object' tasks is $[(4 \times \text{dimension}) + (\text{dimension}^2)]$; in the other settings it is $[2 \times \text{dimension}]$.

Continued on next page

Table 5.1: Hyperparameters (continued)

Hyper-param.	Value	Comment
Neurons per hidden layer	cf. comment	We draw inspiration from common rules of thumb when selecting the number of neurons in hidden layers, e.g. it should be between the number of input neurons and output neurons / it should be 2/3 the size of the input layer plus the size of the output layer / it should be a power of 2 for computational efficiency. Using these rules of thumb as rough heuristics, we select the number of neurons per hidden layer by rounding the size of the input layer down to the nearest power of 2. The layers are all fully connected.
Hidden layers	2	We experimented with 2 and 3 hidden layers in the Q-networks. While 3 hidden layers gave similar or slightly better performance, we selected 2 for increased computational speed for conducting our experiments.
Activation function	ReLU	This is a common choice in deep RL.
K	100	K is chosen to be large enough to see the average return converging.
M	50	We tested M in {50,100} and found that the lower value was sufficient to achieve convergence while minimising training time. It may be possible to converge with even smaller choices of M.
L	50	We tested L in {50,100} and found that the lower value was sufficient to achieve convergence while minimising training time. It may be possible to converge with even smaller choices of L.
E	20	We tested E in {20,50,100}, 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.
C_p	1	As in Ch. 4, we choose this value to show the convergence benefits brought by even a single communication round, even in networks that may have limited connectivity; higher choices are likely to give even better performance.
C_e	1	Similar to C_p, we choose this value to show the ability of our algorithm to appropriately estimate the mean field even with only a single communication round, even in networks that may have limited connectivity.
γ	0.9	Standard choice across RL literature.
τ_q	0.03	We tested τ_q in {0.01,0.02,0.03,0.04,0.05}, as well as linearly decreasing τ_q from $0.05 \rightarrow 0$ as k increases. However, only 0.03 gave stable increase in return. Note that this is the value also chosen in Vieillard et al. [260].

Continued on next page

Table 5.1: Hyperparameters (continued)

Hyper-param.	Value	Comment
$\|B\|$	32	This is a common choice of batch size that trades off noisy updates and computational efficiency.
cl	-1	We use the same value as in Vieillard et al. [260].
ν	$L-1$	We tested ν in $\{1, 4, 20, L-1\}$. We found that in our setting, updating $\theta' \leftarrow \theta$ once per k iteration s.t. $\theta'_{k+1,l} = \theta_{k,l}\ \forall l$ gave sufficient learning that was similar to the other potential choices of ν, so we do this for simplicity, rather than arbitrarily choosing a frequency to update θ' during each k loop. Setting the target to be the policy from the previous iteration is similar to the method in Laurière et al. [20]. Whilst Wu et al. [153] updates the target within the L loops for stability, we do not find this to be a problem in our experiments.
Optimiser	Adam	As in Vieillard et al. [260], we use the Adam optimiser with initial learning rate 0.01.
τ_k^{comm}	cf. comment	τ_k^{comm} increases linearly from 0.001 to 1 across the K iterations. This is a simplification of the inverse annealing scheme used in Ch. 4. Further optimising the inverse annealing process may lead to better results.

5.7.4 Results and discussion

5.7.4.1 Population-independent policies in large state-spaces

Figs. 5.1-5.5 illustrate that introducing function approximation to algorithms in this setting allows them to converge within a practical number of iterations ($k \ll 100$), even for large state spaces (100×100 and 50×50 grids). By contrast, the tabular algorithms in Ch. 4 appear only just to converge by $k = 200$ for the same tasks for the larger of the two grids, which is only 16×16.

In Figs. 5.1-5.3, no populations appear to have significantly different exploitability to each other, while in Fig. 5.4 the central-agent population may have lower exploitability, but not significantly so. As discussed in Sec. 5.7.2.1, the exploitability metric is noisy and provides limited information in these coordination games. Nevertheless we can see that in all four plots the independent agents hardly improve their returns at all, while the central-agent populations hardly improve their returns in the 'target agreement' games in Figs. 5.1 and 5.2. There is therefore little a

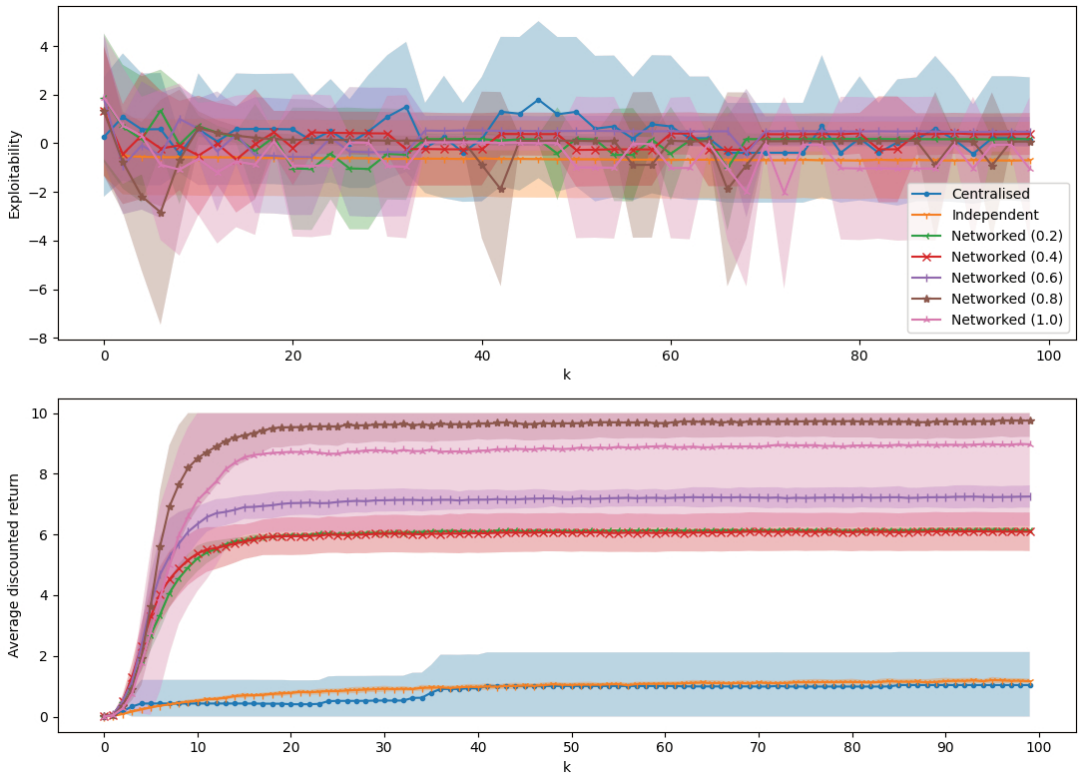


Figure 5.1: 'Target agreement', population-independent, 100 × 100 grid. The networked populations of all broadcast radii significantly outperform the central-agent and independent populations in terms of average return, where the latter two cases hardly appear to learn at all.

deviating agent can do to increase its return in these coordination games, meaning exploitability appears low, despite these being undesirable equilibria.

Meanwhile, the networked agents do learn to improve their returns and therefore significantly outperform the stagnant independent agents in Figs. 5.1-5.4 and the stagnant central-agent populations in Figs. 5.1 and 5.2, indicating that our communication scheme helps agents to reach substantially 'preferable' (i.e. Pareto-dominant) equilibria (for reasons related to why networked populations can increase their returns faster as per Sec. 5.6). While the central-agent populations do appear to increase their returns in the 'cluster' tasks in Figs. 5.3 and 5.4, they appear to do so more slowly and reaching a lower final value than all networked populations in the 100 × 100 grid case, and than all networked populations apart from those with the smallest broadcast radii in the 50 × 50 grid case. Indeed in the 100 × 100 grids in Figs. 5.1 and 5.3, the central-agent populations appear to perform less well than

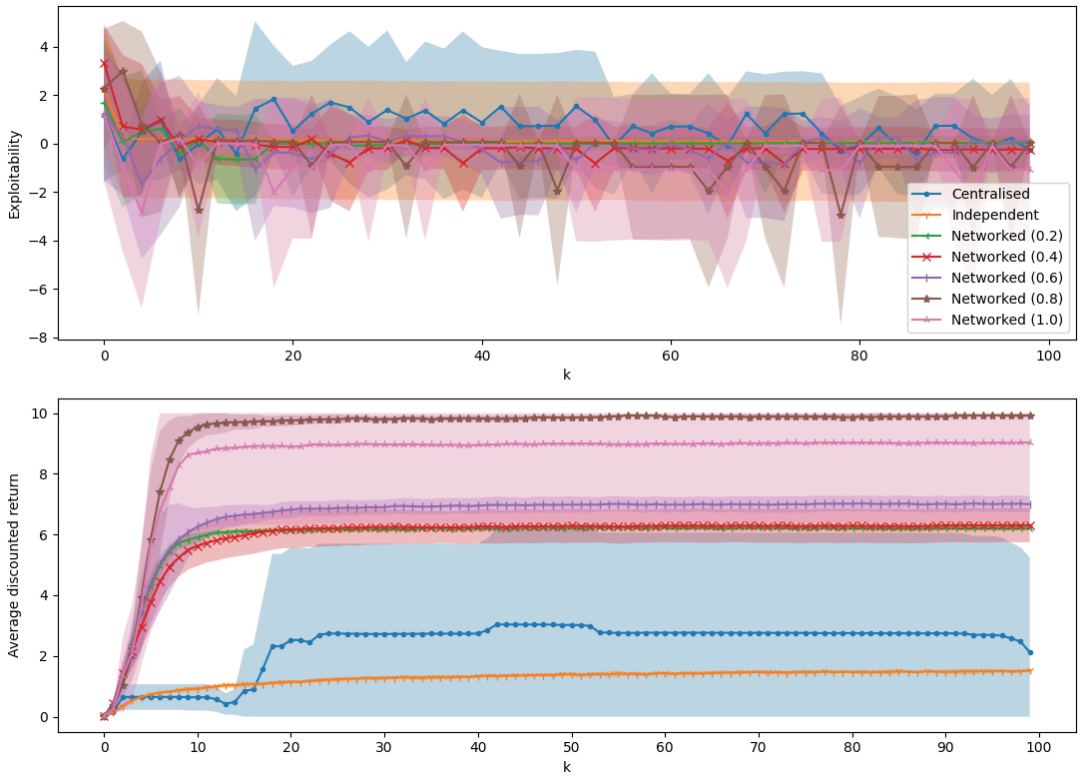


Figure 5.2: 'Target agreement' task, population-independent policies, 50×50 grid. The networked populations of all broadcast radii significantly outperform the central-agent and independent populations in terms of average return, where the latter two cases hardly appear to learn at all.

they do in the 50×50 grids in Figs. 5.2 and 5.4, whereas the networked populations do not suffer a performance decrease, indicating that our networked communication scheme scales better and is robust to larger state spaces than the central-agent paradigm. Similarly, in the 'target agreement' and 'cluster' tasks in the tabular setting in Ch. 4, the central-agent populations generally perform similarly to or outperform the networked populations, indicating that the networked architecture is more robust than the central-agent alternative when moving to non-tabular settings.

In the non-coordination 'disperse' task in Fig. 5.5 below, networked agents significantly outperform both independent and central-agent populations in terms of average return. They also significantly outperform independent agents in terms of exploitability, and perhaps also central-agent populations though not significantly so. The fact that this happens in this non-coordination, non-cooperative game shows that agents may have an incentive to communicate with each other even if

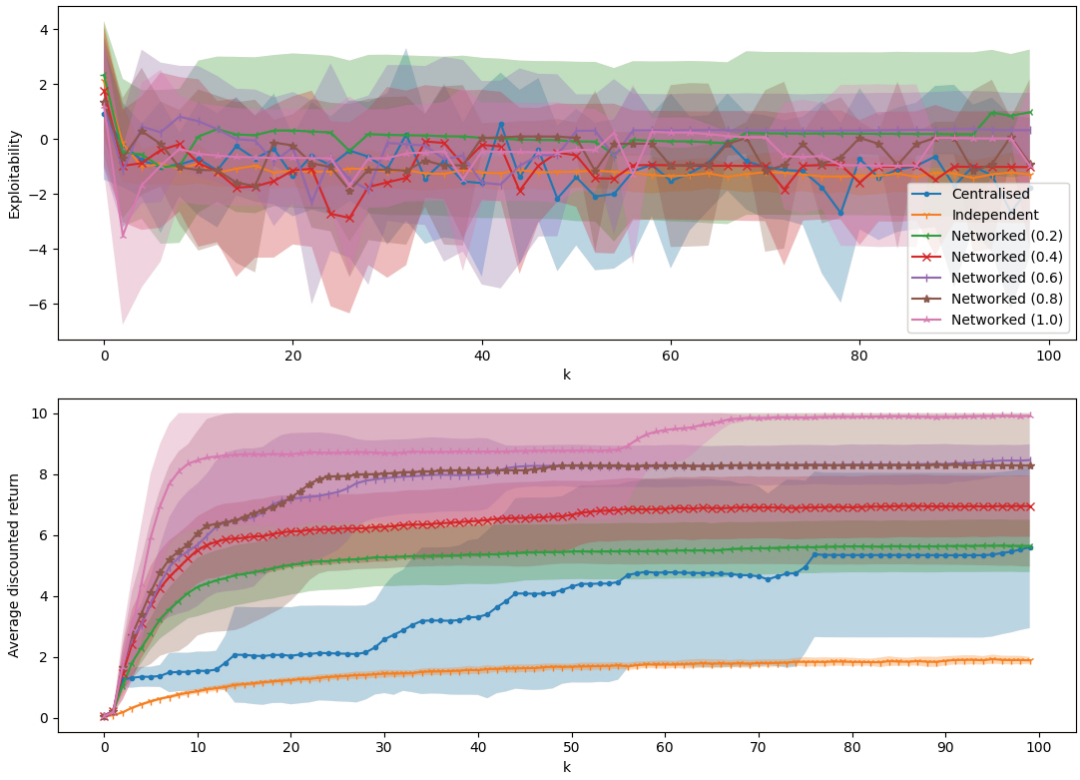


Figure 5.3: 'Cluster', population-independent, 100×100 grid. The networked populations of all broadcast radii outperform the central-agent and independent populations in terms of average return; independent agents hardly appear to learn at all.

they are self-interested. Indeed the agents learning independently do not appear to improve their returns at all, despite this being the paradigm that in some senses might be expected to perform best in a non-coordination, non-cooperative setting.

5.7.4.2 Population-dependent policies in complex environments

We also demonstrate the ability of our algorithms to handle more complex tasks, using population-dependent policies and estimated mean-field observations.

Figs. 5.6a and 5.7a, where agents estimate the mean-field distribution via Alg. 5, differ minimally from Figs. 5.6b and 5.7b, where agents directly receive the global mean-field distribution. This indicates that our estimation algorithm allows agents to appropriately estimate the distribution, even with only one round of communication for agents to help each other improve their local counts. Only in the 'push object' task in Fig. 5.6a, and there only with the smaller broadcast radii,

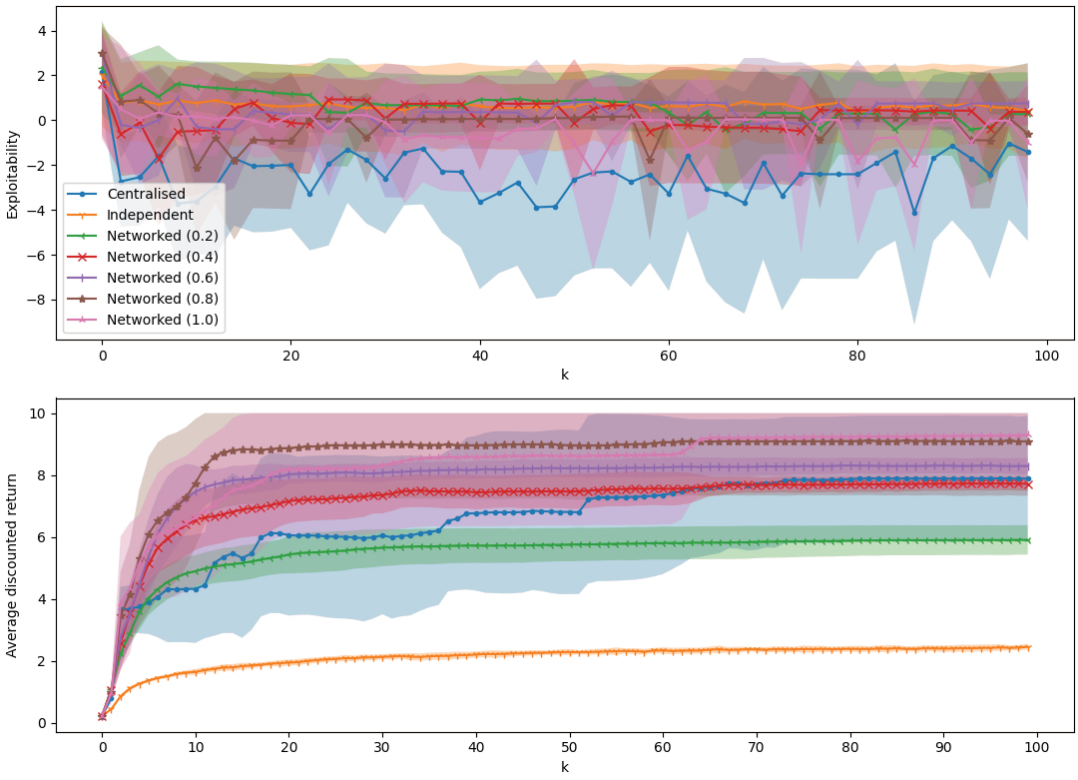


Figure 5.4: 'Cluster' task, population-independent policies, 50×50 grid. The networked agents of all broadcast radii significantly outperform the independent agents in terms of average return, where the latter case hardly appears to learn at all. The higher broadcast radii also appear to outperform the central-agent case in terms of average return, with the latter outperforming all others in terms of exploitability, but perhaps not significantly so.

do networked agents slightly underperform the returns of equivalent agents in the

global observability case in Fig. 5.6b, as might be expected.

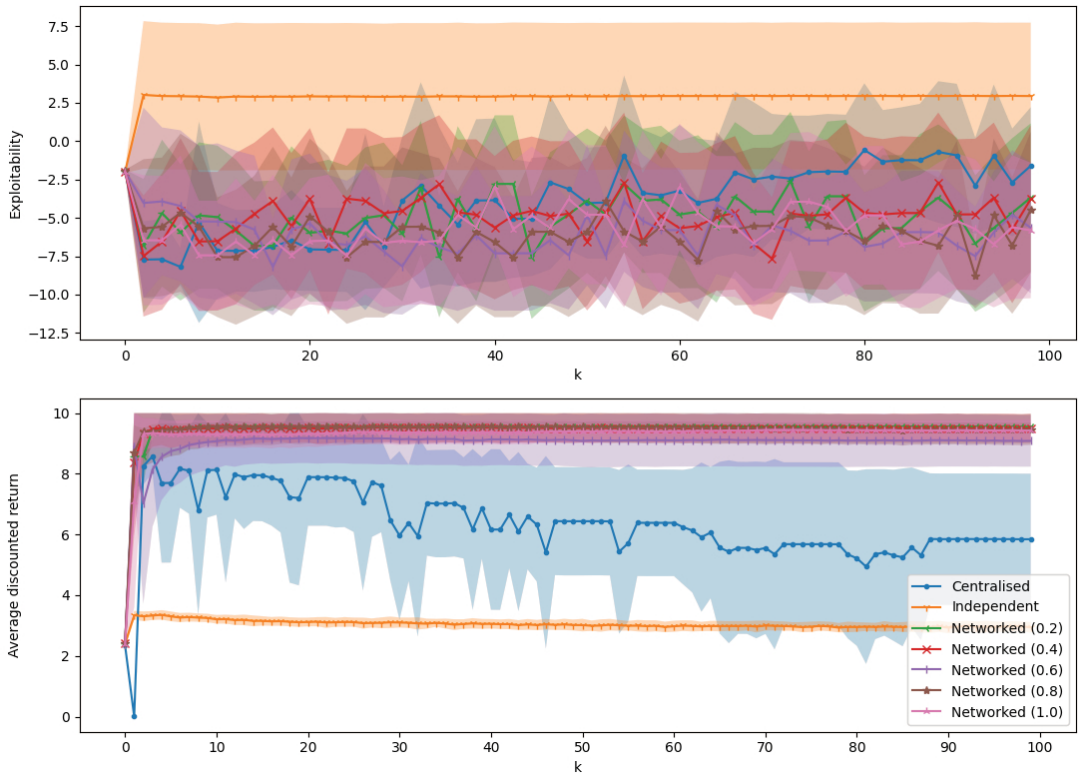
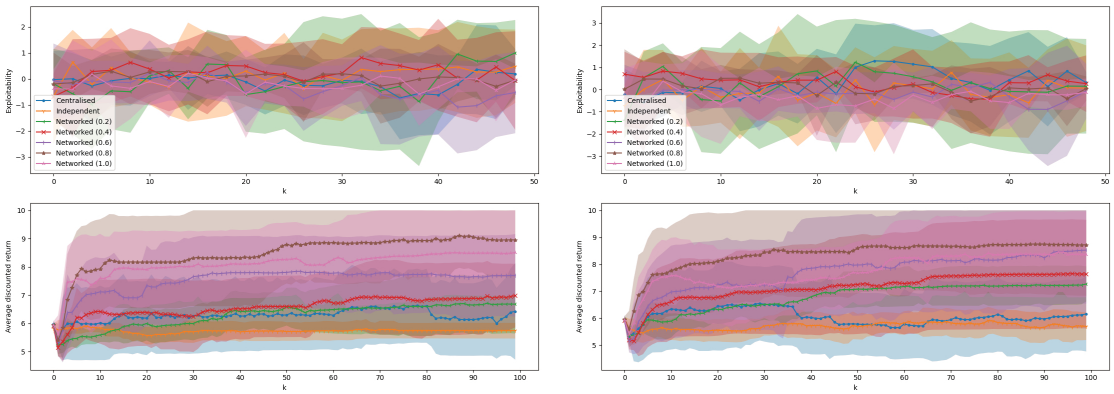


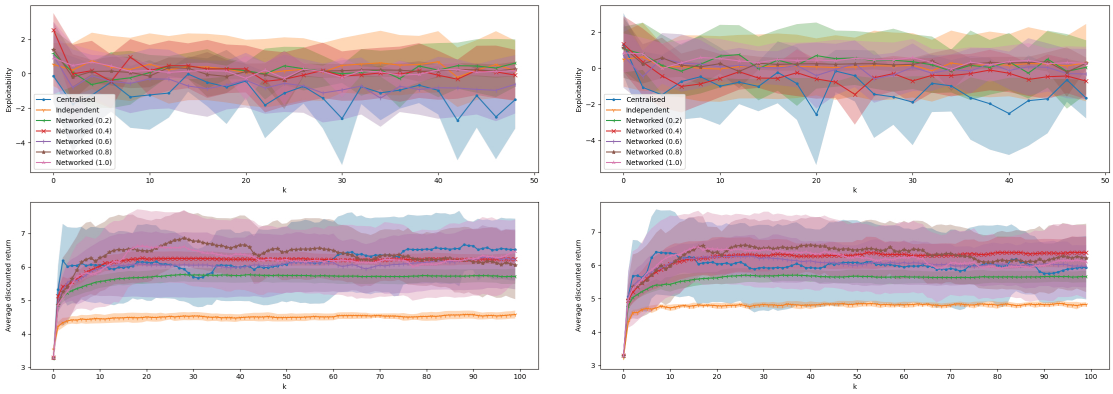
Figure 5.5: 'Disperse' task, population-independent policies, 100 × 100 grid. The networked populations of all broadcast radii significantly outperform the central-agent and independent populations in terms of average return (and exploitability in the case of independent agents), with independent agents not learning at all.



(a) Estimated mean-field distribution.

(b) Global observability of mean field.

Figure 5.6: 'Push object' task, population-dependent policies on a 10 × 10 grid. The networked populations of all broadcast radii appear to outperform the central-agent and independent populations in terms of average return; the latter two cases hardly appear to learn at all.



(a) Estimated mean-field distribution. **(b)** Global observability of mean field.

Figure 5.7: 'Evade' task, population-dependent policies on a 10×10 grid. The networked agents of all broadcast radii significantly outperform the independent agents in terms of average return, and perform similarly to the central-agent populations.

For the reasons given in Sec. 5.7.2.1 regarding coordination games, the exploitability metric gives limited information in the 'push object' and 'evade' tasks in Figs. 5.6 and 5.7: for example, the return of a best-responding agent in the 'push object' task still depends on the extent to which other agents coordinate on which direction in which to push the box, meaning it cannot significantly increase its return by deviating. However, all of the networked cases significantly outperform the independent learners in terms of the average return to which they converge in both tasks. In the 'push object' task networked learners also appear to outperform central-agent populations, while in the 'evade' task all networked cases perform similarly to the central-agent case. Recall though that in the real world a central-agent architecture is a strong assumption, a computational bottleneck and single point of failure.

5.8 Conclusion

We novelly contributed function approximation to the setting of solving MFGs online from the empirical distribution, and also contributed two novel algorithms for locally estimating the empirical mean field for population-dependent policies. We have justified theoretically why our networked communication algorithm is able to learn faster even than the central-agent architecture in this function approximation

setting, and demonstrated empirically the ability of our algorithms to handle large state spaces and population-dependent policies, and also to estimate the mean field in non-stationary games. For discussion of potential avenues for future work, please see Ch. 7.

6

Networked Communication in Mean-Field Control

Contents

6.1 Introduction

We begin by recapping the content and contributions of this chapter, first stated in Sec. 1.2.3.

MFC has generally received less attention than MFGs, and no work on MFC has met all of our criteria for practical mean-field algorithms. Recent works in MFC give decentralised but model-based training [13], or are model-free but require centralised, episodic training [7]. Building on our communication scheme and learning algorithm for MFGs, we now introduce networked communication to the cooperative MFC setting, where populations conceptually have even more incentive to communicate. This allows us to present a model-free deep learning algorithm that fulfils all of our proposed desiderata, including learning online from a single non-episodic run of the empirical system, and decentralised training without the restrictive need to observe global information as in prior works. In addition to incorporating our existing sub-routine for estimating the global mean field from Alg. 5 in Ch. 5, in this chapter we contribute a novel sub-routine Alg. 6 for estimating the global average reward from local communication for the MFC setting.

Our previous theoretical analysis of networked communication in the non-cooperative MFG setting does not extend trivially to MFC. Therefore we contribute new theoretical proofs showing that proliferating high-performing policies through the population via decentralised communication allows networked populations to learn faster than both the independent *and* the central-agent alternatives in the MFC setting, where now we conduct the theoretical analysis separately for the coordination and anti-coordination classes of cooperative game. We also demonstrate this finding experimentally in numerous games, as well as contributing an empirical study of the algorithms' robustness to communication failures, along with several ablation studies.

In summary, our contributions include:

- We provide the first algorithms in MFC for model-free training without any central coordination or provision of information, as well as the first MFC algorithms for online learning from a single, non-episodic run of the empirical system.
  - We contribute a novel sub-routine allowing decentralised agents to estimate the global average reward via networked communication, and incorporate our existing sub-routine used in MFGs for estimating the global mean field aided by local communication.
- We prove theoretically that in this context, decentralised networked communication can improve learning speed over the independent *and* central-agent architectures.
- We provide extensive experiments supporting our theoretical results in numerous games, and give ablation studies of various parts of our algorithms, as well as a study of robustness to communication failures.

The rest of this chapter is structured as follows. We provide further comparison with related work in Sec. 6.2, give preliminaries in Sec. 6.3, and our algorithms in Sec. 6.4. We present theoretical results in Sec. 6.5 and experiments in Sec. 6.6.

6.2 Related work

We discuss here the research most closely related to this chapter, focusing on decentralisation and networked communication, and clarifying the differences with prior methods and settings. Please see Ch. 2 for work more generally related to networked communication in the mean-field framework.

Similar to the dynamic we identify for MFGs in Ch. 2, numerous works claiming to study decentralisation in MFC take this to mean only that agents do not have access to the specific states of all other agents, and have policies depending on their local state and possibly the mean field, all of which we take as a given in our work. They nevertheless rely on a central learner or coordinator that provides global

information to all agents, a dependence that we remove in our work. This applies, for example, to Grammatico et al. [141], where a 'central population coordinator' broadcasts a common signal to all agents, and to Tajeddini et al. [264], which presents a leader-follower setting where a virtual 'central population coordinator' estimates the mean-field trajectory of the whole population in place of an empirical population. Farzaneh et al. [265] similarly requires a central coordinator, and also presents a non-cooperative scenario so does not actually fall under MFC despite being referred to as such.

Cui et al. [7] presents a model-free deep learning algorithm that gives decentralised execution but requires centralised, episodic training (Cui et al. [126] also offers a centralised-training decentralised-execution method). They suggest that decentralised training could be achieved if all agents can directly observe the global mean-field distribution and use the same seed to correlate their actions (though they only provide empirical results for the centralised scenario) whilst we do not require either assumption for our decentralised training algorithm. They also train episodically whereas we learn online from a single run of the system. Finally, their experiments focus only on coordination games, whereas we additionally explore empirical effects resulting from decentralised training in anti-coordination games, where populations can gain higher rewards by diversifying their behaviour.

Bayraktar and Kara [13] considers independent, 'online' learning for MFC in a setting that is different from ours. Crucially, their method involves agents first estimating a model (reward and transition functions) of the system by conducting 'online' updates using samples collected while following exploration policies. Only once having done so do they compute execution policies that are optimal with respect to the estimated model. We argue that having a dedicated exploration phase is infeasible for many real-world applications, and instead present a fully model-free online learning algorithm. Moreover, their setting only permits independent learning if N is large but finite. For infinite populations, a central coordinator is required to supply common noise to aid exploration during the initial phase, and if the optimal policy for the estimated model is not unique, centralised coordination is

required to allow the agents to agree on which policy to execute. Our networked algorithm requires no such special considerations. Finally, their work is purely theoretical, whereas we provide extensive empirical results.

Angiuli et al. [109] and Angiuli et al. [125] provide algorithms for MFC learning from a single run, but there it is a single run only of a 'representative' player that is used to simulate the mean field, rather than a single run of the empirical population as in our work. Their algorithms are thus inherently centralised, as well as involving two timescales for updating the mean-field approximation, which we argue is unlikely to be a practical paradigm for training in complex real-world systems such as robotic swarms.

6.3 Preliminaries

6.3.1 Mean-field control

We use the notation introduced in Sec 3.1, as well as the following. For agent $i \in \{1 \dots N\}$, i's policy $\pi^i \in \Pi$ depends on its observation o_t^i. We restate from Sec. 5.3 the different forms that this observation can take, and relatedly a more formal definition of the policy, after the following.

Definition 6.3.1 (N-player stochastic cooperative control problem with symmetric, anonymous agents)**.** Similar to the games in Defs. 4.3.1 and 5.3.1, this 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 $\{s_0^i\}_{i=1}^N \in \mathcal{S}^N$ sampled from some initial distribution $\mu_0 \in \Delta_{\mathcal{S}}$, 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, both identical to all agents, and which map each agent's local state and action and the population's empirical distribution to transition probabilities and bounded rewards, respectively, i.e. $\forall i \in \{1, \dots, N\}$: $s_{t+1}^i \sim P(\cdot | s_t^i, a_t^i, \hat{\mu}_t)$ and $r_t^i = R(s_t^i, a_t^i, \hat{\mu}_t)$.

For the joint policy $\boldsymbol{\pi} := (\pi^1, \dots, \pi^N) \in \Pi^N$, an individual agent's discounted return is given by:

Definition 6.3.2 (Individual expected discounted return)**.** For all $i, j \in \{1, \dots, N\}$, i's return is

$$V^i(\boldsymbol{\pi}, \mu_{\bar{t}}) = \mathbb{E}\left[\sum_{t=\bar{t}}^{\infty} \gamma^t R(s_t^i, a_t^i, \hat{\mu}_t) \middle|_{\substack{s_{\bar{t}}^j \sim \mu_{\bar{t}} \\ a_t^j \sim \pi^j(o_t^j) \\ s_{t+1}^j \sim P(\cdot|s_t^j, a_t^j, \hat{\mu}_t)}}\right].$$

However, the maximisation objective for this *cooperative* problem is:

Definition 6.3.3 (Population-average expected discounted return)**.** For all $i, j \in \{1, \dots, N\}$ the average return is

$$V^{pop}(\boldsymbol{\pi}, \mu_{\bar{t}}) = \frac{1}{N}\sum_{i}^{N} V^i(\boldsymbol{\pi}, \mu_{\bar{t}}) = \mathbb{E}\left[\frac{1}{N}\sum_{t=\bar{t}}^{\infty}\sum_{i}^{N} \gamma^t R(s_t^i, a_t^i, \hat{\mu}_t) \middle|_{\substack{s_{\bar{t}}^j \sim \mu_{\bar{t}} \\ a_t^j \sim \pi^j(o_t^j) \\ s_{t+1}^j \sim P(\cdot|s_t^j, a_t^j, \hat{\mu}_t)}}\right].$$

That is, the solution to the control problem is $\boldsymbol{\pi}^* = \arg\max_{\boldsymbol{\pi} \in \Pi^N} V^{pop}(\boldsymbol{\pi}, \mu_{\bar{t}})$.

At the limit as $N \to \infty$, the infinite population of agents can be characterised as a limit distribution $\mu \in \Delta_{\mathcal{S}}$; the infinite-agent setting is termed a MFC problem. As in Ch. 5, the *mean-field flow* $\boldsymbol{\mu}$ is given by the infinite sequence of mean-field distributions s.t. $\boldsymbol{\mu} = (\mu_t)_{t \geq 0}$.

Definition 6.3.4 (Induced mean-field flow)**.** This is restated from Def. 5.3.2. We denote by $I(\pi)$ the mean-field flow $\boldsymbol{\mu}$ induced when all the agents follow π, where this is generated from π by

$$\mu_{t+1}(s') = \sum_{s,a} \mu_t(s)\pi(a|o_t)P(s'|s, a, \mu_t).$$

The snapshot of this induced flow at t is given by $I(\pi)_t$.

Definition 6.3.5 (Social welfare)**.** When all agents follow policy π giving mean-field flow $\boldsymbol{\mu} = I(\pi)$, π's social welfare is

$$W(\pi; I(\pi)) = \mathbb{E}\left[\sum_{t=\bar{t}}^{\infty} \gamma^t (R(s_t, a_t, I(\pi)_t)) \middle|_{\substack{s_{\bar{t}} \sim \mu_{\bar{t}} \\ a_t \sim \pi(\cdot|o_t) \\ s_{t+1} \sim P(\cdot|s_t, a_t, I(\pi)_t)}}\right].$$

Definition 6.3.6 (Social optimum)**.** The solution to the MFC problem is a social optimum policy $\pi^* \in \Pi$ that maximises the social welfare function in Def. 6.3.5, i.e. $\pi^* = \arg\max_{\pi \in \Pi} W(\pi; I(\pi))$.

Remark 6.3.7. Previous works showed that the MFC social optimum π^* gives a good approximate solution to the harder-to-solve finite-agent problem (i.e. if $\boldsymbol{\pi} = (\pi^*, \dots, \pi^*)$), with the error characterised by $\mathcal{O}(\frac{1}{\sqrt{N}})$ [4–7, 13].

As in previous chapters, when the distribution is the same for all t, i.e. $\mu_t = \mu_{t+1}$ $\forall t \geq 0$, we say the mean-field flow is stationary, giving a stationary MFC problem. Non-stationary problems may require the policy to depend on the mean field such that $o_t^i = (s_t^i, \hat{\mu}_t)$, whereas the observation in the stationary case can be simplified to $o_t^i = s_t^i$. However, since classical approaches to the MFC problem often conceive of a central planner trying to guide the population to a distribution that maximises the expected return, they sometimes have policies that depend on the mean field even in the stationary case [7, 100, 266]. Therefore *we permit mean field-dependent policies for the sake of generality, but show through our ablation studies that in practice our algorithms require only* $\pi^i(a|o_t^i) = \pi^i(a|s_t^i)$ in our experimental tasks, which have stationary solutions.

Furthermore, as we argued in Ch. 5, it is unrealistic to assume that decentralised agents with a possibly limited communication radius would have perfect observability of the global mean field $\hat{\mu}_t$. Therefore we allow agents to form a local estimate $\tilde{\hat{\mu}}_t^i$ which can be improved by communication with neighbours, using Alg. 9 (taken from Alg. 5 in Ch. 5 for the MFG setting). We thus have $o_t^i = (s_t^i, \tilde{\hat{\mu}}_t^i)$. Formally we can now say that when $o_t^i = (s_t^i, \hat{\mu}_t)$ or $(s_t^i, \tilde{\hat{\mu}}_t^i)$, we have the set of policies defined as $\Pi = \{\pi : \mathcal{S} \times \Delta_{\mathcal{S}} \to \Delta_{\mathcal{A}}\}$, and the set of Q-functions denoted $\mathcal{Q} = \{q : \mathcal{S} \times \Delta_{\mathcal{S}} \times \mathcal{A} \to \mathbb{R}\}$. (N.b. when $o_t^i = s_t^i$, we instead have $\Pi = \{\pi : \mathcal{S} \to \Delta_{\mathcal{A}}\}$ and $\mathcal{Q} = \{q : \mathcal{S} \times \mathcal{A} \to \mathbb{R}\}$.)

6.4 Learning and estimation algorithms

In Ch. 5 we gave algorithms for the MFG setting, where networked communication is used 1) to form local estimates of the global empirical mean field, and 2) to allow agents to adopt better-performing policies from neighbours to accelerate learning. We adapt these algorithms for cooperative MFC, where decentralised agents must optimise the population-average return instead of their individual one

Algorithm 6 Average reward estimation and communication

Require: Time-dependent communication graph $\mathcal{G}_t^{comm}$, rewards $\{r_t^i\}_{i=1}^N$, number of communication rounds C_r

1: $\forall i$: Initialise reward sets $\hat{\mathcal{R}}_{t,1}^i \leftarrow \{(ID^i, r_t^i)\}$
2: **for** c_r in $1, \ldots, C_r$ **do**
3: $\quad \forall i$: Broadcast $\hat{\mathcal{R}}_{t,c_r}^i$
4: $\quad \forall i : J_t^i \leftarrow \{j \in \mathcal{N} : (i,j) \in \mathcal{E}_t^{comm}\}$
5: $\quad \forall i : \hat{\mathcal{R}}_{t,(c_r+1)}^i \leftarrow \hat{\mathcal{R}}_{t,c_r}^i \cup \bigcup_{j \in J_t^i} \hat{\mathcal{R}}_{t,c_r}^j$
6: **end for**
7: $\forall i : \tilde{\hat{r}}_t^i \leftarrow \frac{1}{|\hat{\mathcal{R}}_{t,C_r}^i|} \sum_{(ID,r) \in \hat{\mathcal{R}}_{t,C_r}^i} r$
8: **return** Estimates of average reward $\left\{\tilde{\hat{r}}_t^i\right\}_{i=1}^N$

(the decentralised agents may not always follow a common policy while training unless we make assumptions on the communication network as discussed in Sec. 6.5, so we do not directly optimise social welfare from Def. 6.3.5).

It is unrealistic to assume that decentralised agents have access to the global average reward, so we find a third use of the communication network in 3) allowing agents to estimate the global average reward $\hat{r}_t$ from a local neighbourhood. We contribute a novel sub-routine Alg. 6 for this purpose (Sec. 6.4.1), and we describe our main learning method Alg. 7 in Sec. 6.4.2. Our policy communication algorithm Alg. 8, based on that in Ch. 5 for the MFG setting, is described in Sec. 6.4.3 - for clarity we now break this off into a separate sub-routine from the main learning algorithm. Meanwhile Alg. 9 for estimating the mean field, which is the same as Alg. 5 in Ch. 5 for the MFG setting, is restated for ease of reference in Sec. 6.4.4.

6.4.1 Sub-routine for networked estimation of global average reward

Our novel Alg. 6 involves agents using the communication network $\mathcal{G}_t^{comm}$ to locally estimate the global population-average reward received after a given step in the environment. Maximising the population-average reward ensures agents are solving the cooperative MFC problem instead of the non-cooperative MFG. Agents broadcast their received reward with a unique ID to ensure each reward is only counted once (Line 1). They add those received from neighbours to their collection,

Algorithm 7 Decentralised MFC learning from non-episodic system run

Require: loop parameters $K, M, L, E, C_e, C_r, C_p$, learning parameters $\gamma, \tau_q, |B|, cl, \nu$, $\{\tau_k^{comm}\}_{k\in\{0,\dots,K-1\}}$

Require: initial states $\{s_0^i\}_{i=1}^N$; $t \leftarrow 0$

1: $\forall i$: Randomly initialise parameters θ_0^i of Q-networks $\check{Q}_{\theta_0^i}(o,\cdot)$, and set $\pi_0^i(a|o) = \text{softmax}\Big(\frac{1}{\tau_q}\check{Q}_{\theta_0^i}(o,\cdot)\Big)(a)$ and $\check{Q}_{\theta_{0,0}^{i,\prime}} \leftarrow \check{Q}_{\theta_0^i}(o,\cdot)$

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

3: $\quad\forall i$: Empty i's buffer

4: $\quad$**for** $m \in 0,\dots,M-1$ **do**

5: $\quad\quad\{o_t^i\}_{i=1}^N \leftarrow$ **EstimateMeanFieldAlg. 9**$\Big(\mathcal{G}_t^{vis}, \mathcal{G}_t^{comm}, \{s_t^i\}_{i=1}^N\Big)$

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

7: $\quad\quad\{\tilde{\hat{r}}_t^i\}_{i=1}^N \leftarrow$ **EstimateAverageRewardAlg. 6**$\Big(\mathcal{G}_t^{comm}, \{r_t^i\}_{i=1}^N\Big)$

8: $\quad\quad\forall i$: Add $\Big(o_t^i, a_t^i, \tilde{\hat{r}}_t^i, o_{t+1}^i\Big)$ to i's buffer

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

10: $\quad$**for** $l \in 0,\dots,L-1$ **do**

11: $\quad\quad\forall i$: Sample batch $B_{k,l}^i$ from i's buffer

12: $\quad\quad$Update θ to minimise $\hat{\mathcal{L}}(\theta,\theta')$ as in Def. 6.4.1

13: $\quad\quad$If $l \mod \nu = 0$, set $\theta' \leftarrow \theta$

14: $\quad$**end for**

15: $\quad\check{Q}_{\theta_{k+1}^i}(o,\cdot) \leftarrow \check{Q}_{\theta_{k,L}^i}(o,\cdot)$

16: $\quad\forall i$: $\pi_{k+1}^i(a|o) \leftarrow \text{softmax}\Big(\frac{1}{\tau_q}\check{Q}_{\theta_{k+1}^i}(o,\cdot)\Big)(a)$

17: $\quad\Big(\{\pi_{k+1}^i\}_i, \{s_t^i\}_i, t\Big) \leftarrow$ **CommunicatePolicyAlg. 8**$\Big(\mathcal{G}_t^{comm}, \{\pi_{k+1}^i\}_i, \{s_t^i\}_i, t\Big)$

18: **end for**

19: **return** policies $\{\pi_K^i\}_{i=1}^N$

and repeat the process of broadcasting and expanding their collections for a further $C_r - 1$ rounds, so as to receive rewards from agents more than one hop away on the network (Lines 2-6). They finally set their estimate of the global average to the average of the rewards they have collected (Line 7).

6.4.2 Main learning algorithm for updating Q-networks and policies

In Ch. 5 we solved MFGs online from non-episodic runs of the finite-population empirical system using a form of policy iteration called MOMD, introduced in Secs. 5.3.2 and 5.4.1. We now adapt our non-cooperative Alg. 3 from Ch. 5, to learn a social optimum in the MFC setting via our novel Alg. 7, which differs

by its use of the (estimated) average reward.

Our MOMD-based method works as follows. Each agent i approximates its Q-function $\check{Q}_{\theta_k^i}(o,\cdot)$ with its own neural network parametrised by θ_k^i. Agent i's policy is determined by

$$\pi_{\theta_k^i}(a|o) = \text{softmax}\left(\frac{1}{\tau_q}\check{Q}_{\theta_k^i}(o,\cdot)\right)(a)\,.$$

We denote this as $\pi_k^i(a|o)$ for simplicity when appropriate. Each agent maintains a buffer (with size M) of collected transitions of the form $\left(o_t^i, a_t^i, \tilde{\hat{r}}_t^i, o_{t+1}^i\right)$, where $\tilde{\hat{r}}_t^i$ is i's local estimate of the global average reward obtained by running Alg. 6 (Line 7). At each iteration k, agents empty their buffer (Line 3) before collecting M new transitions in the environment (Lines 4-9). Each decentralised agent then trains its Q-network $\check{Q}_{\theta_k^i}$ via L updates (Lines 10-14) as follows.

For stability, i also maintains a target network $\check{Q}_{\theta_{k,l}^{i,\prime}}$ with the same architecture but parameters $\theta_{k,l}^{i,\prime}$ copied from $\theta_{k,l}^i$ less regularly than $\theta_{k,l}^i$ themselves are updated, i.e. only every ν learning iterations (Line 13). At each iteration l, the agent samples a random batch $B_{k,l}^i$ of $|B|$ transitions from its buffer (Line 11). It then trains its Q-network using stochastic gradient descent to minimise the loss in Def 6.4.1 below (Line 12). The trained Q-network determines i's updated policy (Line 16).

Definition 6.4.1 (Q-network empirical loss)**.** The training loss to be minimised is given by

$$\hat{\mathcal{L}}(\theta,\theta') = \frac{1}{|B|}\sum_{transition\in B_{k,l}^i}\left|\check{Q}_{\theta_{k,l}^i}(o_t,a_t) - T\right|^2,$$

where the target T is given by

$$\begin{aligned} T = \tilde{\hat{r}}_t &+ \left[\tau_q \ln \pi_{\theta_{k,l}^{i,\prime}}(a_t|o_t)\right]_{cl}^0 \\ &+ \gamma\sum_{a\in\mathcal{A}}\pi_{\theta_{k,l}^{i,\prime}}(a|o_{t+1})\left(\check{Q}_{\theta_{k,l}^{i,\prime}}(o_{t+1},a) - \tau_q\ln\pi_{\theta_{k,l}^{i,\prime}}(a|o_{t+1})\right). \end{aligned}$$

As in Ch. 5, for $cl<0$, $[\cdot]_{cl}^0$ is a clipping function used in Munchausen RL to prevent numerical issues if the policy is too close to deterministic, as the log-policy term is otherwise unbounded [153, 260].

Algorithm 8 Policy communication and selection

Require: Time-dependent communication graph $\mathcal{G}_t^{comm}$, loop parameters E, C_p, learning parameters γ, $\{\tau_k^{comm}\}_{k\in\{0,\dots,K-1\}}$
Require: policies $\{\pi_{k+1}^i\}_{i=1}^N$; states $\{s_t^i\}_{i=1}^N$; t
1: $\forall i : \sigma_{k+1}^i \leftarrow 0$
2: **for** $e \in 0, \dots, E-1$ evaluation steps **do**
3: $\{o_t^i\}_{i=1}^N \leftarrow$ **EstimateMeanFieldAlg. 9**$\left(\mathcal{G}_t^{vis}, \mathcal{G}_t^{comm}, \{s_t^i\}_{i=1}^N\right)$
4: Take step $\forall i : a_t^i \sim \pi_k^i(\cdot|o_t^i), r_t^i = R(s_t^i, a_t^i, \hat{\mu}_t), s_{t+1}^i \sim P(\cdot|s_t^i, a_t^i, \hat{\mu}_t)$
5: $\forall i : \sigma_{k+1}^i \leftarrow \sigma_{k+1}^i + \gamma^e \cdot r_t^i$
6: $t \leftarrow t+1$
7: **end for**
8: **for** C_p rounds **do**
9: $\forall i$: Broadcast $\sigma_{k+1}^i, \pi_{k+1}^i$
10: $\forall i : J_t^i \leftarrow i \cup \{j \in \mathcal{N} : (i,j) \in \mathcal{E}_t^{comm}\}$
11: $\forall i$: Select $\text{adopted}^i \sim \Pr\left(\text{adopted}^i = j\right) = \frac{\exp(\sigma_{k+1}^j/\tau_k^{comm})}{\sum_{x\in J_t^i}\exp(\sigma_{k+1}^x/\tau_k^{comm})}\ \forall j \in J_t^i$
12: $\forall i : \sigma_{k+1}^i \leftarrow \sigma_{k+1}^{\text{adopted}^i}, \pi_{k+1}^i \leftarrow \pi_{k+1}^{\text{adopted}^i}$
13: $\{o_t^i\}_{i=1}^N \leftarrow$ **EstimateMeanFieldAlg. 9**$\left(\mathcal{G}_t^{vis}, \mathcal{G}_t^{comm}, \{s_t^i\}_{i=1}^N\right)$
14: Take step $\forall i : a_t^i \sim \pi_k^i(\cdot|o_t^i), r_t^i = R(s_t^i, a_t^i, \hat{\mu}_t), s_{t+1}^i \sim P(\cdot|s_t^i, a_t^i, \hat{\mu}_t)$; $t \leftarrow t+1$
15: **end for**
16: **return** (policies $\{\pi_{k+1}^i\}_{i=1}^N$, states $\{s_t^i\}_{i=1}^N$, t)

6.4.3 Sub-routine for communicating and refining policies

Alg. 8 (based on Lines 15-27 of our Alg. 3 in Ch. 5 for MFGs) uses the communication network $\mathcal{G}_t^{comm}$ to spread policy updates that are estimated to be better performing through the population, allowing faster learning than in the independent and central-agent cases.

Alg. 8 is run after agents have independently updated their policies according to their newly trained Q-networks at each iteration k of the main learning algorithm (Line 17, Alg. 7). In Alg. 8, agents obtain an approximation of their *individual* expected discounted return $\{V^i(\boldsymbol{\pi}, \mu_t)\}_{i=1}^N$ (Def. 6.3.2), i.e. *not* the population-average return, which would not give differentiation between the different updated policies - we discuss this further in Rem. 6.5.1 below. They do so by collecting individual rewards for E steps (not added to the training buffer), and calculating the discounted sum of rewards over these finite steps, setting this value to σ_{k+1}^i (Lines 1-7). We can characterise this approximation of the infinite-step return

Algorithm 9 Mean-field estimation and communication for environments with $\mathcal{G}_t^{vis}$

Require: Time-dependent visibility graph $\mathcal{G}_t^{vis}$, time-dependent communication graph $\mathcal{G}_t^{comm}$, states $\{s_t^i\}_{i=1}^N$, number of communication rounds C_e

1: $\forall i, s$: Initialise count vector $\hat{v}_{t,1}^i[s]$ with $\emptyset$
2: $\forall i,\ \forall s' \in \mathcal{S}' : (s_t^i, s') \in \mathcal{E}_t^{vis}\ :\ \hat{v}_{t,1}^i[s'] \leftarrow \sum_{j \in \{1,\dots,N\}: s_t^j = s'} 1$
3: **for** $c_e \in 1, \dots, C_e$ **do**
4: $\quad \forall i$: Broadcast $\hat{v}_{t,c_e}^i$
5: $\quad \forall i : J_t^i \leftarrow i \cup \{j \in \mathcal{N} : (i,j) \in \mathcal{E}_t^{comm}\}$
6: $\quad \forall i, s$: Initialise new count vector $\hat{v}_{t,(c_e+1)}^i[s]$ with $\emptyset$
7: $\quad \forall i, s$ and $\forall j \in J_t^i : \hat{v}_{t,(c_e+1)}^i[s] \leftarrow \hat{v}_{t,c_e}^j[s]$ if $\hat{v}_{t,c_e}^j[s] \neq \emptyset$
8: **end for**
9: $\forall i : counted\_agents_t^i \leftarrow \sum_{s \in \mathcal{S}: \hat{v}_t^i[s] \neq \emptyset} \hat{v}_t^i[s]$
10: $\forall i : uncounted\_agents_t^i \leftarrow N - counted\_agents_t^i$
11: $\forall i : unseen\_states_t^i \leftarrow \sum_{s \in \mathcal{S}: \hat{v}_t^i[s] = \emptyset} 1$
12: $\forall i, s$ where $\hat{v}_t^i[s]$ is not $\emptyset$: $\tilde{\hat{\mu}}_t^i[s] \leftarrow \frac{\hat{v}_t^i[s]}{N}$
13: $\forall i, s$ where $\hat{v}_t^i[s]$ is $\emptyset$: $\tilde{\hat{\mu}}_t^i[s] \leftarrow \frac{uncounted\_agents_t^i}{N \times unseen\_states_t^i}$
14: **return** $\{$(states s_t^i, mean-field estimates $\tilde{\hat{\mu}}_t^i$)$\}_{i=1}^N$

as $\{\sigma_{k+1}^i\}_{i=1}^N = \{\hat{V}^i(\boldsymbol{\pi}_{k+1}, \mu_t; E)\}_{i=1}^N$.

They then broadcast their Q-network parameters along with σ_{k+1}^i (Line 9). Receiving these from their neighbours J_t^i on the network, agents select which set of parameters to adopt by taking a softmax over their own and the received estimate values $\sigma_{k+1}^j\ \forall j \in J_t^i$, defined as follows (Lines 10-12):

$$\text{adopted}^i \sim \Pr\left(\text{adopted}^i = j\right) = \frac{\exp\left(\sigma_{k+1}^j / \tau_k^{comm}\right)}{\sum_{x \in J_t^i} \exp\left(\sigma_{k+1}^x / \tau_k^{comm}\right)}.$$

They repeat this broadcast and adoption process for C_p rounds (distinct from the C_r/C_e communication rounds for the other sub-routines).

6.4.4 Sub-routine for networked estimation of global empirical mean field

Networked agents use Alg. 9 (this is the same as Alg. 5 from Ch. 5 for the MFG setting) to locally estimate the global empirical mean field, to serve as an observation input for their Q-/policy-networks. Recall that we include this added observation and sub-routine for generality, especially for non-stationary problems. However it is often not necessary, particularly in stationary problems like those in

our experiments, where agents can find the social optimum while only observing $o_t^i = s_t^i$, and therefore would not need to estimate the mean field.

We restate how Alg. 9 works based on Sec. 5.5.2. This sub-routine involves agents using the visibility graph $\mathcal{G}_t^{vis}$ to count the number of agents in locations that fall within the visibility radius (Line 2). (In Ch. 5 we also discuss an alternative algorithm for more general settings where the visibility graph $\mathcal{G}_t^{vis}$ does not apply, which could equally be used in the present chapter if desired.) For C_e communication rounds, agents can supplement this local count with those received from neighbours over the communication network $\mathcal{G}_t^{comm}$, in order to count agents that do not fall within the visibility radius (Lines 3-8). We assume agents know the population's total size N, and therefore can distribute the uncounted agents uniformly over the states that remain unaccounted for after the communication rounds (Lines 9-11). Agents now have a vector containing a true or estimated count for every state; this is converted to an estimated empirical mean field by dividing all counts by N (Lines 12-13).

6.5 Theoretical results

6.5.1 Introduction

We follow the definitions of the central-agent and independent-learning baseline architectures from Chs. 4, 5 and Yardim et al. [15], which solve MFGs online from a non-episodic run of the empirical system. Both alternative architectures can each be seen as special cases of our networked algorithm:

- In the **central-agent** case, only arbitrary central agent $i = 1$ updates a Q-network and automatically pushes this to all other agents in place of the decentralised policy communication in Line 17 of Alg. 7. Additionally, the true global mean-field distribution and average reward are always used in place of the local estimates, i.e. $\tilde{\hat{\mu}}_t^i = \hat{\mu}_t$ and $\tilde{\hat{r}}_t^i = \hat{r}_t$.

- In the **independent** case, there are never any links in $\mathcal{G}_t^{comm}$ or $\mathcal{G}_t^{vis}$, i.e. $\mathcal{E}_t^{comm} = \mathcal{E}_t^{vis} = \emptyset$.

All expectations in this section are taken jointly over:

- the initial joint state $\{s_0^i\}_{i=1}^N$ sampled from μ_0;
- the stochastic transitions, actions, and rewards collected by each agent into its individual buffer;
- the random mini-batch samples $B_{k,l}^i$ used to minimise the loss in Def. 6.4.1;
- the random initialisation of the Q-network parameters θ_0^i in Line 1 of Alg. 7;
- the softmax draws used during policy adoption in Line 11 of Alg. 8.

As in Ch. 5, the communication and visibility networks $\mathcal{G}_t^{comm}, \mathcal{G}_t^{vis}$ are supplied as exogenous inputs at each round and may or may not depend on agents' positions: in our experiments they are defined by communication and visibility radii, which makes them deterministic conditional on positions and stochastic only through the random state evolution, but the theory accepts any sequence of graphs satisfying the relevant structural conditions. We do not consider explicit random graph models in our analysis.

We prove theoretically that our policy communication and adoption scheme allows networked agents to increase their returns faster than these alternatives (with the central-agent paradigm being potentially unrealistic and vulnerable in any case). Rem. 6.5.1 suggests informal reasons for our formal results to aid intuitive understanding.

Remark 6.5.1. Like many cooperative learning paradigms, the central-agent alternative to our networked architecture may suffer from the credit-assignment problem, in that it is not clear how learners' local state s_t^i and local action a_t^i contributed to the (locally estimated) *average* reward $\tilde{\hat{r}}_t^i$ [267, 268]. Agents may receive low individual reward r_t^i by taking action a_t^i given o_t^i, but would nevertheless learn that doing so was 'good' if the rest of the population took highly rewarded actions at the same step giving high average reward $\tilde{\hat{r}}_t^i$. By drawing spurious relations, an agent's updated policy $\pi_{k+1}^i(a|o)$ may negatively impact (or simply not

advance) the goal of maximising social welfare, which is problematic if such a policy is automatically pushed from the central learner to the rest of the population. (A similar argument could also be applied to independent agents, were it not for the fact that realistically they only have access to their own rewards in any case, though we given an ablation of this in Fig. 6.8 which shows that our logic still applies.) Including the (estimated) empirical mean field in the observation $o_t^i = (s_t^i, \tilde{\hat{\mu}}_t^i)$ might mitigate this slightly by indicating which mean fields gave high average rewards. However, this does not solve the issue of allowing learners to distinguish between helpful or unhelpful local actions a_t^i, whether those learners are centralised or not, since actions can affect rewards in ways other than simply by helping to reach a certain mean field. By updating policies with respect to average return but then spreading updates through the population which are estimated to give a higher *individual* return, despite this being a cooperative problem, we reduce the credit-assignment problem by replicating updated policies that should contribute positively to the population-average return, and filtering out those that do not.

Moreover, even if we assumed credit assignment were not a problem, there is randomness in the Q-network update: agents have stochastic policies and thus may collect a wide variety of transitions to add to their individual buffers, from which they sample randomly when training their Q-networks. There may therefore be considerable variance in the quality of their estimated Q-functions, leading in turn to variance in the quality of policy updates. Similar to the analysis in Sec. 5.6, at each iteration of the central-agent algorithm, in *expectation* the central learner will by definition have an average-quality update, and its updated policy will be pushed to the entire population whether or not it performs well. Our decentralised networked approach permits beneficial parallelisation in place of this single-learner method, by generating a whole population of possible updates, from which the one(s) estimated to be best-performing can be selected via a process akin to the comparison of fitness functions in evolutionary algorithms. These are then spread around the population, biasing networked populations towards better performing updates.

We give the theoretical analysis separately for two important subclasses of cooperative game usually found in MFC, which have different reward structures and therefore can incentivise different population behaviour:

1. *coordination games*, where the social welfare is increased by agents aligning their strategies, such as in consensus/synchronisation/rendezvous tasks;

2. *anti-coordination games*, where the social welfare is increased by the population exhibiting diverse strategies, such as in exploration, coverage or task allocation games.

These subclasses cover a large proportion of cooperative objectives in symmetric, anonymous settings with large populations. We emphasise that the fact that agents would in principle benefit from having diverse policies in anti-coordination games does not contradict the classical MFC framework that simplifies the infinite population problem by finding the single policy to be shared by all agents. In the symmetric (i.e. identical reward and transition functions) MFC limit, an optimal solution can be realised by having the infinite agents all follow the single socially optimal policy, even for reward functions that favour diversity. A very large number of works on both MFC and MFGs conduct experiments on anti-coordination games, particularly dispersal and exploration tasks, despite assuming that the population follows a shared single policy learnt by a central node [100, 124, 127]. We make the distinction between coordination and anti-coordination games to aid theoretical analysis of our decentralised policy adoption scheme compared with entirely independent learning: while it is intuitive that adopting independently-updated policies from neighbours via the communication scheme could be beneficial in coordination games, we also show theoretically and empirically that the adoption scheme provides a benefit in anti-coordination games, though this requires separate analysis.

To define the two types of game, we first introduce the following functions. $\mathbb{I}[\cdot]$ is the indicator function, which equals 1 if the condition inside is true and 0 otherwise. $b : \Pi \to \mathbb{R}_{\geq 0}$ is a *base return function* that quantifies a policy's inherent ability

to receive rewards regardless of how many other agents follow the same strategy. For example, if agents are rewarded for agreeing on one of a number of targets at which to meet, then policies that visit none of the designated targets will have lower returns than those that do, whether agents are aligned or not. Finally, $f_c : \mathbb{N} \to \mathbb{R}_{>0}$ (resp. $f_d : \mathbb{N} \to \mathbb{R}_{>0}$) is a *coordination (resp. anti-coordination) scaling function*. It has minimum $f_c(1) > 0$ (resp. $f_d(0) > 0$), and increases monotonically with the number of agents whose policies match (resp. are different from) i's.

Definition 6.5.2 (Coordination game)**.** The agents' return can be decomposed as follows, $\forall i, j \in \{1, \dots, N\}$: $V^i(\boldsymbol{\pi}, \mu_{\bar{t}}) = h\left(b(\pi^i), f_c\left(\sum_{j\in\{1,\dots,N\}} \mathbb{I}\left[\pi^i = \pi^j\right]\right)\right)$, where $h : \mathbb{R}_{\geq 0} \times \mathbb{R}_{>0} \to \mathbb{R}_{\geq 0}$ is a function that composes $b(\cdot)$ and $f_c(\cdot)$ and is monotonic in both arguments, i.e. an increase in either the policy's intrinsic ability to attain rewards, or the extent to which it is aligned with other agents' policies, gives a higher return.

Definition 6.5.3 (Anti-coordination game)**.** The agents' return can be decomposed as follows, $\forall i, j \in \{1, \dots, N\}$: $V^i(\boldsymbol{\pi}, \mu_{\bar{t}}) = h\left(b(\pi^i), f_d\left(N - \sum_{j\in\{1,\dots,N\}} \mathbb{I}\left[\pi^i = \pi^j\right]\right)\right)$, where $h : \mathbb{R}_{\geq 0} \times \mathbb{R}_{>0} \to \mathbb{R}_{\geq 0}$ is a function that composes $b(\cdot)$ and $f_d(\cdot)$ and is monotonic in both arguments, i.e. an increase in either the policy's intrinsic ability to attain rewards, or the extent to which it is different from other agents' policies, gives a higher return.

Note that in our setting, where policy parameters are directly communicated and adopted among the population, we focus on exact equality of policies for simplicity of the theory. However, these definitions could be made more general and inclusive by instead considering similarity kernels or label mappings of strategically relevant parts of policies.

6.5.2 Analysis

6.5.2.1 Networked vs central-agent populations

For simplicity of the theory, we make several assumptions. We explore the conditions under which these assumptions apply in practice, and discuss how even when

loosening the assumptions, they still provide useful heuristic insight as to how our networked communication scheme affords benefits over the central-agent and independent-learning architectures. We do not enforce the assumptions in our experiments, and our empirical results nevertheless follow our theoretical theorems in all but some specific instances that we discuss.

Our sub-routines involve time-varying networks sharing different types of information at different points in the algorithm, meaning that theoretical analysis can potentially grow complicated. We seek to simplify this analysis to make it more intuitive and useful by focusing on the benefit of the decentralised policy exchange scheme in Alg. 8. This is because our ablation studies of Algs. 6 (average reward estimation) and 9 (mean field estimation) in Sec. 6.6.3 indicate that the policy exchange scheme is the dominant factor in driving the benefit of the networked paradigm in our experimental settings. Moreover, recall that Alg. 9 is only necessary when we allow population-dependent policies such that $o_t^i = (s_t^i, \tilde{\hat{\mu}}_t^i)$, whereas for stationary problems, including all those in our experiments and many others, using a mean field observation or estimation is not actually required for finding the optimal policy.

Therefore, similar to Assumption 5.6.1 in the previous chapter, the first assumption presumes that it is only the decentralised policy communication scheme that creates a difference in learning between the networked and central-agent cases, by assuming that the estimated mean fields and average rewards are equivalent to the true ones used in the central-agent case. Note that populations with fully connected networks will in any case always be able to accurately estimate $\hat{r}_t$ and $\hat{\mu}_t$ by Algs. 6 and 9, even for $C_r = 1$ and $C_e = 0$. This is may apply reasonably commonly in practice depending on the scenario; for example, if the network is defined by a broadcast radius (as in our experiments), then the network will be fully connected whenever that radius is at least large enough to cover the area that all the agents fall within. We leave analysis of the theoretical impact of worsening mean-field and average-reward estimates for future work.

Assumption 6.5.4. Assume that Algs. 6 and 9 allow networked agents to obtain accurate estimates of the true population-average rewards and global empirical mean field respectively, i.e. $\forall i\ \tilde{\hat{\mu}}_t^i = \hat{\mu}_t$ and $\tilde{\hat{r}}_t^i = \hat{r}_t$.

Now recall that at each iteration k of Alg. 7, after individually updating their policies in Line 16, the population has the policies $\{\pi_{k+1}^i\}_{i=1}^N$. There is randomness in these individual policy updates, stemming from the random sampling of each agent's individually collected buffer. In Lines 1-7 of Alg. 8, agents estimate the individual infinite-step discounted returns $\{V^i(\boldsymbol{\pi}, \mu_0)\}_{i=1}^N$ (Def. 6.3.2) of their updated policies by computing $\{\sigma_{k+1}^i\}_{i=1}^N$: the E-step discounted return with respect to the empirical mean field generated when agents follow policies $\{\pi_{k+1}^i\}_{i=1}^N$.

We next assume that the populations' policies are all pair-wise distinct after the updates in Line 16 and before the policy communication. This ensures that policies that are estimated to receive higher returns (and are thus adopted) are being evaluated as higher-performing due to receiving higher base returns, rather than simply because of how aligned or distinct they already happen to be with regard to other policies. This avoids scenarios where, for example, significantly suboptimal policies that are shared across multiple agents after the update (in the case of a coordination game) end up spreading through the population by communication at the expense of a more promising but less common policy, decelerating rather than accelerating improvement. In practice, this assumption is highly likely to apply in most situations in any case. Even if agents start a given iteration with identical policies, their different random seeds are likely to mean that they collect different sample transitions to add to their reinitialised buffers. Even if their buffers happen to end up containing identical transitions, their different random seeds are likely to mean that they sample differently from their buffers, leading to slightly different updates to their policy networks.

Assumption 6.5.5. Assume that directly after the policy updates in Line 16 (Alg. 7), before any policy transfer as in the networked or central-agent algorithms, all policies are pair-wise distinct due to the randomness in these updates, i.e.

$\forall i, j \in \{1, \ldots, N\}\ \pi^i_{k+1} \neq \pi^j_{k+1}$. This means the function f_c attains its minimum $f_c(1)$, and f_d attains its maximum $f_d(N-1)$.

As in Ch. 5, we now assume that the finite-step estimates of the returns give sufficiently accurate comparisons between policies, so that better policies are indeed the ones that get adopted in expectation.

Assumption 6.5.6. Assume that $\{\sigma^i_{k+1}\}_{i=1}^N$ are sufficiently good estimates so as to respect the ordering of the true infinite discounted individual returns $\{V^i(\boldsymbol{\pi}_{k+1}, \mu_0)\}_{i=1}^N$, i.e.

$$V^i(\boldsymbol{\pi}_{k+1}, \mu_0) > V^j(\boldsymbol{\pi}_{k+1}, \mu_0) \iff \sigma^i_{k+1} > \sigma^j_{k+1} \qquad \forall i, j \in \{1, \ldots, N\}.$$

In practice, even if Assumption 6.5.6 does not strictly hold, the softmax parameter τ^{comm}_k allows a smooth degradation as the ordering of the estimates worsens with respect to the ordering of the true values. That is, if instead of the exact correct policy ordering we have that better policies are simply *more likely* to be given higher estimated evaluations, then the softmax means that these policies remain *more likely* to spread, and a better policy may still be adopted even if it is not evaluated as being better.

As in Ch. 5, the next assumption presumes that the networked population reaches consensus on a single policy within the policy communication rounds of each k iteration. We use this assumption in only one of our three theorems (Thm. 6.5.9), and we do so to give general and intuitive comparison with the central-agent population which always shares a single policy. Incomplete consensus would give different levels of alignment/diversity, such that the relative performance of the central-agent and networked architectures might then depend on the specific reward function of the task, and whether base return or alignment/diversity were more important in that reward function.

Assumption 6.5.7. Assume that after the C_p rounds in Lines 8-15 (Alg. 8), in which agents exchange and adopt policies from neighbours, the networked population is left with a single policy such that $\forall i, j \in \{1, \ldots, N\}\ \pi^i_{k+1} = \pi^j_{k+1}$.

We repeat the following discussion of this assumption from Ch. 5. While this may sound like a strong assumption, we phrase it like this so as not to make overly strong restrictions on the communication network instead - we intentionally leave it so that Assumption 6.5.7 can be fulfilled in numerous ways. Most simply we can think of Assumption 6.5.7 holding if:

1. we set τ_k^{comm} close to 0 for all k, such that the softmax essentially becomes a max function; and

2. the communication network $\mathcal{G}_t^{comm}$ is static and connected during the C_p communication rounds, where C_p is at least as large as the network diameter $d_{\mathcal{G}_t^{comm}}$.

Under these conditions, previous results on max-consensus algorithms show that all agents in the network will converge on the highest value σ_{k+1}^{max} (and hence the associated π_{k+1}^{max}) within a number of rounds equal to the diameter $d_{\mathcal{G}_t^{comm}}$, as we also discuss in Chs. 4 and 5 [255]. If we assumed more strongly that the network was always *fully* connected, policy consensus would be achieved within a single communication round.

Policy consensus can be achieved even outside of these conditions, including if the network is dynamic and not connected at every step. Recall from Sec. 3.2 that a collection of graphs is *jointly connected* if its members' union is connected. Now, instead of assuming that the communication network is static and connected, we assume instead only that the sequence of networks contains one or more sequential jointly connected collections. Then max-consensus is reached within C_p if C_p is large enough that the number of sequential jointly connected collections occurring within C_p is equal to the largest diameter of the union of any such collection.

Thus Assumption 6.5.7 may not hold if C_p is not large enough or if parts of the population remain isolated. However, we do not enforce this assumption in our experiments, where we use $C_p = 1$ to show the benefit of even just one communication round, yet we still see networked populations significantly outperforming central-agent populations across anti-coordination games. In coordination games, while

networked populations that are more connected (due to having larger communication radii) usually perform similarly to or better than central-agent populations, those that are less connected occasionally perform less well than the central-agent populations. This is probably due to Assumption 6.5.7 being empirically more likely to be violated in less connected populations, which in turn is more of an issue in coordination games (where consensus is more likely to be beneficial) than in anti-coordination games (where some lack of consensus does not prevent, or even helps, networked populations to outperform central-agent ones in practice).

The next assumption presumes that if a certain policy, when followed by all members of a finite population, is better than another policy when the latter is followed by all members of a finite population, then the same quality ordering will apply when members of infinite populations follow each policy. We require this in order to relate our analysis of learning in the empirical finite population back to the mean field limit when comparing with central-agent learning. Since the finite population can be arbitrarily large, and in many environments when all agents follow the same policy the finite population-average return will converge smoothly to the infinite population social welfare, this assumption will naturally hold in many scenarios. For example, a policy that is better than another at getting a population of 500 or 5,000,000 agents to cluster in a particular location will also be better than the other policy at getting an infinite population to gather at the location. Nevertheless this order preservation is not a completely general phenomenon, and strict inequalities can vanish or reverse in the limit, especially in models with thresholds or discontinuities in the dependence of rewards or transitions on the mean field, so we state it as an explicit condition in the following.

Assumption 6.5.8. Say we have two different policies that could be shared by the whole population such that $\boldsymbol{\pi}^x = (\pi^x, \ldots, \pi^x)$ and $\boldsymbol{\pi}^y = (\pi^y, \ldots, \pi^y)$. We assume that:

$$V^{pop}(\boldsymbol{\pi}^x, \mu_0) > V^{pop}(\boldsymbol{\pi}^y, \mu_0) \iff W(\pi^x, I(\pi^x)) > W(\pi^y, I(\pi^y)).$$

We have now given all the assumptions for our first theorem. Assumption 6.5.7 assumes that after the C_p policy exchange rounds in Lines 8-15 of Alg. 8, the networked population is left with a single policy. Call this consensus policy π^{net}_{k+1}, and its associated finitely estimated return $\sigma^{\text{net}}_{k+1}$. Recall that the central-agent case is where the Q-network update of arbitrary agent $i = 1$ is automatically pushed to all the others instead of the policy evaluation and exchange in Line 17 of Alg. 7; this is equivalent to a networked case where policy consensus is reached on a *random* one of the policies $\{\pi^i_{k+1}\}^N_{i=1}$. Call this policy *arbitrarily* given to the whole population π^{cent}_{k+1}, and its associated finitely estimated return $\sigma^{\text{cent}}_{k+1}$.

We can now give our first theorem, namely that in expectation networked populations will increase their returns at least as fast as central-agent ones.

Theorem 6.5.9. *Let us set $\tau^{comm}_k \in \mathbb{R}_{>0}$. In coordination and anti-coordination games where Assumptions 6.5.4-6.5.8 apply, we have*

$$\mathbb{E}[W(\pi^{\text{net}}_{k+1}, I(\pi^{\text{net}}_{k+1}))] \geq \mathbb{E}[W(\pi^{\text{cent}}_{k+1}, I(\pi^{\text{cent}}_{k+1}))].$$

Remark 6.5.10. Assumption 6.5.5 presumes that all policies are pairwise distinct after the updates, but does not restrict their returns in the same way. If we additionally make the very weak assumption that at least one of these distinct policies in each k iteration has a base return that is distinct from the others (which is likely to hold in all but the most trivial environments), the inequality in the theorem above will be strict, i.e. *networked learning will always be faster in expectation.*

Proof. Recall that before the communication rounds in Line 8 (Alg. 8), the randomly updated policies $\{\pi^i_{k+1}\}^N_{i=1}$ have associated estimated returns $\{\sigma^i_{k+1}\}^N_{i=1}$. Denote the mean and maximum of this set $\sigma^{\text{mean}}_{k+1}$ and $\sigma^{\text{max}}_{k+1}$ respectively. Since π^{cent}_{k+1} is chosen arbitrarily from $\{\pi^i_{k+1}\}^N_{i=1}$, it will obey $\mathbb{E}[\sigma^{\text{cent}}_{k+1}] = \sigma^{\text{mean}}_{k+1}\ \forall k$, though there will be high variance. Conversely, for the networked case the softmax adoption scheme (Line 11, Alg. 8), which for $\tau^{comm}_k \in \mathbb{R}_{>0}$ gives non-uniform adoption probabilities for distinct σ values, means by definition that some communicated policies are more likely to be adopted than others if they have distinct finitely estimated returns

(those with higher σ^i_{k+1} are more likely to be adopted at each communication round). Thus the consensus π^{net}_{k+1} that gets adopted by the whole networked population will obey $\mathbb{E}[\sigma^{\text{net}}_{k+1}] > \sigma^{\text{mean}}_{k+1}$ if at least one policy receives a distinct return from the others, or $\mathbb{E}[\sigma^{\text{net}}_{k+1}] \geq \sigma^{\text{mean}}_{k+1}$ in the rare circumstance that all policies receive the same return. If τ^{comm}_{k+1} is close to 0, the consensus policy will obey $\mathbb{E}[\sigma^{\text{net}}_{k+1}] = \sigma^{\max}_{k+1}\ \forall k$. As such:

$$\mathbb{E}[\sigma^{\text{net}}_{k+1}] \geq \mathbb{E}[\sigma^{\text{cent}}_{k+1}]. \tag{6.1}$$

In Eq. 6.1 and the remaining equations of the proof, bear in mind that the equality will be strict if at least one policy receives a distinct return from the others.

Refer to the agent whose update originally gave rise to π^{net}_{k+1} and $\sigma^{\text{net}}_{k+1}$ as agent (i, net); we equivalently also have the arbitrary agent (j, cent). Prior to consensus being attained in each case, the joint policy can be written as $\boldsymbol{\pi}^{(i,\text{net};j,\text{cent})}$ $:= (\pi^1, \dots, \pi^{i-1}, \pi^{(i,\text{net})}, \pi^{i+1}, \dots, \pi^{j-1}, \pi^{(j,\text{cent})}, \pi^{j+1}, \dots, \pi^N)$.

Given Eq. 6.1, and by Assumption 6.5.6 on the quality of finite-step estimates, we know that directly after the policy update in Line 16 (Alg. 7), *prior to the consensus being reached*, we have:

$$\mathbb{E}\left[V^{(i,\text{net})}(\boldsymbol{\pi}^{(i,\text{net};j,\text{cent})}_{k+1}, \mu_t)\right] \geq \mathbb{E}\left[V^{(j,\text{cent})}(\boldsymbol{\pi}^{(i,\text{net};j,\text{cent})}_{k+1}, \mu_t)\right]. \tag{6.2}$$

We now need to show that this ordering is maintained in the case that each policy is given to the whole population.

By Assumption 6.5.5 we know that straight after the random policy updates there is no alignment among policies, i.e. in a coordination task we have $f_c^{(i,\text{net})}$ $= f_c^{(j,\text{cent})} = \min f_c$, and in an anti-coordination task we have $f_d^{(i,\text{net})} = f_d^{(j,\text{cent})} =$ $\max f_d$. Therefore if Eq. 6.2 pertains, by Def. 6.5.2 it must be because:

$$\mathbb{E}[b(\pi^{(i,\text{net})})] \geq \mathbb{E}[b(\pi^{(j,\text{cent})})], \tag{6.3}$$

i.e. because the base policy quality is higher for $\pi^{(i,\text{net})}$ than for $\pi^{(j,\text{cent})}$.

By Assumption 6.5.7 on policy consensus, we know that in the networked and central-agent cases the joint policies respectively become $\boldsymbol{\pi}^{\text{net}} := (\pi^{\text{net}}, \pi^{\text{net}}, \pi^{\text{net}}, \dots)$ and $\boldsymbol{\pi}^{\text{cent}} := (\pi^{\text{cent}}, \pi^{\text{cent}}, \pi^{\text{cent}}, \dots)$. We therefore end up with maximum alignment

in both cases, such that $f_c^{\text{net}} = f_c^{\text{cent}} = \max f_c$ in a coordination game, and $f_d^{\text{net}} = f_d^{\text{cent}} = \min f_d$ in an anti-coordination game. Due to this, along with Eqs. 6.2 and 6.3, we know

$$\mathbb{E}\left[V^i(\boldsymbol{\pi}_{k+1}^{\text{net}}, \mu_t)\right] \geq \mathbb{E}\left[V^j(\boldsymbol{\pi}_{k+1}^{\text{cent}}, \mu_t)\right]. \tag{6.4}$$

In turn we have:

$$\mathbb{E}\left[V^{pop}(\boldsymbol{\pi}_{k+1}^{\text{net}}, \mu_t)\right] \geq \mathbb{E}\left[V^{pop}(\boldsymbol{\pi}_{k+1}^{\text{cent}}, \mu_t)\right], \tag{6.5}$$

which by Assumption 6.5.8 gives

$$\mathbb{E}[W(\pi_{k+1}^{\text{net}}, I(\pi_{k+1}^{\text{net}}))] \geq \mathbb{E}[W(\pi_{k+1}^{\text{cent}}, I(\pi_{k+1}^{\text{cent}}))],$$

namely the result. □

6.5.2.2 Networked vs independent populations in coordination games

We now give results showing that learning is at least as fast in the networked case than in the independent case - empirically we find networked learning always to be strictly faster. We give separate theorems for coordination and anti-coordination games. Since we cannot necessarily expect the independent agents to share a single policy π_{k+1} after the update in each iteration of learning, we give these results in terms of the population-average return (Def. 6.3.3) instead of the single-policy social welfare (Def. 6.3.5) as before.

Again, we assume for simplicity of the theory that it is only the policy communication scheme that creates a difference in learning between the networked and independent cases, i.e. we assume that networked agents receive the same estimates of the mean field and average reward as independent agents. As mentioned above, our ablation studies suggest policy communication is the dominant factor in our experimental settings anyway, with the estimated mean field not required at all in the broad class of stationary problems. Nevertheless, in practice the networked estimates of the (mean field and) average reward are likely to be substantially better than the independent ones, giving an additional performance increase over the independent case. Thus loosening this assumption is likely to actually enhance the effects identified in the theorems.

Assumption 6.5.11. Assume that the estimated global mean field and average reward in the networked case are the same as the independent case, i.e. $\forall i, j \quad \tilde{\hat{\mu}}_t^{(i,net)} = \tilde{\hat{\mu}}_t^{(j,ind)}$ and $\tilde{\hat{r}}_t^{(i,net)} = r_t^i$.

We refer to the joint policy in the networked case after communication round c as $\boldsymbol{\pi}_{k+1,c}^{\text{net}} = \left(\pi_{k+1,c}^{(1,\text{net})}, \ldots, \pi_{k+1,c}^{(N,\text{net})}\right)$, and the joint policy in the independent case as $\boldsymbol{\pi}_{k+1}^{\text{ind}} = \left(\pi_{k+1}^{(1,\text{ind})}, \ldots, \pi_{k+1}^{(N,\text{ind})}\right)$.

We can now give our second theorem, namely that in expectation networked populations will increase their returns at least as fast as independent ones in coordination games with only a single round of policy communication in each iteration.

Theorem 6.5.12. *Let us again set $\tau_k^{comm} \in \mathbb{R}_{>0}$. In a coordination game, given Assumptions 6.5.5, 6.5.6 and 6.5.11, for $c = 0$,*

$$\mathbb{E}\left[V^{pop}(\boldsymbol{\pi}_{k+1,c+1}^{\text{net}}, \mu_t)\right] \geq \mathbb{E}\left[V^{pop}(\boldsymbol{\pi}_{k+1}^{\text{ind}}, \mu_t)\right].$$

Remark 6.5.13. As mentioned in Rem. 6.5.10, Assumption 6.5.5 presumes that all policies are pairwise distinct after the updates, but does not restrict their returns in the same way. If we additionally make the very weak assumption that at least one of the distinct policies in each k iteration has a distinct base return from the others (as is generally likely to be the case), the inequality in the theorem above will be strict, i.e. *networked learning will always be faster in expectation.*

Proof. Let us consider two scenarios. Firstly let us imagine that within the communication round, agents swap policies, but no policy drops out of the population, such that if agent i adopts policy π_{k+1}^j, there exists an agent i' that adopts policy π_{k+1}^i, and so on. That way we end up with the same policies in the population as before the change, but with each one possibly carried by different arbitrary agents. This is equivalent to if no communication had taken place, meaning that in this scenario $V^{pop}(\boldsymbol{\pi}_{k+1,c+1}^{\text{net}}, \mu_t) = V^{pop}(\boldsymbol{\pi}_{k+1}^{\text{ind}}, \mu_t)$. The mostly likely circumstance for this to occur is when no σ_{k+1}^i value is distinct from the others, since then every policy is equally likely to remain in the population in expectation.

Let us now consider an alternative scenario. The softmax adoption scheme (Line 11, Alg. 8), which for $\tau_k^{comm} \in \mathbb{R}_{>0}$ gives non-uniform adoption probabilities for distinct σ values, means by definition that some communicated policies are more likely to be adopted than others if they have distinct finitely estimated returns. Thus in expectation the number of distinct policies in the population will decrease if at least one policy has a distinct return from the others (of course, there is a possibility of this still happening even if no policy has a distinct return from the others). Let us start by saying for simplicity that during the first communication round a single $\pi_{k+1,c}^{(j,\text{net})}$ is replaced by $\pi_{k+1,c}^{(i,\text{net})}$, such that for $c = 0$

$$\boldsymbol{\pi}_{k+1,c}^{\text{net}} = \left(\pi_{k+1,c}^{(1,\text{net})}, \ldots, \pi_{k+1,c}^{(\mathbf{i},\text{net})}, \ldots, \pi_{k+1,c}^{(\mathbf{j},\text{net})}, \ldots \pi_{k+1,c}^{(N,\text{net})}\right),$$
$$\text{and} \quad \boldsymbol{\pi}_{k+1,c+1}^{\text{net}} = \left(\pi_{k+1,c+1}^{(1,\text{net})}, \ldots, \pi_{k+1,c+1}^{(\mathbf{i},\text{net})}, \ldots, \pi_{k+1,c+1}^{(\mathbf{i},\text{net})}, \ldots \pi_{k+1,c+1}^{(N,\text{net})}\right).$$

For this to have occurred, we know that

$$\mathbb{E}[\sigma_{k+1,c}^{(i,\text{net})}] > \mathbb{E}[\sigma_{k+1,c}^{(j,\text{net})}],$$

and therefore by Assumption 6.5.6 that

$$\mathbb{E}\left[V^{(i,\text{net})}(\boldsymbol{\pi}_{k+1,c}^{\text{net}}, \mu_t)\right] > \mathbb{E}\left[V^{(j,\text{net})}(\boldsymbol{\pi}_{k+1,c}^{\text{net}}, \mu_t)\right]. \tag{6.6}$$

By Assumption 6.5.5 we know that straight after the random policy updates there is no alignment among policies, i.e. in a coordination game we have $f_c^{(i,\text{net})} = f_c^{(j,\text{net})} = \min f_c$. Therefore if Eq. 6.6 pertains, by Def. 6.5.2 it must be because:

$$\mathbb{E}[b(\pi^{(i,\text{net})})] > \mathbb{E}[b(\pi^{(j,\text{net})})], \tag{6.7}$$

i.e. because the base policy quality is higher for $\pi^{(i,\text{net})}$ than for $\pi^{(j,\text{net})}$. For this reason we have, for $c = 0$:

$$\mathbb{E}\left[V^{pop}(\boldsymbol{\pi}_{k+1,c+1}^{\text{net}}, \mu_t)\right] > \mathbb{E}\left[V^{pop}(\boldsymbol{\pi}_{k+1,c}^{\text{net}}, \mu_t)\right]. \tag{6.8}$$

Additionally, replacing $\pi_{k+1,c}^{(j,\text{net})}$ with a second copy of $\pi_{k+1,c}^{(i,\text{net})}$ will increase the alignment (f_c) of $\pi_{k+1,c}^{(i,\text{net})}$ such that $\mathbb{E}\left[V^{(i,\text{net})}(\boldsymbol{\pi}_{k+1,c+1}^{\text{net}}, \mu_t)\right] > \mathbb{E}\left[V^{(i,\text{net})}(\boldsymbol{\pi}_{k+1,c}^{\text{net}}, \mu_t)\right]$,

increasing the improvement even further. This effect is even greater if more than one policy is replaced.

Since the independent case is equivalent to the networked case when $C_p = 0$, we can say that $\boldsymbol{\pi}^{\text{ind}}_{k+1} = \boldsymbol{\pi}^{\text{net}}_{k+1,0}$. This gives the result, i.e.

$$\mathbb{E}\left[V^{pop}(\boldsymbol{\pi}^{\text{net}}_{k+1,c+1}, \mu_t)\right] \geq \mathbb{E}\left[V^{pop}(\boldsymbol{\pi}^{\text{ind}}_{k+1}, \mu_t)\right],$$

where this inequality will be strict if the first scenario does not apply in expectation.

□

6.5.2.3 Networked vs independent populations in anti-coordination games

To prove the benefit of the networked case over the independent case in anti-coordination games, we use a final additional assumption. This presumes that the base return is not yet fully maximised (as naturally applies for a certain amount of time during training), and that the benefit to an agent's overall return of increasing its base return by adopting a neighbour's better-performing policy outweighs the resulting decrease in diversity. This establishes the conditions under which our policy adoption scheme is able to advantage networked agents over those whose policies are always independent. The second part of the assumption applies in most non-trivial scenarios, namely where the goal of the task is not simply for agents to have distinct policies that are otherwise inconsequential, and thus where the benefit of diverse behaviour can only be fully felt once agents have a certain level of aptitude at accomplishing the given task. For example, in all of the anti-coordination games in our experiments, agents are always penalised for moving, and only start to receive higher rewards if they are stationary. Therefore in these anti-coordination games agents will receive higher returns by *aligning* on policies that prioritise stationarity, than by maintaining diverse policies that have high levels of movement. Of course once base return is maximised and the assumption no longer holds, one can consider terminating policy communication and adoption to avoid decreases in diversity (one may also be ready to stop training entirely at this point, as the population is

likely to be reaching the optimal average return). Please see Sec. 6.6.3 for further discussion of the applicability of this assumption in practice.

Assumption 6.5.14. Assume that the agents have not yet maximised their base return function i.e. $b(\pi^i_{k+1}) < \sup_{\pi\in\Pi} b(\pi) \;\; \forall i \in \{1,\dots,N\}$, and that an increase in the base return function outweighs a decrease in the policy diversity, namely $h(b+\Delta b, f_d - \Delta f_d) > h(b, f_d),\ \forall \Delta b > 0, \Delta f_d > 0$.

We now give our final theorem, namely that in anti-coordination games, in expectation networked populations will increase their returns at least as fast as independent ones with only a single round of communication in each iteration.

Theorem 6.5.15. *Let us once again set $\tau^{comm}_k \in \mathbb{R}_{>0}$. In an anti-coordination game, given Assumptions 6.5.5, 6.5.6, 6.5.11and 6.5.14, for $c = 0$,*

$$\mathbb{E}\left[V^{pop}(\boldsymbol{\pi}^{\text{net}}_{k+1,c+1}, \mu_t)\right] \geq \mathbb{E}\left[V^{pop}(\boldsymbol{\pi}^{\text{ind}}_{k+1}, \mu_t)\right].$$

Remark 6.5.16. As mentioned in Rems. 6.5.10 and 6.5.13, Assumption 6.5.5 presumes that all policies are pairwise distinct after the updates, but does not restrict their returns in the same way. If we additionally make the very weak assumption that at least one of the distinct policies in each k iteration has a distinct base return from the others (as is generally likely to be the case), the inequality in the theorem above will be strict, i.e. *networked learning will always be faster in expectation.*

Proof. The proof begins similarly to that for a coordination game. Let us consider two scenarios. Firstly let us imagine that within the communication round, agents swap policies, but no policy drops out of the population, such that if agent i adopts policy π^j_{k+1}, there exists an agent i' that adopts policy π^i_{k+1}, and so on. That way we end up with the same policies in the population as before the change, but with each one possibly carried by different arbitrary agents. This is equivalent to if no communication had taken place, meaning that in this scenario $V^{pop}(\boldsymbol{\pi}^{\text{net}}_{k+1,c+1}, \mu_t) = V^{pop}(\boldsymbol{\pi}^{\text{ind}}_{k+1}, \mu_t)$. The mostly likely circumstance for this to occur is when no σ^i_{k+1}

value is distinct from the others, since then every policy is equally likely to remain in the population in expectation.

Let us now consider an alternative scenario. The softmax adoption scheme (Line 11, Alg. 8), which for $\tau_k^{comm} \in \mathbb{R}_{>0}$ gives non-uniform adoption probabilities for distinct σ values, means by definition that some communicated policies are more likely to be adopted than others if they have distinct finitely estimated returns. Thus in expectation the number of distinct policies in the population will decrease if at least one policy has a distinct return from the others. Say for simplicity that during the first communication round a $\pi_{k+1,c}^{(j,\text{net})}$ is replaced by $\pi_{k+1,c}^{(i,\text{net})}$, such that for $c = 0$

$$\boldsymbol{\pi}_{k+1,c}^{\text{net}} = \left(\pi_{k+1,c}^{(1,\text{net})}, \ldots, \pi_{k+1,c}^{(\mathbf{i},\text{net})}, \ldots, \pi_{k+1,c}^{(\mathbf{j},\text{net})}, \ldots \pi_{k+1,c}^{(N,\text{net})}\right),$$
$$\text{and} \quad \boldsymbol{\pi}_{k+1,c+1}^{\text{net}} = \left(\pi_{k+1,c+1}^{(1,\text{net})}, \ldots, \pi_{k+1,c+1}^{(\mathbf{i},\text{net})}, \ldots, \pi_{k+1,c+1}^{(\mathbf{i},\text{net})}, \ldots \pi_{k+1,c+1}^{(N,\text{net})}\right).$$

For this to have occurred, we know that

$$\mathbb{E}[\sigma_{k+1,c}^{(i,\text{net})}] > \mathbb{E}[\sigma_{k+1,c}^{(j,\text{net})}],$$

and therefore by Assumption 6.5.6 that

$$\mathbb{E}\left[V^{(i,\text{net})}(\boldsymbol{\pi}_{k+1,c}^{\text{net}}, \mu_t)\right] > \mathbb{E}\left[V^{(j,\text{net})}(\boldsymbol{\pi}_{k+1,c}^{\text{net}}, \mu_t)\right]. \tag{6.9}$$

By Assumption 6.5.5 we know that straight after the random policy updates there is no alignment among policies, i.e. in the anti-coordination game we have $f_d^{(i,\text{net})} = f_d^{(j,\text{net})} = \max f_d$, while by Assumption 6.5.14 we know that the agents have not yet maximised their base return function. Therefore if Eq. 6.9 pertains, by Def. 6.5.2 it must be because:

$$\mathbb{E}[b(\pi^{(i,\text{net})})] > \mathbb{E}[b(\pi^{(j,\text{net})})], \tag{6.10}$$

i.e. because the base policy quality is higher for $\pi^{(i,\text{net})}$ than for $\pi^{(j,\text{net})}$.

Assumption 6.5.14 assumes that any increase in the base quality of the policy will outweigh the decrease in diversity that will come from having more than one agent following $\pi_{k+1,c+1}^{(i,\text{net})}$. Therefore we have, for $c = 0$:

$$\mathbb{E}\left[V^{pop}(\boldsymbol{\pi}_{k+1,c+1}^{\text{net}}, \mu_t)\right] > \mathbb{E}\left[V^{pop}(\boldsymbol{\pi}_{k+1,c}^{\text{net}}, \mu_t)\right].$$

These steps apply similarly if more than one policy is replaced.

Since the independent case is equivalent to the networked case when $C_p = 0$, we can say that $\boldsymbol{\pi}^{\text{ind}}_{k+1} = \boldsymbol{\pi}^{\text{net}}_{k+1,0}$. This gives the result, i.e.

$$\mathbb{E}\left[V^{pop}(\boldsymbol{\pi}^{\text{net}}_{k+1,c+1}, \mu_t)\right] \geq \mathbb{E}\left[V^{pop}(\boldsymbol{\pi}^{\text{ind}}_{k+1}, \mu_t)\right],$$

where this inequality will be strict if the first scenario does not apply in expectation.

□

6.6 Experiments

6.6.1 Experimental setup

We present experiments from grid worlds, following Chs. 4 and 5 and the gold standard in similar works on MFGs and MFC [100]. We give results from six cooperative tasks similar to those found in prior works, defined by the agents' reward functions and relating to agents' positions relative to other agents. Two are coordination tasks and four are anti-coordination tasks, where in each case the reward function reflects a coordination/anti-coordination (f_c/f_d) element alongside other elements that may be crucial for receiving reward, reflected in the policies' base quality $b(\pi)$ (Sec. 6.5). In all cases, rewards are normalised in [0,1] after they are computed, and the cooperative objective is to maximise the social welfare / population-average return.

The two coordination tasks are:

- **Cluster.** This game is also used in Chs. 4 and 5, but we restate it here for ease of reference. Agents are encouraged to gather together by the reward function $R(s^i_t, a^i_t, \hat{\mu}_t) = \log(\hat{\mu}_t(s^i_t))$. 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.

  As in the previous chapters, in the 'cluster' task we expect the population to concentrate at a single state. There are $|\mathcal{S}|$ social optima, one per choice

of clustering state. The sub-optimal NE described for the analogous MFG task are not recognised solutions in MFC, since the welfare-maximisation objective rules them out as targets, but they can still arise as sub-optimal outcomes if the algorithm fails to reach a social optimum. In practice we find that welfare-maximising populations usually cluster at one of the four corners despite the symmetry of the reward across all states: three of the five available actions leave an agent in any corner cell (the 'stay' action plus the two cardinal moves that bounce off a wall), two leave them in any non-corner edge cell, and only 'stay' keeps them at any interior cell - so exploratory policies drift the population toward the corners, and once a cluster begins to form the per-state reward reinforces this bias.

- **Target selection.** This game is also used in Chs. 4 and 5, but we restate it here for ease of reference. 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. Define the magnitude of the distances between x, y at t as $dist_t(x, y)$. The reward function is given by $R(s_t^i, a_t^i, \hat{\mu}_t) = r_{targ}(r_{coord}(\hat{\mu}_t(s_t^i)))$, where

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

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

As in the previous chapters, in the 'target selection' task we expect the population to concentrate at a single target corner. There are $|\Phi| = 4$ social optima, one per choice of target. The sub-optimal NE described for the

analogous MFG task are not recognised solutions in MFC, since the welfare-maximisation objective rules them out as targets, but they can still arise as sub-optimal outcomes if the algorithm fails to reach a social optimum.

The anti-coordination tasks are:

- **Disperse.** This game is also used in Ch. 5 and is similar to the 'exploration' tasks in Laurière et al. [20], Wu et al. [153] and other MFG works. In our version agents are rewarded for being located in more sparsely populated areas *but only if they are stationary*, to avoid trivial random policies. The reward function is given by $R(s_t^i, a_t^i, \hat{\mu}_t) = r_{stationary}(-\log(\hat{\mu}_t(s_t^i)))$, where

$$r_{stationary}(x) = \begin{cases} x & \text{if } a_t^i \text{ is 'remain stationary'} \\ -1 & \text{otherwise.} \end{cases}$$

  As in the 'disperse' task in Ch. 5, the uniform stationary distribution is the unique optimum. Here it is characterised as the social optimum under MFC rather than as the Pareto-dominant MFG-NE.

- **Target coverage.** The population is rewarded for spreading across a certain number of targets, as long as agents are stationary at the target. As in the 'target selection' game, we have targets $\phi \in \Phi$, where in our experiments we place one target in each of the four corners of the grid. Again define the magnitude of the distances between x, y at t as $dist_t(x, y)$. The reward function is given by

$$R(s_t^i, a_t^i, \hat{\mu}_t) = r_{stationary}\left(r_{targ}\left(-\log(\hat{\mu}_t(s_t^i))\right)\right),$$

  where $r_{stationary}$ and r_{targ} are as defined above.

  In the 'target coverage' task we expect the population to spread evenly across the four targets, with population mass $\approx N/|\Phi|$ at each and all agents stationary. Uniformity over the four targets is the unique social optimum.

- **Beach bar.** Such games are very common in MFG works [19, 100, 126, 153]. In our version agents are rewarded for being stationary in sparsely populated locations as close as possible to a target ϕ_b, located in the centre of the grid. The maximum possible distance from the target is denoted *maxDist*. The reward is given by

$$R(s_t^i, a_t^i, \hat{\mu}_t) = r_{stationary}\left(maxDist - dist_t(s_t^i, \phi_b) - \log(\hat{\mu}_t(s_t^i))\right),$$

  where $r_{stationary}$ is as defined above.

  In the 'beach bar' task the unique social optimum is a stationary Boltzmann distribution, with density $\propto e^{-dist(s,\phi_b)}$, peaked at ϕ_b and decaying exponentially with distance from it.

- **Shape formation.** The population is rewarded for spreading around a ring shape, accomplished by encouraging agents to be dispersed a distance of 3 (chosen arbitrarily to fit the grid) from a centre point ϕ_c. The reward is given by

$$R(s_t^i, a_t^i, \hat{\mu}_t) = r_{stationary}\left(r_{ring}\left(-\log(\hat{\mu}_t(s_t^i))\right)\right),$$

  where $r_{stationary}$ is as defined above, and

$$r_{ring}(x) = \begin{cases} x & \text{if } dist_t(s_t^i, \phi_c) = 3 \\ -1 & \text{otherwise.} \end{cases}$$

  In this task the unique social optimum has all agents stationary on a ring of distance 3 around ϕ_c, with mass distributed uniformly along the ring.

We now consider whether the MFC solutions to these tasks differ from their non-cooperative MFG counterparts, particularly for the tasks with direct analogues in earlier MFG chapters. The MFC social optimum coincides with the *Pareto-dominant* MFG-NE in all six of our tasks. In each case both are realised by a population-symmetric joint policy $\boldsymbol{\pi} = (\pi^*, \ldots, \pi^*)$: in the coordination tasks ('cluster', 'target selection') every individual benefit from being in a high-density region is also a

population-average benefit, and in the anti-coordination tasks ('disperse', 'target coverage', 'beach bar', 'shape formation') the reward functions are concave in $\hat{\mu}(s)$ so the uniform-or-prescribed-shape distribution that maximises social welfare is also one that no individual can improve upon. Whether the MFG algorithm is more like to reach this Pareto-dominant NE rather than one of the sub-optimal NE described in previous chapters (e.g. partition-based NE in the 'cluster' and 'target agreement' tasks) depends on the learning architecture (networked, central-agent, or independent), the communication radius, and stochastic factors (initialisation and sampling realisations) that may by chance drive the population toward the Pareto-dominant NE. We do not observe a consistent gap between MFG-NE and MFC social-welfare returns in our results in Sec. 6.6.3.

As in the previous chapters, in these spatial environments, we choose to define both the communication network $\mathcal{G}_t^{comm}$ and the visibility graph $\mathcal{G}_t^{vis}$ by the physical distance from i, though this does not need to be the case. We show plots for various transmission radii, given as fractions of the maximum distance in the grid. Note that the networked population with the largest radius is always fully connected, and therefore these agents are always able to accurately estimate $\hat{r}_t$ and $\hat{\mu}_t$ even for $C_r = 1$ and $C_e = 0$. That is, when we set $C_r = C_e > 0$ their observations are equivalent to those that the central-agent population would receive, albeit that policies are updated and spread differently.

Similar to the second metric used in the previous two chapters, we evaluate our experiments according to a finite-step estimate of the population-average discounted return (Def. 6.3.3) over the M steps within each outer k loop (Line 4, Alg. 7), i.e. $\hat{V}^{pop}(\boldsymbol{\pi}_k, \mu_t; M)$. Experiments were conducted on a Linux-based machine with 2 x Intel Xeon Gold 6248 CPUs (40 physical cores, 80 threads total, 55 MiB L3 cache). We use the JAX framework to accelerate and vectorise our code. We run five trials with different random seeds for each experiment, and plot the mean and standard deviation of the mean across the seeds. Random seeds are set in our code in a fixed way dependent on the trial number to allow easy replication of experiments.

6.6.2 Hyperparameters

See Table 6.1 for our hyperparameter choices. We can group our hyperparameters into those controlling the size of the experiment, those controlling the size of the Q-network, those controlling the number of iterations of each loop in the algorithms and those affecting the learning/policy updates or policy adoption.

In our experiments we generally want to demonstrate that our communication-based algorithm learns faster than the central-agent and independent architectures, even when the Q-function / mean field / average reward are poorly estimated as is likely to be the case in complex real-world scenarios. We have a similar motivation in the MFG setting in Chs. 4 and 5. Moreover we want to show that there is a large benefit even to a small amount of communication, so that communication rounds themselves do not excessively add to time complexity. As such, we generally select hyperparameters at the lowest end of those we tested during development, to show that our algorithms are particularly successful and robust given what might otherwise be considered 'undesirable' hyperparameter choices.

Table 6.1: Hyperparameters

Hyper-param.	Value	Comment
Trials	5	We run 5 trials with different random seeds for each experiment. We plot the mean and standard deviation of the mean for each metric across the seeds.
Gridsize	20x20	-
Population	500	We chose 500 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, while also being feasible to simulate with respect to time and computation constraints [105, 108, 126, 142, 153, 201, 229–232]. For example, the MFC work in Carmona et al. [120] uses 10 agents; the work on decentralised execution for MFC by Cui et al. [7] uses 200 agents.
Number of neurons in input layer	440	The agent's position is represented by two concatenated one-hot vectors, indicating the agent's row and column. The mean-field distribution is a flattened vector of the same size as the grid. As such, the input size is $[(2 \times \text{dimension}) + (\text{dimension}^2)]$.

Continued on next page

Table 6.1: Hyperparameters (continued)

Hyper-param.	Value	Comment
Neurons per hidden layer	256	We draw inspiration from common rules of thumb when selecting the number of neurons in hidden layers, e.g. it should be between the number of input neurons and output neurons / it should be 2/3 the size of the input layer plus the size of the output layer / it should be a power of 2 for computational efficiency. Using these rules of thumb as rough heuristics, we select the number of neurons per hidden layer by rounding the size of the input layer down to the nearest power of 2. The layers are all fully connected.
Hidden layers	2	We achieved sufficient learning speed with just 2 hidden layers, but further optimising the number of layers may lead to better results.
Activation function	ReLU	This is a common choice in deep RL.
K	150	K is chosen to be large enough to see convergence in most networked cases.
M	20	We tested M in {20,50,100} and found that the lowest value was sufficient to achieve convergence while minimising training time. It may be possible to converge with even smaller choices of M.
L	20	We tested L in {20,50,100} and found that the lowest value was sufficient to achieve convergence while minimising training time. It may be possible to converge with even smaller choices of L.
E	20	We tested E in {20,50,100}, 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.
C_p	1 (10/50)	As in Chs. 4 and 5, we choose a value of 1 for most experiments to show the convergence benefits brought by even a single policy communication round, even in networks that may have limited connectivity. We also conduct additional studies to show the effect of further rounds in Figs. 6.3 and 6.4.
C_r	1 (10/50)	Similar to C_p, we choose this value to show our algorithm's ability to appropriately estimate the average reward even with only a single communication round, even in networks that may have limited connectivity. We conduct additional studies to show the effect of further rounds in Figs. 6.3 and 6.4.
C_e	1 (10/50)	Similar to C_p, we choose this value to show the ability of our algorithm to appropriately estimate the mean field even with only a single communication round, even in networks that may have limited connectivity. We also conduct additional studies to show the effect of further rounds in Figs. 6.3 and 6.4.
γ	0.9	Standard choice across RL literature.
τ_q	0.03	We follow Vieillard et al. [260] and Ch. 5, where we tested a range of values.
$\|B\|$	32	This is a common choice of batch size that trades off noisy updates and computational efficiency.

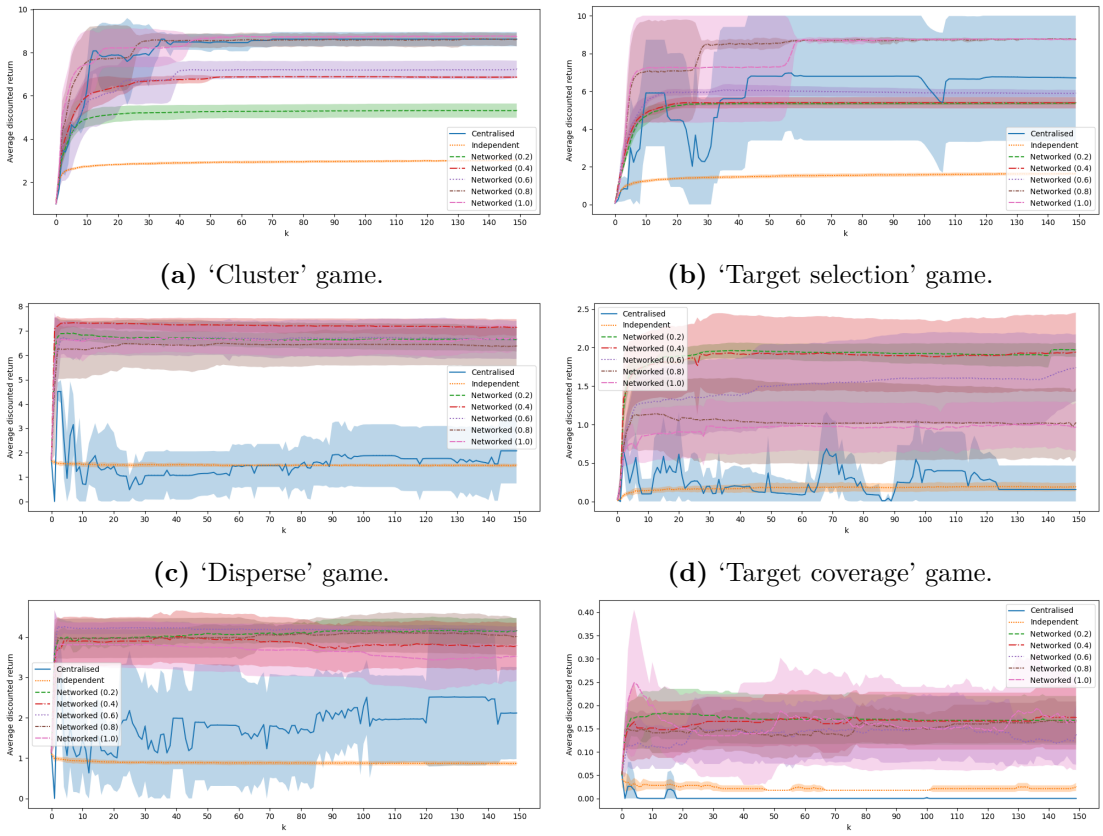
Continued on next page

Table 6.1: Hyperparameters (continued)

Hyper-param.	Value	Comment
cl	-1	We use the same value as in Ch. 5 and Vieillard et al. [260].
ν	$L-1$	We follow Ch. 5, which is similar to Laurière et al. [20].
Optimiser	Adam	As in Vieillard et al. [260], we use the Adam optimiser with initial learning rate 0.01.
τ_k^{comm}	cf. comment	We follow Ch. 5, where τ_k^{comm} increases linearly from 0.001 to 1 across the K iterations. Further optimising this inverse annealing process may lead to better results; we provide an ablation study in Fig. 6.9.

6.6.3 Results and discussion

Fig. 6.1 gives results for our standard experimental settings involving 500 agents, each with their own Q-network. When networked agents communicate, they have only a *single* communication round. Fig. 6.1 shows that in all of our tasks, networked populations of all broadcast radii significantly outperform independent (orange) agents, which hardly appear to increase their returns, if at all. Networked populations of all broadcast radii also significantly outperform the central-agent (blue) populations in all but the two coordination tasks, where only networked agents of the smaller radii (green, 0.2; red, 0.4; purple, 0.6) underperform them (probably due to these less connected populations being more likely to experience violations of Assumption 6.5.7 on policy consensus, which is a disadvantage in scenarios where alignment is beneficial). In the anti-coordination tasks the central-agent populations perform similarly to purely independent ones in hardly appearing to increase their returns, performing even worse than independent agents in the 'shape formation' task. The central-agent populations also have markedly higher variance than networked ones in several tasks ('target selection', 'disperse', 'beach bar'). This reflects our theoretical analysis in Sec. 6.5 that the central learner pushes an arbitrary updated policy to the whole population regardless of its quality, leading to large fluctuations in performance, whereas our communication scheme biases networked populations towards better performing updates.



(a) 'Cluster' game. **(b)** 'Target selection' game.

(c) 'Disperse' game. **(d)** 'Target coverage' game.

(e) 'Beach bar' game. **(f)** 'Shape formation' game.

Figure 6.1: Standard settings with $C_e = C_r = C_p = 1$. In all tasks networked agents of all broadcast radii significantly outperform the independent (orange) populations, and in most tasks they also outperform the central-agent (blue) populations, reflecting our theoretical results. The central-agent populations also have markedly higher variance than networked ones in several tasks, since the central learner pushes an arbitrary updated policy to the whole population regardless of its quality, leading to large fluctuations in performance, whereas our communication scheme biases networked populations towards better performing updates.

In the 'target coverage' task, and sometimes the other anti-coordination tasks to a lesser extent, networked agents of smaller broadcast radii appear to outperform those of larger radii, i.e. the ordering is reversed from that of the coordination tasks, albeit not necessarily significantly so. This reflects the point up to which our Assumption 6.5.14 (base return is not yet maximised, and increase in base return outweighs decrease in diversity in anti-coordination tasks) holds in practice, which we discuss in the following.

The second part of Assumption 6.5.14 strictly holds throughout the 'disperse', 'target coverage' and 'shape formation' anti-coordination tasks: agents get no reward

for diversity unless they are stationary (and also unless they are in one of the correct locations in the latter two cases). This means that any increase in base return (likelihood of being stationary or in the right location) achieved by policy adoption does indeed outweigh the loss of diversity. The second part of Assumption 6.5.14 mostly holds in the 'beach bar' game, apart from in a small window for agents that are stationary close to the bar target, with the window defined by the size of the empirical population and hence the potential magnitude of the $\log(\hat{\mu}_t(s_t^i)$ term in the reward function. Inside this window, increasing base return by moving even closer to the target, at the cost of being in a more crowded area, would not necessarily be beneficial. Regardless, in all of these tasks the networked populations of all broadcast radii significantly outperform the independent agents, which do not appear to be able to learn at all without the helpful bias towards policies with better base returns enabled by the communication scheme.

However, among these networked populations, the base return quickly reaches its capacity, i.e. agents learn to be primarily stationary in one of the right locations, such that the first part of Assumption 6.5.14 no longer holds. This is not an issue when comparing with the independent populations, which have not maximised their base returns and therefore perform worse, but it does give rise to the reverse ordering of returns which we see among networked populations of different radii and hence connectivities. Once base return is maximised, policies that are estimated to receive higher returns in these anti-coordination tasks may be less aligned with other policies than those other policies are with each other (at least regarding the strategically relevant parts of policies which are rewarded for greater diversity, e.g. these policies visit the less congested locations), or they simply visited the less congested locations by chance during the finite evaluation steps. Either way, more adoption of policies now becomes a disadvantage, since it reduces diversity without an additional positive impact on base return. Therefore architectures that give less communication now perform better by preserving diversity. Populations with lower broadcast radii usually have less connected networks, especially if sub-populations become isolated from each other, which is more likely in our 'target coverage' game

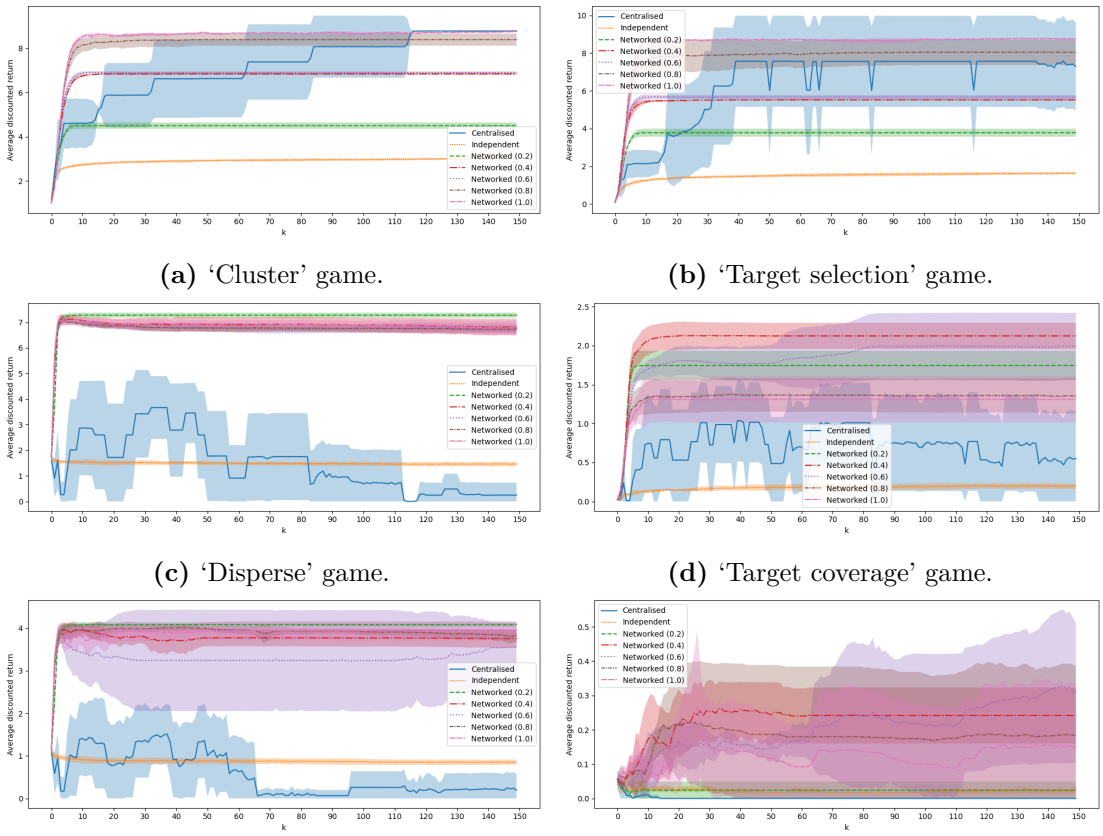
than the others since the target locations are as far apart as possible from each other. Therefore these populations have less communication than those with larger broadcast radii and so may perform better, even while all networked populations outperform the independent agents that have not maximised their base returns.

This intuition also gives additional justification for why central-agent populations significantly underperform networked populations in these anti-coordination games, especially when policy consensus is not enforced for the networked populations. The ultimate choice of consensus level might depend on the considerations from Rem. 1.2.1: namely, whether one is using the empirical population as a practical way of learning the social optimum for a MFC problem (Def. 6.3.6), where a single policy π^* is desired to be given to an infinite population, or whether one is solving the MFC problem to approximate the solution to a finite-agent control problem (maximising Def. 6.3.3) involving the same number of agents as the empirical population from which one is learning. In the latter case some policy diversity may be accepted/desired if it affords a better approximation to the N-agent solution.

Further studies We provide numerous additional experiments and ablation studies. We list these below, but please find the full discussion of results in the caption for each figure. Of particular note, the ablation studies of Algs. 6 (estimating global average reward) and 9 (estimating global empirical mean field) suggest that in our experimental settings the policy communication scheme (Alg. 8) is the dominant factor in the better performance of networked populations over the other architectures.

- Robustness to communication failures - Fig. 6.2.
- Increased communication rounds - Figs. 6.3 and 6.4.
- Ablation study with population-independent policies - Fig. 6.5.
- Ablation study of Alg. 9 for estimating the empirical mean field (all agents directly receive the true empirical mean field) - Fig. 6.6.

- Ablation study for observation of true/estimated average reward (agents only see their individual reward) - Fig. 6.7.
- Ablation study for Alg. 6 for estimating the true global average reward (all agents directly receive the true global average reward) - Fig. 6.8.
- Ablation study of the choice of τ_k^{comm} - Fig. 6.9.



(a) 'Cluster' game.

(b) 'Target selection' game.

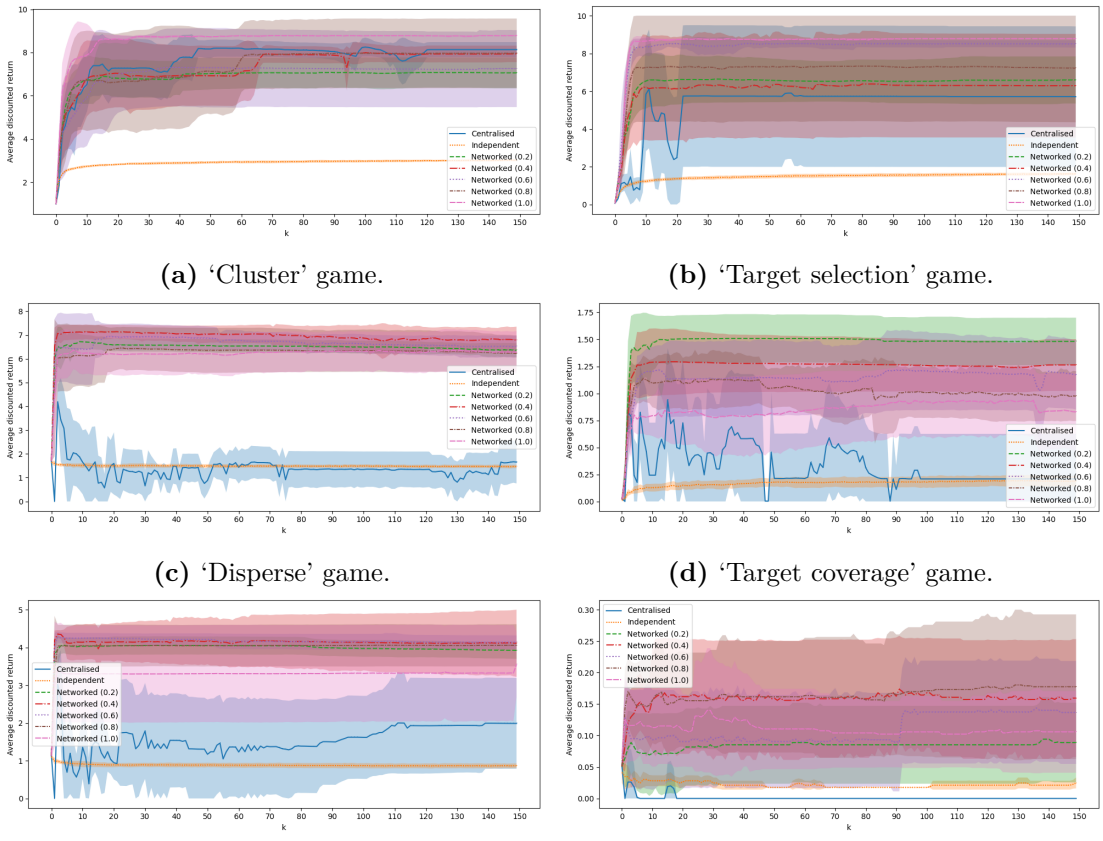
(c) 'Disperse' game.

(d) 'Target coverage' game.

(e) 'Beach bar' game.

(f) 'Shape formation' game.

Figure 6.2: All communication links suffer a 90% probability of failure, including in the central-agent case, where the link between the central learner and the rest of the population may fail. $C_e = C_r = C_p = 1$. The central-agent population, which in the standard setting matched networked performance only in the 'cluster' game, now learns slower even in this game, due to suffering from the single point of failure. Our networked scheme appears robust to the failures in all tasks, with only small differences compared to performance in the standard setting. In fact, several broadcast radii appear to perform better in the 'shape formation' game with these failures than without (though not significantly so), probably because the reduced communication permits greater diversity in policies while still having an advantage over purely independent learners (as discussed in the body of Sec. 6.6.3). However, the smallest broadcast radius (green, 0.2) does drop in performance in this game, which might be expected given it now acts similarly to the independent case. Networked populations appear to have less variance in this setting than in the standard setting, at least in the first four games. This is possibly because the communication failures prevent both particularly high- and particularly low-performing policies from spreading fast in the population, preventing large performance fluctuations and smoothing learning progress. Meanwhile central-agent populations still have large variance even with communication failures, due to enforcing the adoption of an arbitrarily-chosen consensus policy - in some games variance is higher in this setting (though in some it may be marginally lower). This points to an additional benefit of our networked scheme over the central-agent case.



(a) 'Cluster' game.

(b) 'Target selection' game.

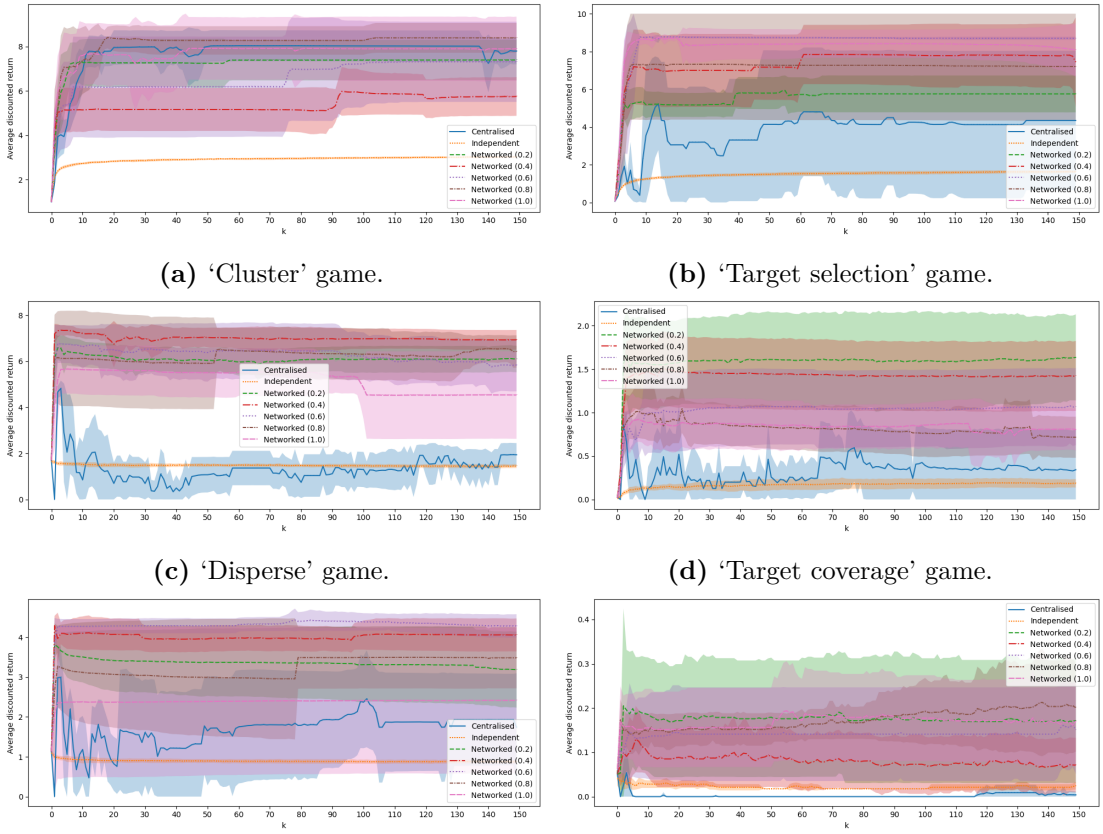
(c) 'Disperse' game.

(d) 'Target coverage' game.

(e) 'Beach bar' game.

(f) 'Shape formation' game.

Figure 6.3: Standard algorithms but $C_e = C_r = C_p = \mathbf{10}$. As is expected, in the coordination games the networked agents with lower broadcast radii now receive returns almost as high as those with larger radii, albeit at the cost of greater variance (having more communication rounds leads to greater policy consensus in the population at each iteration of the outer loop, and there may be some noise in the quality of these consensus policies). In the 'target selection' game, now all networked populations appear to outperform the central-agent (orange) population, though again with high variance. In the anti-coordination 'target coverage' game, the smaller broadcast radii (green, 0.2; red, 0.4; purple, 0.6) receive slightly lower returns than before, since the additional communication rounds now make policy alignment more likely, reducing f_d as per Def 6.5.3. The same is true of the smallest radius population (green, 0.2) in the 'shape formation' game, which receives a lower return than before. This reflects the discussion in the body of Sec. 6.6.3 regarding the detrimental effect of additional policy adoption once the maximum base return has been achieved in anti-coordination games. Nevertheless, *all* networked populations receive higher returns than the independent agents in all games, and also than the central-agent population in all but the 'cluster' game. This shows that in our experimental settings there is a very large benefit to a single communication round, with limited benefit to increasing the algorithms' time complexity with additional communication rounds.



(a) 'Cluster' game.

(b) 'Target selection' game.

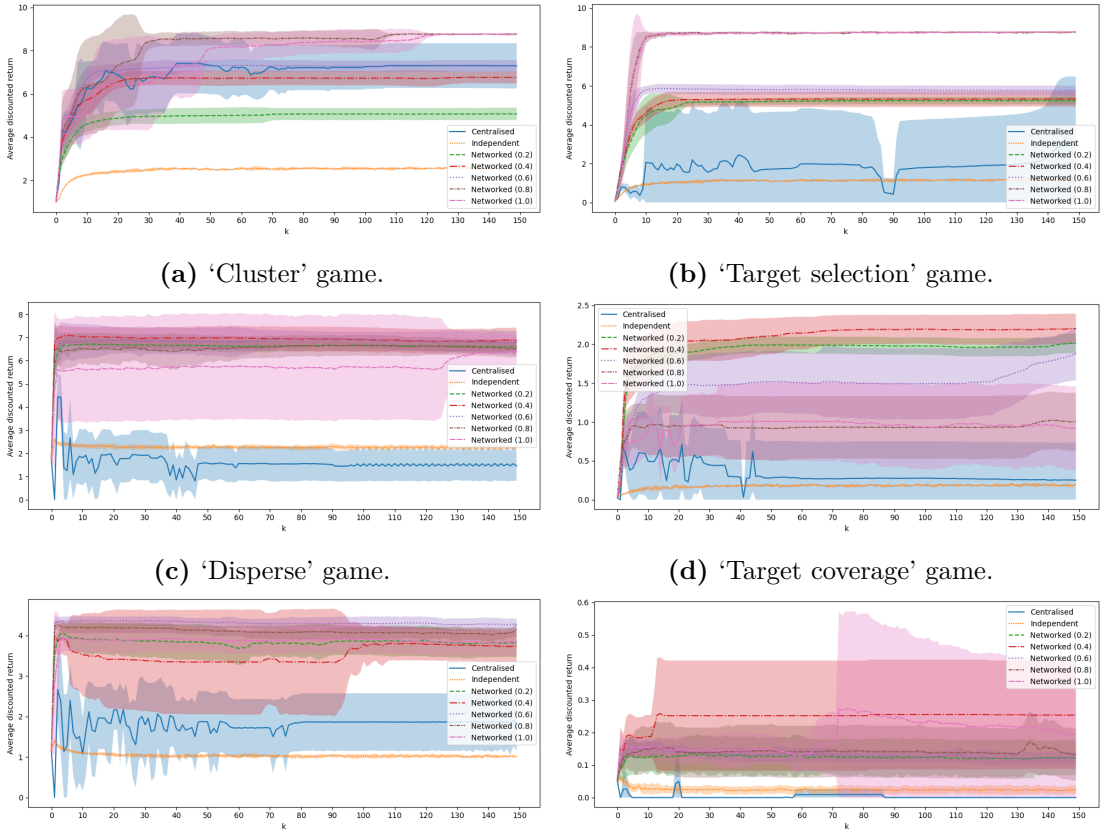
(c) 'Disperse' game.

(d) 'Target coverage' game.

(e) 'Beach bar' game.

(f) 'Shape formation' game.

Figure 6.4: Standard algorithms but $C_e = C_r = C_p = \mathbf{50}$. Having 50 communication rounds does not appear to significantly change networked performance compared to 10 rounds (Fig. 6.3), with most increases or decreases in average return appearing within the margin of error. Most notably, the largest broadcast radius (pink, 1.0) receives slightly lower return now than with 10 rounds in the 'disperse' game, while pink (1.0), brown (0.8) and green (0.2) receive lower returns and have higher variance now in the 'beach bar' game. As in the case of $C_e = C_r = C_p = 10$, additional communication rounds make policy alignment more likely, reducing f_d as per Def 6.5.3. This reflects the discussion in the body of Sec. 6.6.3 regarding the detrimental effect of additional policy adoption once the maximum base return has been achieved in anti-coordination games. Nevertheless, *all* networked populations receive higher returns than the independent agents in all games, and also than the central-agent population in all but the 'cluster' game. This shows that in our experimental settings there is a very large benefit to a single communication round, with limited benefit to increasing the algorithms' time complexity with additional communication rounds.

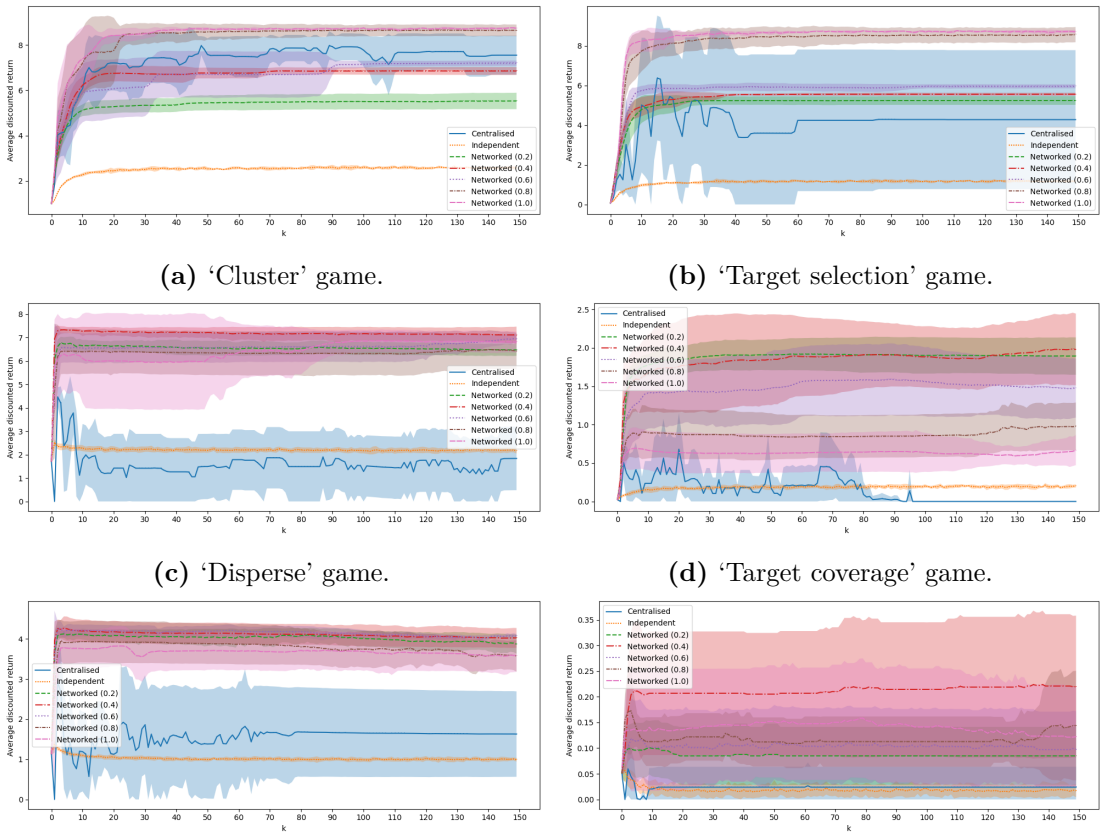


(a) 'Cluster' game. **(b)** 'Target selection' game.

(c) 'Disperse' game. **(d)** 'Target coverage' game.

(e) 'Beach bar' game. **(f)** 'Shape formation' game.

Figure 6.5: Ablation study of population-*independent* policies. No agents, including centralised and networked ones, observe or estimate the empirical mean field, and all receive a vector of zeros in its place (so as to keep the neural networks the same size as in the standard setting). $C_r = C_p = 1$. Networked populations do not appear to perform substantially differently to the standard population-dependent setting, though some radii (red, 0.4; pink, 1.0) appear to perform slightly better in the 'shape formation' game. This is likely because all of our games have stationary solutions, such that observing the mean field is not actually necessary, even if it could potentially be useful (see Sec. 6.3.1 for discussion of the conception of MFC as a central planner trying to guide the population to a distribution that maximises the expected return). Indeed, in the coordination games, and particularly the 'target selection' game, the central-agent population receives a lower return in this setting, whereas our networked populations are robust to this change.

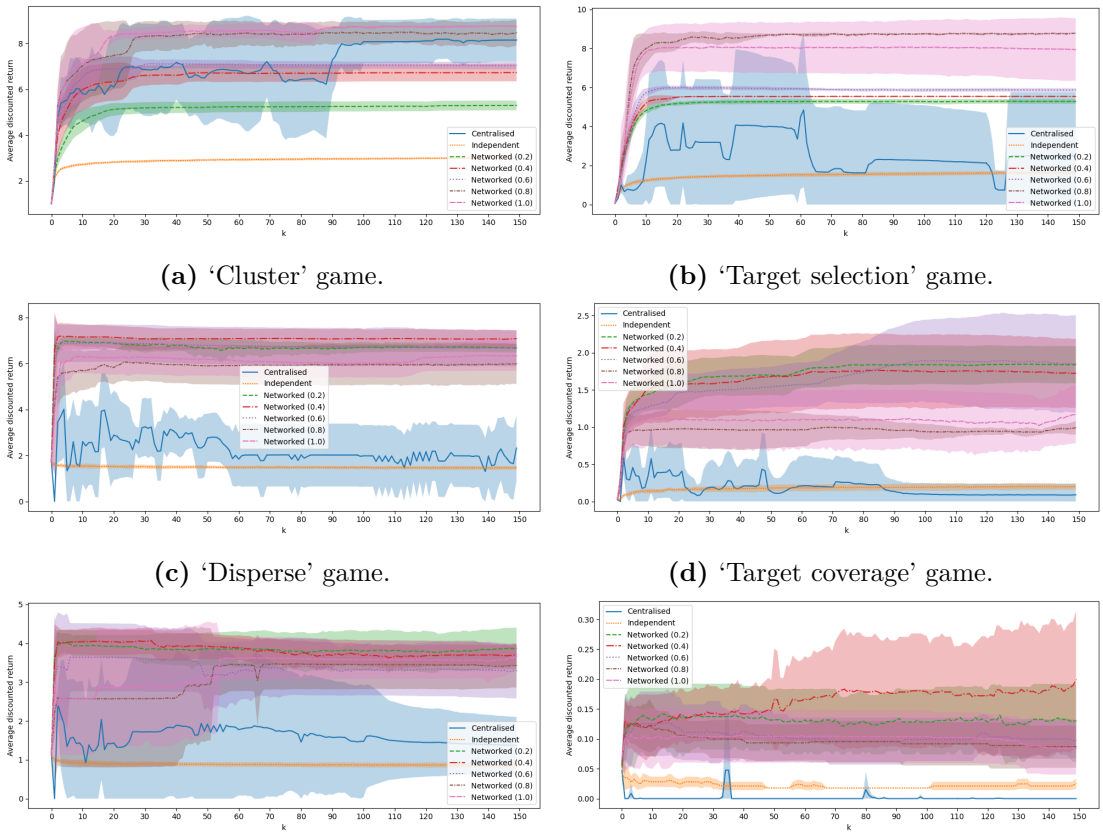


(a) 'Cluster' game. **(b)** 'Target selection' game.

(c) 'Disperse' game. **(d)** 'Target coverage' game.

(e) 'Beach bar' game. **(f)** 'Shape formation' game.

Figure 6.6: Ablation study of Alg. 9 for estimating the empirical mean field - all agents, including independent ones, directly receive the true global empirical mean field. $C_r = C_p = 1$. This does not appear to change performance in the networked populations (apart from greater variance here in the 'shape formation' game), nor does it help independent agents. This may be evidence that Alg. 9 enables networked agents to accurately estimate the global mean field from local observations. However, our ablation study on population-independent policies (Fig. 6.5) suggests that not observing the mean field does not markedly disadvantage agents in our experimental settings in any case (apart from for the central-agent populations in the coordination games). This is likely because all of our games have stationary solutions, such that observing the mean field is not necessary. Therefore further evidence is perhaps needed in MFC settings that require population-dependent policies, in order to confirm the efficacy of Alg. 9 for estimating the mean field, though in Ch. 5 we already showed this for non-stationary games in the non-cooperative MFG setting.



(a) 'Cluster' game.

(b) 'Target selection' game.

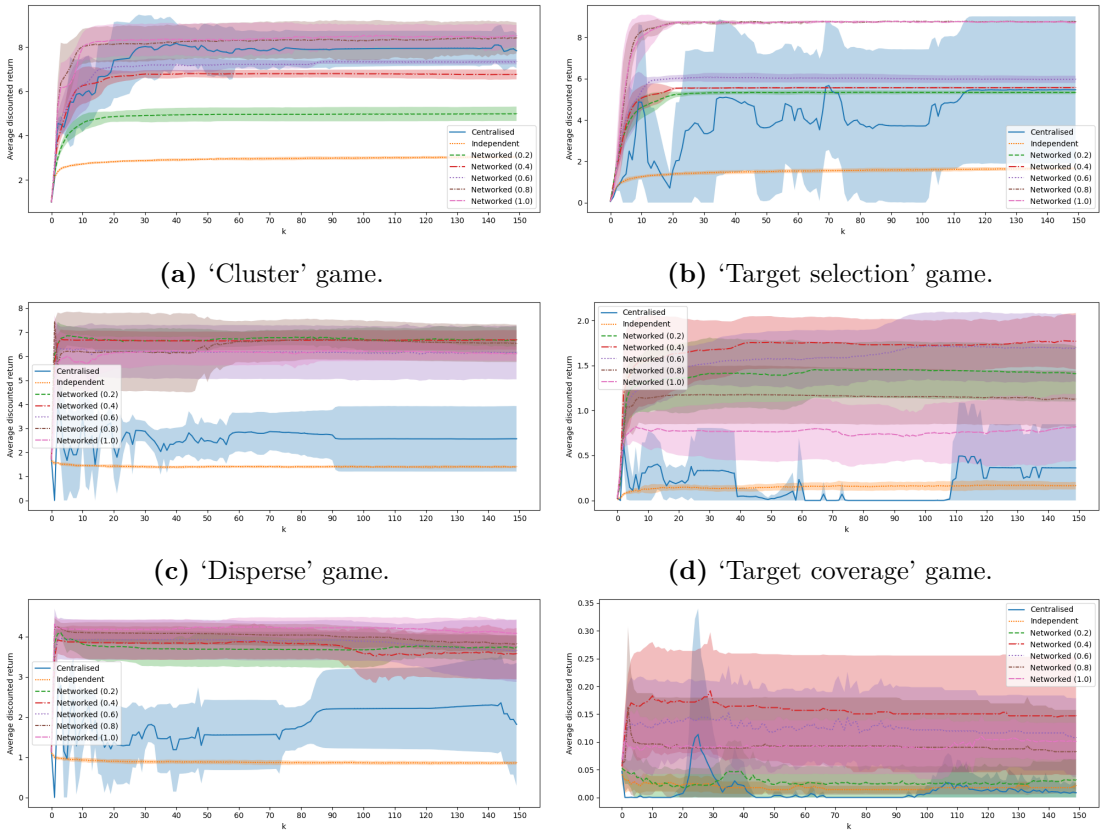
(c) 'Disperse' game.

(d) 'Target coverage' game.

(e) 'Beach bar' game.

(f) 'Shape formation' game.

Figure 6.7: Ablation study for observation of true/estimated global average reward $\hat{r}_t/\tilde{\hat{r}}_t^i$, where all agents, including centralised ones, only have access to r_t^i, where in the central-agent case $i = 1$. $C_e = C_p = 1$. The greatest effect of this is on the central-agent (blue) populations, which perform much worse in the 'target selection' game, and with higher variance in the 'cluster' and 'beach bar' games, i.e. they suffer without access to the global average reward. The networked agents appear more robust to the loss of the (estimated) average reward, pointing to an additional benefit of the policy communication scheme, though do experience a slight performance decrease, mostly among populations with the largest broadcast radii (pink, 1.0; brown, 0.8), i.e. those most similar to the central-agent case in terms of $\tilde{\hat{r}}_t^i$, as might be expected. In particular, note the greater variance of pink (1.0) in the 'target selection' game; slower learning and higher variance of pink (1.0) and brown (0.8) in the 'beach bar' game; lower returns for pink (1.0) and brown (0.8) in the 'shape formation' game; and slower learning and convergence of the smallest radii (green, 0.2; red, 0.4) in the 'target coverage' game. This all demonstrates the usefulness and efficacy of our novel Alg. 6 for decentralised estimation of the global average reward.



(a) 'Cluster' game.

(b) 'Target selection' game.

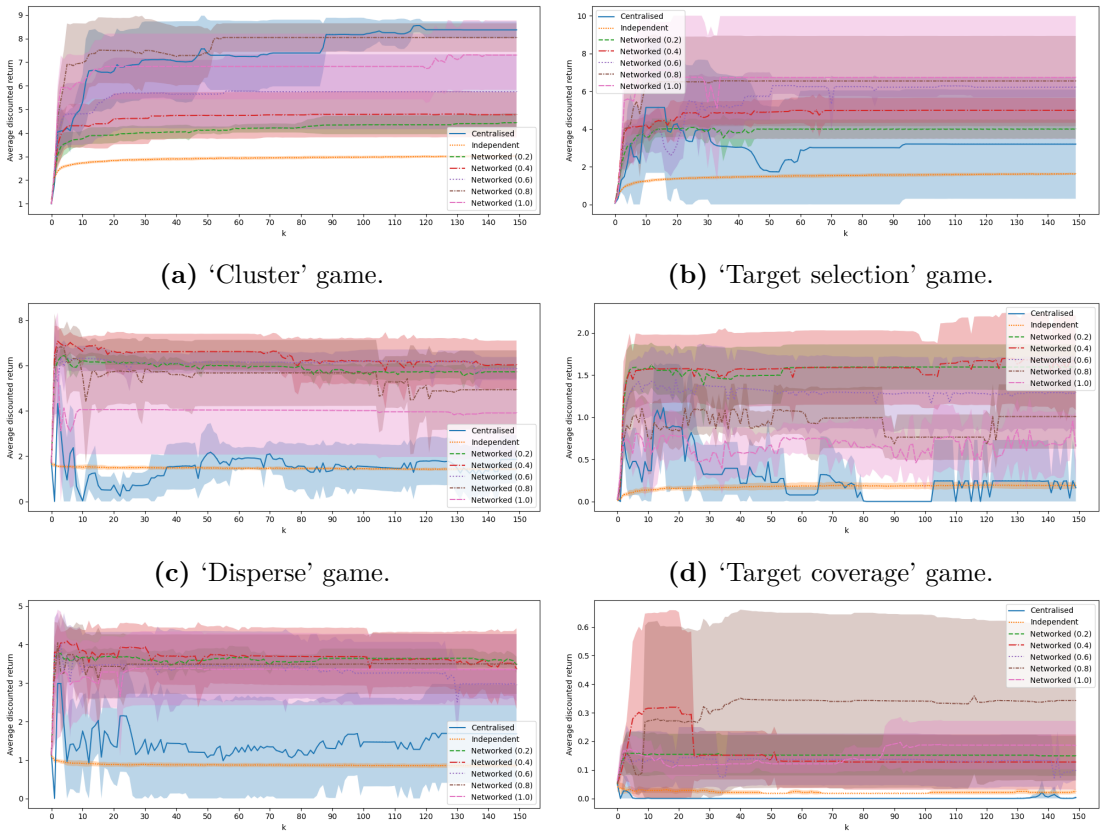
(c) 'Disperse' game.

(d) 'Target coverage' game.

(e) 'Beach bar' game.

(f) 'Shape formation' game.

Figure 6.8: Ablation study for Alg. 6 for estimating the global average reward. All agents, including both networked and independent ones, directly receive the true global average reward such that $\tilde{\hat{r}}_t^i = \hat{r}_t$. $C_e = C_p = 1$. Access to the true average reward does not help networked agents to improve their returns, demonstrating that our novel Alg. 6 already affords networked populations robustness against the lack of access to this global information (having this global information would be an unrealistic assumption in practice). Access to the true average reward also does not help independent agents to improve their returns, suggesting that the *policy* communication scheme is the dominant factor in improving the performance of decentralised agents.



(a) 'Cluster' game. **(b)** 'Target selection' game.

(c) 'Disperse' game. **(d)** 'Target coverage' game.

(e) 'Beach bar' game. **(f)** 'Shape formation' game.

Figure 6.9: Ablation study of the choice of τ_k^{comm}. Here $\forall k\ \tau_k^{comm} = 1\text{e-}18$ (i.e. τ_k^{comm} is close to 0, turning the softmax into a max function), rather than linearly increasing from 0.001 to 1 across the K iterations as in all other experiments (see Table 6.1). $C_e = C_r = C_p = 1$. In this setting, networked agents continue to outperform the central-agent (blue) and independent (orange) populations in all games except the 'cluster' game, but otherwise generally appear to receive lower average returns than before and with greater variance. This is because Assumption 6.5.6 on the quality of the finite-step approximations $\{\sigma_{k+1}^i\}_{i=1}^N = \{\hat{V}^i(\boldsymbol{\pi}_{k+1}, \mu_t; E)\}_{i=1}^N$ may not always apply in practice, especially as the difference between updated policies becomes less stark once they are closer to convergence. This means the policy estimated to perform the best may not actually be among the best updates, such that enforcing the adoption of this policy can lead to noisy, unstable learning. Using a higher temperature value smooths out this noise. On the other hand, having a lower temperature ensures faster learning at the beginning of training when the difference in the quality of nascent policies is likely to be more stark, hence our inverse annealing scheme. Moreover, using τ_k^{comm} close to 0 more effectively enforces consensus on a single policy in the networked case, which in anti-coordination games may also reduce the average return (see the body of Sec. 6.6.3). This all provides empirical support for our inverse annealing scheme for τ_k^{comm}, but further optimising the choice might lead to additional performance increase.

6.7 Conclusion

We provided the first algorithms for decentralised training in MFC, as well as the first for online learning in MFC from a single non-episodic run of the empirical system. We did so by modifying our algorithms from the MFG setting, and contributing a novel algorithm for estimating the global average reward via local communication. We proved theoretically that networked communication can accelerate learning over both the independent and central-agent architectures. We supported this with extensive numerical results, accompanied by ablation studies and discussion of the empirical effects of communication radii. For discussion of potential avenues for future work, please see Ch. 7.

7 Conclusion

Contents

7.1 Conclusion

In this thesis we have introduced networked communication to the mean-field framework. In Ch. 4 we theoretically related a general policy exchange scheme to existing sample guarantees for central-agent and independent-learning algorithms for solving MFGs from a single, non-episodic run of the empirical N-agent population, in settings with tabular Q-functions. We observed that the theoretical conditions underpinning these sample guarantees would give infeasibly slow learning in practice, but found that incorporating replay buffers made convergence much more attainable, allowing us to give the first empirical demonstrations of all three architectures in this setting. We also introduced the specific policy adoption scheme that we continue to use in the rest of our work: updated policies that are estimated to perform better

are more likely to be adopted from neighbours. We showed empirically that in this tabular setting our networked agents markedly outperform independent ones while avoiding the undesirable assumption of a central learner; we also showed that our architecture is more robust than the alternatives in various ways.

In Ch. 5 we modified our algorithms to allow function approximation, which permits both computational scalability to more complex environments, and the consideration of policies that depend on the mean field as well as the agent's local state. This in turn allowed us to move to non-stationary MFGs and those experiencing common noise. We also introduced a second use of the communication network, namely sub-routines where agents estimate the mean field from their local neighbourhood, and can improve their estimates via communication with neighbours. We proved that in this setting our communication architecture allows networked agents to outperform even central-agent populations, with the difference being more marked as variance in the quality of Q-function approximation grows, meaning the algorithms can be run with fewer iterations, making learning faster in practice. Our experiments showed that our networked architecture allows populations to significantly outperform both alternatives in this setting.

In Ch. 6 we modified our function approximation algorithms toward the MFC setting, and introduced a third use of the communication network: for estimating the global average reward from a local neighbourhood to serve as a cooperative objective. In the cooperative problem we can conceptually broaden the incentives for adopting neighbours' policies to more tasks, namely anti-coordination games as well as coordination games (though in Ch. 5 we indicated that the distinction may not be necessary in practice even in MFGs). We gave theoretical analysis of the conditions under which our networked architecture outperforms the central-agent and independent baselines in the different classes of game, and discussed how even when loosening our assumptions, our results still provide heuristic insight into the benefits of our policy communication scheme. We also gave extensive experiments in this setting, showing that our networked architecture can learn faster than the alternatives even when facing massive communication failures.

These contributions are of general conceptual interest, and touch upon a number of different areas, including mean-field theory, (deep) reinforcement learning, networked communication and distributed evolutionary algorithms. They also represent progress in our stated objectives of mean-field algorithms that face fewer obstacles to deployment in practical scenarios. Most notably, in computationally restricted settings networked communication allows populations to learn online faster and more robustly than the centralised and independent baselines. As we discuss below, a natural next step is of course moving to real-world problems and deployments, to find out how our work can be practically useful and to identify what obstacles still need to be removed.

7.2 Limitations and future work

7.2.1 Experimental and theoretical extensions

Our work follows the gold standard in MFGs by presenting experiments on grid world toy environments, albeit we show in Ch. 5 that our algorithms are able to handle larger and more complex games than prior work. Nevertheless, while these experiments demonstrate the advantages of our networked architecture, they still lack the complexity of the real-world applications to which we wish to address the approach. Thus future work lies in moving from these environments to real-world settings (where it may not be possible to reduce hyperparameter values to the same extent as we have demonstrated in our simpler experiments). However, there are also experimental gaps that could be filled even before this. Namely:

- While we demonstrate in Ch. 5 that our communication scheme affords faster learning when both the transition and reward functions depend on the mean field in non-tabular MFGs, our experiments in MFC and tabular MFGs have only the reward function depending on the mean field. As future work we could extend these experiments to transition functions that also depend on the mean field.

- In Ch. 5 we show that our networked architecture is beneficial in non-stationary MFGs; we could extend our experiments to include MFC problems with non-stationary solutions.

- In Ch. 4 we explore robustness to increases in population size and to scenarios where agents fail to update their policies by the time they are required to communicate in tabular MFGs, while in Ch. 6 we explore communication failures in MFC. As future work we could conduct complementary experiments in the other settings.

- In Chs. 4 and 5 our experiments only use one round of communication within each iteration, to show the benefit that even this can have on learning speed, while in Ch. 6 we include experiments with greater numbers of communication round. During our initial development we did test more communication rounds in the earlier settings too, but future work could include conducting this again more formally and reporting results. In more realistic environments it may be especially informative to study trade-offs in communication cost.

While our mean-field *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. While we do not expect that even larger populations than those we currently use would perform significantly differently in experiments, they would nevertheless be interesting to study empirically, since our algorithms facilitate precisely that.

There are also gaps that can be filled in the theoretical analysis. In Ch. 4 we compare the sample guarantees of our buffer-less networked algorithm with those of the central-agent and independent alternatives in tabular MFGs, in terms of numbers of communication round. Future work should consider updating these theoretical guarantees in light of our practical algorithmic enhancements (the replay buffer and the performance-based generation of σ values).

Similarly in Ch. 5 we prove that in non-tabular MFGs our networked algorithm can outperform the central-agent alternative (with implication that this also covers the independent-learning alternative), while in Ch. 6 we prove that our networked algorithm can outperform both the central-agent and independent alternative in MFC. We leave more general theoretical results for these latter two settings, such as proofs of convergence and sample complexity, for future work, as well as results when loosening our assumptions, for example regarding the accuracy of mean-field and average-reward estimates.

7.2.2 Enhancements to mean-field estimation and usage

In Ch. 5 we give our Alg. 5 for estimating the mean field in spatial environments (repeated as Alg. 9 for Ch. 6). It assumes that if a state s' is connected to s on the visibility graph $\mathcal{G}_t^{vis}$, an agent in s is able to *accurately* count all the agents in s', i.e. it either counts the exact total or cannot observe the state at all. We assume this for simplicity but it is not inherently the case, since a real-world agent may have only noisy observations even of others located nearby, due to imperfect sensors. We suggest two ways to deal with this.

Firstly, if agents share unique IDs as in Alg. 4 for the more general setting, then when communicating their vectors of collected IDs with each other via $\mathcal{G}_t^{comm}$, agents would gain the most accurate picture possible of all the agents that have been observed in a given state. However, as we note in Ch. 5, there are various reasons why sharing IDs might be undesirable, including privacy and scalability. If instead only counts are taken, and if the noise on each agent's count is assumed to be independent and, for example, subject to a Gaussian distribution, the algorithm can easily be updated such that communicating agents compute averages of their local and received counts to improve their accuracy, rather than simply using communication to fill in counts for previously unseen states. (Note that we can also consider the original case without noise as a special case of averaging, since averaging identical values equates to using the original value). Since the algorithm is intended to aid in local estimation of the mean-field distribution, which is inherently

approximate due to the uniform method for distributing the uncounted agents, we are not concerned with reaching exact consensus between agents on the communicated counts, so we do not require repeated averaging to ensure asymptotic convergence.

We may also wish to consider more sophisticated methods for distributing the uncounted agents across states, in place of the current uniform distribution. Such choices may be domain-specific based on knowledge of a particular environment. For example, one might use the counts to perform Bayesian updates on a specific prior, where this prior may relate to the estimated mean-field distribution at the previous time step $t-1$. If agents sought to learn to predict the *evolution* of the mean field based on their own policy or by learning a model, the Bayesian prior may also be based on forward prediction from the estimated mean-field distribution at $t-1$. Future work lies in conducting experiments in all of these more specific settings.

In grid-world settings such as those in our experiments, passing the (estimated or true global) mean-field distribution as a flat vector to the Q-network ignores the geometric structure of the problem. Perrin et al. [156] therefore proposes to create an embedding of the distribution by first passing the vector to a convolutional neural network (CNN), essentially treating the categorical distribution as an image. This technique is also followed in Wu et al. [153] (for their additional experiments, but not in the main body of their paper). During our development, we did integrate such a CNN setup into the Q-network, but were not able to find an architecture that reliably permitted learning. As future work we could try to resolve this, to see whether such a method increases the usefulness of observing the mean field in population-dependent policies, and therefore increases the importance of being able to accurately estimate the global mean field via Alg. 5 / Alg. 9.

7.2.3 Simplifying nested loops

Our networked algorithms, as well as the central-agent and independent baselines from Yardim et al. [15], 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. In Ch. 4 we show that

networked communication, in combination with the replay buffer, allows us to reduce the hyperparameter M_{td} (required by the theoretical algorithms) to 1, essentially removing the inner 'waiting' loop, which is not required at all in our deep learning algorithms in Chs. 5 and 6. Moreover, in Ch. 4 we show that our networked architecture provides redundancy and robustness in case of learning failures that may result from the necessities of synchronisation, which the alternative architectures lack. Similarly in Ch. 6 our ablation studies of the sub-routines, and our experiments on robustness to communication failures, indicate that synchronisation failure is not necessarily a problem in practice. Nevertheless, our algorithms still feature multiple loops, and future work lies in simplifying the algorithms further to aid practical implementation, possibly by techniques such as asynchronous communication [269].

7.2.4 Malfunctioning or adversarial communication

Since the MFG setting is non-cooperative, we have pre-empted conceptual 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 setting they inherently stand to benefit by exchanging their policies with others, as also in the MFC setting (though in Ch. 5 we see that the restriction to coordination games may not be necessary in any case).

Nevertheless, in real-world scenarios, the communication network could still be vulnerable to malfunctioning agents or adversarial actors poisoning the equilibrium/solution by broadcasting untrue policy information [270], or equally to unreliable communication channels. It is outside the scope of our work to analyse how much false information would have to be broadcast by what proportion of 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. [271], where 'power regularisation' of information flow is proposed to limit the adverse effects of communication by misaligned agents.

Bibliography

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

[2] 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.

[3] 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`.

[4] Haotian Gu, Xin Guo, Xiaoli Wei, and Renyuan Xu. Mean-Field Controls with Q-Learning for Cooperative MARL: Convergence and Complexity Analysis. *SIAM Journal on Mathematics of Data Science*, 3(4):1168–1196, 2021. doi: 10.1137/20M1360700. URL `https://doi.org/10.1137/20M1360700`.

[5] Washim Uddin Mondal, Mridul Agarwal, Vaneet Aggarwal, and Satish V. Ukkusuri. On the approximation of cooperative heterogeneous multi-agent reinforcement learning (MARL) using Mean Field Control (MFC). *J. Mach. Learn. Res.*, 23(1), January 2022. ISSN 1532-4435.

[6] Kai Cui, Christian Fabian, and Heinz Koeppl. Multi-agent reinforcement learning via mean field control: Common noise, major agents and approximation properties. 03 2023. doi: 10.48550/arXiv.2303.10665.

[7] Kai Cui, Sascha Hauck, Christian Fabian, and Heinz Koeppl. Learning Decentralized Partially Observable Mean Field Control for Artificial Collective Behavior. *arXiv preprint arXiv:2307.06175*, 2023.

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

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

[10] 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`.

[11] 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.

[12] Yufan Chen, Lan Wu, Renyuan Xu, and Ruixun Zhang. Periodic Trading Activities in Financial Markets: Mean-field Liquidation Game with Major-Minor Players. *arXiv preprint arXiv:2408.09505*, 2024.

[13] Erhan Bayraktar and Ali D Kara. Learning with Linear Function Approximations in Mean-Field Control. *arXiv preprint arXiv:2408.00991*, 2024.

[14] 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, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 11436–11447. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/xie21g.html.

[15] 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*, pages 39722–39754. PMLR, 2023.

[16] 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, page 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.

[17] 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*, pages 1–5, 2019.

[18] 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, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 20238–20248. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/e9bcd1b063077573285ae1a41025f5dc-Paper.pdf.

[19] 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.

[20] Mathieu Laurière, 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, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 12078–12095. PMLR, 17–23 Jul 2022. URL `https://proceedings.mlr.press/v162/lauriere22a.html`.

[21] 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`.

[22] Yueheng Li, Guangming Xie, and Zongqing Lu. Revisiting Cooperative Off-Policy Multi-Agent Reinforcement Learning. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu, editors, *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 36435–36450. PMLR, 13–19 Jul 2025. URL `https://proceedings.mlr.press/v267/li25dc.html`.

[23] Batuhan Yardim and Niao He. Exploiting Approximate Symmetry for Efficient Multi-Agent Reinforcement Learning. *arXiv preprint arXiv:2408.15173*, 2024.

[24] Sihan Zeng, Sujay Bhatt, Alec Koppel, and Sumitra Ganesh. A Single-Loop Finite-Time Convergent Policy Optimization Algorithm for Mean Field Games (and Average-Reward Markov Decision Processes). *arXiv e-prints*, pages arXiv–2408, 2024.

[25] 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.

[26] 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, 2020.

[27] 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.

[28] 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)*, pages 1–6. IEEE, 2019.

[29] 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.

[30] 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.

[31] 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.

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

[33] 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.

[34] 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.

[35] 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.

[36] 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.

[37] Rinel Foguen Tchuendom, Roland Malhamé, and Peter E. Caines. On a class of linear quadratic Gaussian quantilized mean field games. *Automatica*, 170:111878, 2024. ISSN 0005-1098. doi: https://doi.org/10.1016/j.automatica.2024.111878. URL https://www.sciencedirect.com/science/article/pii/S0005109824003728.

[38] Dirk Becherer and Stefanie Hesse. Common Noise by Random Measures: Mean-Field Equilibria for Competitive Investment and Hedging. *arXiv preprint arXiv:2408.01175*, 2024.

[39] Jingguo Zhang and Lianhai Ren. A mean field game model of green economy. *Digital Finance*, pages 1–36, 2024.

[40] Fan Chen, Nicholas Martin, Po-Yu Chen, Xiaozhen Wang, Zhenjie Ren, and Francois Buet-Golfouse. Deciding Bank Interest Rates–A Major-Minor Impulse Control Mean-Field Game Perspective. *arXiv preprint arXiv:2411.14481*, 2024.

[41] Martino Bernasconi, E. Vittori, F. Trovò, and M. Restelli. Dealer markets: A reinforcement learning mean field game approach. *The North American Journal of Economics and Finance*, 68:101974, 2023. ISSN 1062-9408. doi: https://doi.org/10.1016/j.najef.2023.101974. URL `https://www.sciencedirect.com/science/article/pii/S1062940823000979`.

[42] Alekos Cecchin, Markus Fischer, Claudio Fontana, and Giacomo Lanaro. Weak equilibria of a mean-field market model under asymmetric information. *arXiv preprint arXiv:2504.09356*, 2025.

[43] Lijun Bo, Yijie Huang, and Xiang Yu. Mean Field Game of Optimal Tracking Portfolio. *arXiv preprint arXiv:2505.01858*, 2025.

[44] Guanxing Fu and Ulrich Horst. Mean Field Portfolio Games with Epstein-Zin Preferences. *arXiv preprint arXiv:2505.07231*, 2025.

[45] Xu Wang, Samy Wu Fung, and Levon Nurbekyan. A primal-dual price-optimization method for computing equilibrium prices in mean-field games models. *arXiv preprint arXiv:2506.04169*, 2025.

[46] Benjamin Moll and Lenya Ryzhik. Mean Field Games without Rational Expectations. *arXiv preprint arXiv:2506.11838*, 2025.

[47] Rinel Foguen Tchuendom, Dena Firoozi, and Michèle Breton. Ranking quantilized mean-field games with an application to early-stage venture investments. *arXiv preprint arXiv:2507.00853*, 2025.

[48] Bing-Chang Wang. Mean field hierarchical control for production output adjustment with noisy sticky prices. *Automatica*, 176:112260, 2025. ISSN 0005-1098. doi: https://doi.org/10.1016/j.automatica.2025.112260. URL `https://www.sciencedirect.com/science/article/pii/S0005109825001529`.

[49] Na Li, Yilin Wei, and Qingfeng Zhu. Stochastic Linear-Quadratic Mean-Field Games of Controls for Delayed Systems with Jump Diffusion. *Journal of Optimization Theory and Applications*, 206(3):66, 2025. doi: 10.1007/s10957-025-02730-4. URL `https://doi.org/10.1007/s10957-025-02730-4`.

[50] Burak Aydin, Emre Parmaksiz, and Ronnie Sircar. Fare Game: A Mean Field Model of Stochastic Intensity Control in Dynamic Ticket Pricing. *arXiv preprint arXiv:2506.13088*, 2025.

[51] Luca Grosset and Elena Sartori. Mean-Field Modeling of Green Technology Adoption: A Competition for Incentives. *Mathematics*, 13(5), 2025. ISSN 2227-7390. doi: 10.3390/math13050691. URL `https://www.mdpi.com/2227-7390/13/5/691`.

[52] Anna Aksamit, Kaustav Das, Ivan Guo, Kihun Nam, and Zhou Zhou. Switching to a Green and sustainable finance setting: a mean field game approach. *arXiv preprint arXiv:2503.06967*, 2025.

[53] Chen Feng and Andrew L Liu. Decentralized Integration of Grid Edge Resources into Wholesale Electricity Markets via Mean-field Games. *arXiv preprint arXiv:2503.07984*, 2025.

[54] Jun He and Andrew L Liu. A Hybrid Mean Field Framework for Aggregators Participating in Wholesale Electricity Markets. *arXiv preprint arXiv:2507.03240*, 2025.

[55] 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.

[56] Zhaobin Mo, Xu Chen, Xuan Di, Elisa Iacomini, Chiara Segala, Michael Herty, and Mathieu Lauriere. A game-theoretic framework for generic second-order traffic flow models using mean field games and adversarial inverse reinforcement learning. *Transportation Science*, 58(6):1403–1426, 2024.

[57] Naman Krishna Pande, Arun Kumar, and Arvind Kumar Gupta. Generative adversarial modelling of traffic flow via second-order mean field games with stochastic driving attributes. 2025.

[58] Zijia Niu, Wang Yao, Yuxin Jin, Sanjin Huang, Xiao Zhang, and Langyu Qian. Integrated Task Assignment and Trajectory Planning for a Massive Number of Agents Based on Bilayer-Coupled Mean Field Games. *IEEE Transactions on Automation Science and Engineering*, 22:1833–1852, 2025. doi: 10.1109/TASE.2024.3370619.

[59] Xu Chen, Shuo Liu, and Xuan Di. Bridging Agent Dynamics and Population Behaviors: Scalable Learning for Mean Field Games on Graph via Neural Operators. AAAI, 2024.

[60] 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.

[61] Yunpeng Li, Antonis Dimakis, and Costas A Courcoubetis. Repositioning, Ride-matching, and Abandonment in On-demand Ride-hailing Platforms: A Mean Field Game Approach. *arXiv preprint arXiv:2504.02346*, 2025.

[62] 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, page 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`.

[63] Jehad Hedel and Nga Nguyen. Price Coordination for Electric Vehicle Fleet Using Mean Field Game Theory. In *2024 56th North American Power Symposium (NAPS)*, pages 1–6, 2024. doi: 10.1109/NAPS61145.2024.10741829.

[64] Jehad Hedel and Nga Nguyen. Optimal Charging Control for Electric Vehicle Fleet Using Mean Field Game Theory. In *2025 IEEE Texas Power and Energy Conference (TPEC)*, pages 1–6, 2025. doi: 10.1109/TPEC63981.2025.10907167.

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

[66] Nicolas Garcia, Ronnie Sircar, and H Mete Soner. Mean Field Games of Control and Cryptocurrency Mining. *arXiv preprint arXiv:2504.15526*, 2025.

[67] 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.

[68] 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*, pages 1–1, 2024. doi: 10.1109/JIOT.2024.3387090.

[69] 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`.

[70] Shubham Aggarwal, Melih Bastopcu, Sennur Ulukus, Tamer Başar, et al. Distributed Offloading in Multi-Access Edge Computing Systems: A Mean-Field Perspective. *arXiv preprint arXiv:2501.18718*, 2025.

[71] 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, pages 1–12, New Orleans, LA, USA, 2022. Curran Associates, Inc.

[72] Yuhan Kang, Hao Gao, and Zhu Han. *Mean Field Game Guided Deep Reinforcement Learning*, pages 75–90. Springer Nature Switzerland, Cham, 2025. ISBN 978-3-031-91859-9. doi: 10.1007/978-3-031-91859-9_5. URL `https://doi.org/10.1007/978-3-031-91859-9_5`.

[73] 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.

[74] 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)*, pages 338–343, 2022. doi: 10.1109/IPCCC55026.2022.9894296.

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

[76] Hao Gao, Yongkang Liu, Emrah Akin Sisbot, Yashar Zeinali Farid, Kentaro Oguchi, and Zhu Han. Hierarchical Federated Learning with Mean Field Game Device Selection for Connected Vehicle Applications. In *2023 IEEE Intelligent Vehicles Symposium (IV)*, pages 1–6, 2023. doi: 10.1109/IV55152.2023.10186687.

[77] Haibo Wang, Hongwei Gao, Pai Jiang, Matthieu De Mari, Panzer Gu, and Yinsheng Liu. Mean Field-based Dynamic Backoff Optimization for MIMO-enabled Grant-Free NOMA in Massive IoT Networks. *arXiv preprint arXiv:2410.12497*, 2024.

[78] Xuesong Wu, Tianshuai Zheng, Runfang Wu, Jie Ren, Junyan Guo, and Ye Du. Hi-SAM: A high-scalable authentication model for satellite-ground Zero-Trust system using mean field game. *arXiv preprint arXiv:2408.06185*, 2024.

[79] Runchen Xu, Zheng Chang, Zhu Han, Sahil Garg, Georges Kaddoum, and Joel J. P. C. Rodrigues. Energy-Efficient Joint Optimization of Sensing and Computation in MEC-Assisted IoT Using Mean-Field Game. *IEEE Internet of Things Journal*, 11(23):37857–37871, 2024. doi: 10.1109/JIOT.2024.3443701.

[80] Hongyi Yang, Jingzhi Liu, Geng Li, Jianming Zhang, Ling Jiang, and Shoulian Yang. Distributed Intelligent Power Distribution Optimization Method Based on Mean Field Game Theory. In *2025 IEEE 5th International Conference on Power, Electronics and Computer Applications (ICPECA)*, pages 818–822, 2025. doi: 10.1109/ICPECA63937.2025.10928821.

[81] Stéphane Le Ménec. Swarm Guidance Based on Mean Field Game Concepts. *International Game Theory Review*, page 2440008, 2024.

[82] Yangqi Lei, Quan Quan, and Zhikun She. Mean-Field-Based Density Control for Swarm Robotics Passing-Through a Virtual Tube. *IEEE Control Systems Letters*, 8:3500–3505, 2024. doi: 10.1109/LCSYS.2025.3550036.

[83] Yu Bai, Di Zhou, and Zhen He. Optimal Pursuit Strategies in Missile Interception: Mean Field Game Approach. *Aerospace*, 12(4), 2025. ISSN 2226-4310. doi: 10.3390/aerospace12040302. URL `https://www.mdpi.com/2226-4310/12/4/302`.

[84] 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)*, pages 195–200, 2020. doi: 10.1109/WCSP49889.2020.9299742.

[85] 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*, pages 1–1, 2024. doi: 10.1109/COMST.2024.3393369.

[86] Yue Xu, Linjiang Zheng, Xiao Wu, Yi Tang, Weining Liu, and Dihua Sun. Joint Resource Allocation for UAV-Assisted V2X Communication With Mean Field Multi-Agent Reinforcement Learning. *IEEE Transactions on Vehicular Technology*, 74(1):1209–1223, 2025. doi: 10.1109/TVT.2024.3466116.

[87] Zejian Zhou, Lijun Qian, and Hao Xu. Decentralized Multi-agent Reinforcement Learning for Large-scale Mobile Wireless Sensor Network Control Using Mean Field Games. In *2024 33rd International Conference on Computer Communications and Networks (ICCCN)*, pages 1–6, 2024. doi: 10.1109/ICCCN61486.2024.10637582.

[88] Chenyu You, Mengru Cai, Shan Yin, Honglei Wang, and Shanguo Huang. Latency-Aware Mean Field Game-Based Task Offloading Strategy in Metro Optical Networks. In *2024 Asia Communications and Photonics Conference (ACP) and International Conference on Information Photonics and Optical Communications (IPOC)*, pages 1–4, 2024. doi: 10.1109/ACP/IPOC63121.2024.10809790.

[89] Yuhan Kang, Hao Gao, and Zhu Han. *Incentive Mechanism Design in Satellite-Based Federated Learning Using Mean Field Evolutionary Approach*, pages 91–114. Springer Nature Switzerland, Cham, 2025. ISBN 978-3-031-91859-9. doi: 10.1007/978-3-031-91859-9_6. URL `https://doi.org/10.1007/978-3-031-91859-9_6`.

[90] Salah Eddine Choutri, Boualem Djehiche, Prajwal Chauhan, and Saif Eddin Jabari. Backpressure-based Mean-field Type Game for Scheduling in Multi-Hop Wireless Sensor Networks. *arXiv preprint arXiv:2506.03059*, 2025.

[91] 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*, pages 1–14, 2024. doi: 10.1109/TVT.2024.3394235.

[92] Yuhan Kang, Hao Gao, and Zhu Han. *Opinion Evolution in Social Networks: Use Generative Adversarial Networks to Solve Mean Field Game*, pages 29–47. Springer Nature Switzerland, Cham, 2025. ISBN 978-3-031-91859-9. doi: 10.1007/978-3-031-91859-9_3. URL `https://doi.org/10.1007/978-3-031-91859-9_3`.

[93] A. I. Glukhov, M. A. Shishlenin, and N. V. Trusov. Modelling the Dynamics of Social Protests: Mean-Field Games and Inverse Problems. *Differencial'nye uravneniya*, 61(6):802–822, 2025. ISSN 0374-0641. URL `https://clinpractice.ru/0374-0641/article/view/685643`.

[94] 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.

[95] Gianmarco Del Sarto, Marta Leocata, and Giulia Livieri. A Mean Field Game approach for pollution regulation of competitive firms. *arXiv preprint arXiv:2407.12754*, 2024.

[96] 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`.

[97] Gokce Dayanikli and Mathieu Lauriere. Cooperation, competition, and common pool resources in mean field games. *arXiv preprint arXiv:2504.09043*, 2025.

[98] Dantong Chu, Kenneth Tsz Hin Ng, Sheung Chi Phillip Yam, and Harry Zheng. Mean field analysis of two-party governance: Competition versus cooperation among leaders. *Automatica*, 173:112028, 2025.

[99] 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*, pages 10178–10206. PMLR, 2023.

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

[101] 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`.

[102] 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`.

[103] 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.

[104] 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*, pages 249–307, 2020.

[105] 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.

[106] 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, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL `https://proceedings.neurips.cc/paper_files/paper/2019/file/030e65da2b1c944090548d36b244b28d-Paper.pdf`.

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

[108] 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, page 251–259, Richland, SC, 2019. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9781450363099.

[109] 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.

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

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

[112] 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`.

[113] 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.

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

[115] 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`.

[116] Han Huang and Rongjie Lai. Unsupervised solution operator learning for mean-field games. *Journal of Computational Physics*, 537:114057, 2025. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2025.114057. URL `https://www.sciencedirect.com/science/article/pii/S0021999125003407`.

[117] 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.

[118] 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, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL `https://proceedings.neurips.cc/paper_files/paper/2019/file/030e65da2b1c944090548d36b244b28d-Paper.pdf`.

[119] Massimo Fornasier and Francesco Solombrino. Mean-Field Optimal Control. *ESAIM: Control, Optimisation and Calculus of Variations*, 20(4):1123–1152, 2014. doi: 10.1051/cocv/2014009.

[120] René Carmona, Mathieu Laurière, and Zongjun Tan. Linear-quadratic mean-field reinforcement learning: convergence of policy gradient methods. *arXiv preprint arXiv:1910.04295*, 2019.

[121] Daisuke Inoue, Yuji Ito, Takahito Kashiwabara, Norikazu Saito, and Hiroaki Yoshida. Partially Centralized Model-Predictive Mean Field Games for controlling multi-agent systems. *IFAC Journal of Systems and Control*, 24:100217, 2023. ISSN 2468-6018. doi: https://doi.org/10.1016/j.ifacsc.2023.100217. URL `https://www.sciencedirect.com/science/article/pii/S2468601823000032`.

[122] Bhavini Jeloka, Yue Guan, and Panagiotis Tsiotras. Learning Large-Scale Competitive Team Behaviors with Mean-Field Interactions. *arXiv preprint arXiv:2504.21164*, 2025.

[123] Zeyu Yang and Yongsheng Song. On Discounted Infinite-Time Mean Field Games. *arXiv preprint arXiv:2505.15131*, 2025.

[124] Lars Ruthotto, Stanley J. Osher, Wuchen Li, Levon Nurbekyan, and Samy Wu Fung. A machine learning framework for solving high-dimensional mean field game and mean field control problems. *Proceedings of the National Academy of Sciences*, 117(17):9183–9193, 2020. doi: 10.1073/pnas.1922204117. URL `https://www.pnas.org/doi/abs/10.1073/pnas.1922204117`.

[125] 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.

[126] 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.

[127] Taeyoung Lee et al. Mean Field Game and Control for Switching Hybrid Systems. *arXiv preprint arXiv:2412.10522*, 2024.

[128] Robert Denkert, Idris Kharroubi, and Huyên Pham. A randomisation method for mean-field control problems with common noise. *arXiv preprint arXiv:2412.20782*, 2024.

[129] 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, page 9672–9683, Red Hook, NY, USA, 2018. Curran Associates Inc.

[130] 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, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning*

Research, pages 5872–5881. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/zhang18n.html.

[131] 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, 2021.

[132] Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. *"Multi-Agent Reinforcement Learning: A Selective Overview of Theories and Algorithms"*, pages 321–384. Springer International Publishing, Cham, 2021. 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.

[133] 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.

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

[135] Kun Xu, Yue Li, Jun Sun, Shuyuan Du, Xinpeng Di, Yuguang Yang, and Bo Li. Targets capture by distributed active swarms via bio-inspired reinforcement learning. *Science China Physics, Mechanics & Astronomy*, 68(1):1–12, 2025.

[136] Bernard T. Agyeman, Benjamin Decardi-Nelson, Jinfeng Liu, and Sirish L. Shah. A semi-centralized multi-agent RL framework for efficient irrigation scheduling. *Control Engineering Practice*, 155:106183, 2025. ISSN 0967-0661. doi: https://doi.org/10.1016/j.conengprac.2024.106183. URL https://www.sciencedirect.com/science/article/pii/S0967066124003423.

[137] Jiří Horyna, Roland Jung, Stephan Weiss, Eliseo Ferrante, and Martin Saska. Swarming Without an Anchor (SWA): Robot Swarms Adapt Better to Localization Dropouts Then a Single Robot. *IEEE Robotics and Automation Letters*, 10(6): 6207–6214, 2025. doi: 10.1109/LRA.2025.3562786.

[138] 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.

[139] 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)*, pages 3548–3553, 2015. doi: 10.1109/ECC.2015.7331083.

[140] 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)*, pages 4412–4417, 2015. doi: 10.1109/CDC.2015.7402908.

[141] 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.

[142] Bora Yongacoglu, Gürdal Arslan, and Serdar Yüksel. Mean-field games with finitely many players: independent learning and subjectivity. *Journal of Machine Learning Research*, 25(419):1–69, 2024.

[143] 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)*, pages 2845–2850, 2022. doi: 10.1109/CDC51059.2022.9992399.

[144] Min Li, Tianyang Nie, Shujun Wang, and Ke Yan. Incomplete Information Mean-Field Games and Related Riccati Equations. *Journal of Optimization Theory and Applications*, pages 1–22, 2024.

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

[146] 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, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1626–1635. PMLR, 09–15 Jun 2019. URL `https://proceedings.mlr.press/v97/doan19a.html`.

[147] 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)*, pages 5562–5567, 2019. doi: 10.1109/CDC40024.2019.9029257.

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

[149] 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.

[150] Yunpeng Li, Antonis Dimakis, and Costas A Courcoubetis. On the Effect of Time Preferences on the Price of Anarchy. *arXiv preprint arXiv:2504.20774*, 2025.

[151] Yohance AP Osborne and Iain Smears. Rates of convergence of finite element approximations of second-order mean field games with nondifferentiable Hamiltonians. *arXiv preprint arXiv:2506.03039*, 2025.

[152] Patrick Benjamin and Alessandro Abate. Networked Communication for Decentralised Agents in Mean-Field Games. *arXiv preprint arXiv:2306.02766*, 2023.

[153] 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*, 2024.

[154] René Carmona, François Delarue, and Daniel Lacker. Mean field games with common noise. *The Annals of Probability*, 44(6):3740–3803, 2016. ISSN 00911798. URL `http://www.jstor.org/stable/44072057`.

[155] Pierre Cardaliaguet, François Delarue, Jean-Michel Lasry, and Pierre-Louis Lions. The master equation and the convergence problem in mean field games, 2015. URL `https://arxiv.org/abs/1509.02505`.

[156] 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, pages 9413–9421, 2022.

[157] Daniel S. Bernstein, Robert Givan, Neil Immerman, and Shlomo Zilberstein. The Complexity of Decentralized Control of Markov Decision Processes. *Mathematics of Operations Research*, 27(4):819–840, 2002. ISSN 0364765X, 15265471. URL `http://www.jstor.org/stable/3690469`.

[158] Muhammad Aneeq Uz Zaman, Mathieu Lauriere, Alec Koppel, and Tamer Başar. Robust cooperative multi-agent reinforcement learning: A mean-field type game perspective. In *6th Annual Learning for Dynamics & Control Conference*, pages 770–783. PMLR, 2024.

[159] Lorenzo Magnino, Yuchen Zhu, and Mathieu Lauriere. Learning to Stop: Deep Learning for Mean Field Optimal Stopping. In *Forty-second International Conference on Machine Learning*, 2025.

[160] P Jameson Graber. A "trembling hand perfect" equilibrium for a certain class of mean field games. *arXiv preprint arXiv:2506.11868*, 2025.

[161] Batuhan Yardim, Semih Cayci, and Niao He. A Variational Inequality Approach to Independent Learning in Static Mean-Field Games. *ACM/IMS Journal of Data Science*, 2025.

[162] Felix Höfer, H Mete Soner, and Atilla Yılmaz. Markov Perfect Equilibria in Discrete Finite-Player and Mean-Field Games. *arXiv preprint arXiv:2507.04540*, 2025.

[163] Yu Si and Jingtao Shi. Decentralized Strategies for Backward Linear-Quadratic Mean Field Games and Teams. *Optimal Control Applications and Methods*, 2025.

[164] Gökçe Dayanikli, Mathieu Laurière, and Jiacheng Zhang. Deep Learning for Population-Dependent Controls in Mean Field Control Problems with Common Noise. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, pages 2231–2233, 2024.

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

[166] Hui Huang and Jethro Warnett. Well-posedness and mean-field limit estimate of a consensus-based algorithm for multiplayer games. *arXiv preprint arXiv:2505.13632*, 2025.

[167] Rita Ferreira, Diogo Gomes, and Melih Ucer. Solving Mean-Field Games with Monotonicity Methods in Banach Spaces. *arXiv preprint arXiv:2506.21212*, 2025.

[168] Razvan-Andrei Lascu and Mateusz B Majka. Non-convex entropic mean-field optimization via Best Response flow. *arXiv preprint arXiv:2505.22760*, 2025.

[169] Meijiao Wang, Maoning Tang, Qiuhong Shi, and Qingxin Meng. A Variational Formula of Forward-Backward Stochastic Differential System of Mean-Field Type with Observation Noise and Some Application. *Communications on Applied Mathematics and Computation*, pages 1–18, 2024.

[170] Tian Chen, Kai Du, and Zhen Wu. Partially observed mean-field game and related mean-field forward-backward stochastic differential equation. *Journal of Differential Equations*, 408:409–448, 2024. ISSN 0022-0396. doi: https://doi.org/10.1016/j.jde.2024.07.014. URL `https://www.sciencedirect.com/science/article/pii/S0022039624004364`.

[171] Lu Ren, Yuxin Jin, Zijia Niu, Wang Yao, and Xiao Zhang. Hierarchical Cooperation in LQ Multi-Population Mean Field Game With Its Application to Opinion Evolution. *IEEE Transactions on Network Science and Engineering*, 11(5): 5008–5022, 2024. doi: 10.1109/TNSE.2024.3418832.

[172] Yu Si and Jingtao Shi. Backward Linear-Quadratic Mean Field Stochastic Differential Games: A Direct Method. *arXiv preprint arXiv:2411.18891*, 2024.

[173] Salvatore Federico, Fausto Gozzi, and Andrzej Święch. On Mean Field Games in Infinite Dimension. *arXiv preprint arXiv:2411.14604*, 2024.

[174] Wei Sun and Theodore B. Trafalis. Risk-aware controller implementation for risk-sensitive mean field games through a game-theoretic differential dynamic programming approach. *International Journal of Control*, 0(0):1–9, 2025. doi: 10.1080/00207179.2025.2491820. URL `https://doi.org/10.1080/00207179.2025.2491820`.

[175] Bing-Chang Wang, Juanjuan Xu, Huanshui Zhang, and Yong Liang. Linear Quadratic Mean Field Stackelberg Games: Open-loop and Feedback Solutions. *arXiv preprint arXiv:2504.09401*, 2025.

[176] Xianjin Yang and Jingguo Zhang. Gaussian Process Policy Iteration with Additive Schwarz Acceleration for Forward and Inverse HJB and Mean Field Game Problems. *arXiv preprint arXiv:2505.00909*, 2025.

[177] Olav Ersland, Espen Robstad Jakobsen, and Alessio Porretta. Long time behaviour of Mean Field Games with fractional diffusion. *arXiv preprint arXiv:2505.06183*, 2025.

[178] Ramen Ghosh. Mean-Field Games for Coordinated Exploration in Dynamic Environments. 2025.

[179] Zhongyuan Cao and Mathieu Laurière. Probabilistic Analysis of Graphon Mean Field Control. *arXiv preprint arXiv:2505.19664*, 2025.

[180] E. Everardo Martinez-Garcia, Fernando Luque-Vásquez, and J. Adolfo Minjárez-Sosa. Statistical Estimation of Mean-Field Equilibria in a Class of Discounted Mean-Field Games. *Applied Mathematics & Optimization*, 91(3):73, 2025. doi: 10.1007/s00245-025-10273-3. URL https://doi.org/10.1007/s00245-025-10273-3.

[181] Manfred Opper and Sebastian Reich. On a mean-field Pontryagin minimum principle for stochastic optimal control. *arXiv preprint arXiv:2506.10506*, 2025.

[182] Elisabetta Carlini and Valentina Coscetti. A semi-Lagrangian scheme for First-Order Mean Field Games based on monotone operators. *arXiv preprint arXiv:2506.10509*, 2025.

[183] Philipp Plank and Yufei Zhang. Policy Optimization for Continuous-time Linear-Quadratic Graphon Mean Field Games. *arXiv preprint arXiv:2506.05894*, 2025.

[184] Tianjiao Hua and Peng Luo. Extended mean field games with terminal constraint via decoupling fields. *arXiv preprint arXiv:2506.07485*, 2025.

[185] Tian Chen, Hongyu Shi, and Zhen Wu. A Progressive Maximum Principle of Fully Coupled Mean-Field System with Jumps. *Journal of Optimization Theory and Applications*, 206(3):75, 2025. doi: 10.1007/s10957-025-02760-y. URL https://doi.org/10.1007/s10957-025-02760-y.

[186] Shawon Dey and Hao Xu. Extended mean field game theoretical optimal distributed control for large scale multi-agent systems: An efficiency-complexity tradeoff. *Information Sciences*, 719:122432, 2025. ISSN 0020-0255. doi: https://doi.org/10.1016/j.ins.2025.122432. URL https://www.sciencedirect.com/science/article/pii/S002002552500564X.

[187] F. A. Fedorov. Studying the well-posedness of the boundary value problem for a system of riccati type equations based on the concept of mean field games. *Moscow University Computational Mathematics and Cybernetics*, 49(2):150–164, 2025. doi: 10.3103/S0278641925700086. URL https://doi.org/10.3103/S0278641925700086.

[188] Juan Li, Yanwei Li, and Wenliang Wang. Mean-field backward stochastic differential equations with random terminal time. *Journal of Mathematical Analysis and Applications*, 553(1):129830, 2026. ISSN 0022-247X. doi: https://doi.org/10.1016/j.jmaa.2025.129830. URL https://www.sciencedirect.com/science/article/pii/S0022247X25006110.

[189] Na Xiang and Jingtao Shi. Robust Incentive Stackelberg Mean Field Stochastic Linear-Quadratic Differential Game with Model Uncertainty. *arXiv preprint arXiv:2507.04585*, 2025.

[190] Ramen Ghosh. Federated Mean-Field Learning with Fairness Constraints: An Optimal Transport Game-Theoretic Approach. 2025.

[191] Ruimin Xu, Kaiyue Dong, Jingyu Zhang, and Ying Zhou. Linear-quadratic-Gaussian mean-field games driven by Poisson jumps with major and minor agents. *AIMS MATHEMATICS*, 10(5):11086–11110, 2025.

[192] 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`.

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

[194] 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.

[195] 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.

[196] 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*, 2024.

[197] 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, editors, *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pages 289–297. PMLR, 02–04 May 2024. URL `https://proceedings.mlr.press/v238/huang24a.html`.

[198] Julian Barreiro-Gomez and Shinkyu Park. Optimal Strategy Revision in Population Games: A Mean Field Game Theory Perspective. *arXiv preprint arXiv:2501.01389*, 2025.

[199] 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`.

[200] 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*, pages 361–372. Springer, 2023.

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

[202] 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)*, pages 1032–1037, 2020. doi: 10.1109/CDC42340.2020.9304340.

[203] 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`.

[204] 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.

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

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

[207] 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.

[208] Chenyu Zhang, Xu Chen, and Xuan Di. Stochastic Semi-Gradient Descent for Learning Mean Field Games with Population-Aware Function Approximation. *arXiv preprint arXiv:2408.08192*, 2024.

[209] Yan Chen, Tao Li, and Nan Qiao. Sampled-data based adaptive mean field games for leader-follower stochastic multi-agent systems. *Mathematical Control and Related Fields*, pages 0–0, 2024.

[210] Xu Chen, Shuo Liu, and Xuan Di. A Hybrid Framework of Reinforcement Learning and Physics-Informed Deep Learning for Spatiotemporal Mean Field Games. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '23, page 1079–1087, Richland, SC, 2023. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9781450394321.

[211] 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, pages 9616–9625, 2024.

[212] Jiajia Yu, Xiuyuan Cheng, Jian-Guo Liu, and Hongkai Zhao. Convergence Analysis and Acceleration of Fictitious Play for General Mean-Field Games via the Best Response. *arXiv preprint arXiv:2411.07989*, 2024.

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

[214] Yu Si and Jingtao Shi. General Linear-Quadratic Mean Field Stochastic Differential Game with Common Noise: a Direct Method. *arXiv preprint arXiv:2506.16779*, 2025.

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

[216] 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`.

[217] 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*, pages 1–8, 2024. doi: 10.1109/LRA.2024.3389813.

[218] 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.

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

[220] 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.

[221] 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, editors, *Proceedings of 3rd 2023 International Conference on Autonomous Unmanned Systems (3rd ICAUS 2023)*, pages 88–96, Singapore, 2024. Springer Nature Singapore. ISBN 978-981-97-1087-4.

[222] 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.

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

[224] 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.

[225] 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`.

[226] 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, 2018. doi: 10.1109/ACCESS.2018.2799138.

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

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

[229] 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, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5571–5580. PMLR, 10–15 Jul 2018. URL `https://proceedings.mlr.press/v80/yang18d.html`.

[230] Sriram 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*, pages 537–545, 2021.

[231] Sriram 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*, pages 411–419, 2020.

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

[233] Agoston Eiben and James Smith. *What Is an Evolutionary Algorithm?*, pages 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`.

[234] Rajeshwari Sissodia, ManMohan Singh Rauthan, Varun Barthwal, and Vinay Dwivedi. Evolutionary Algorithms for Optimization and Swarm Intelligence-Based Optimization. In *Optimization Tools and Techniques for Enhanced Computational Efficiency*, pages 17–42. IGI Global Scientific Publishing, 2025.

[235] 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.

[236] 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, page 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.

[237] 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, page 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.

[238] 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, editors, *Swarm Intelligence*, pages 395–402, Cham, 2018. Springer International Publishing. ISBN 978-3-030-00533-7.

[239] 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, editors, *Applications of Evolutionary Computation*, pages 567–582, Cham, 2019. Springer International Publishing. ISBN 978-3-030-16692-2.

[240] 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, page 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.

[241] 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.

[242] 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, editors, *Swarm Intelligence*, pages 351–359, Cham, 2018. Springer International Publishing.

[243] 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.

[244] 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`.

[245] Max Jaderberg, Valentin Dalibard, Simon Osindero, Wojciech M Czarnecki, Jeff Donahue, Ali Razavi, Oriol Vinyals, Tim Green, Iain Dunning, Karen Simonyan, et al. Population based training of neural networks. *arXiv preprint arXiv:1711.09846*, 2017.

[246] 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.

[247] Javad Soleimani, Reza Farhangi, and Gunes Karabulut Kurt. Distributed Critic-Based Neuro-Fuzzy Learning in Swarm Autonomous Vehicles. In *2024 IEEE 100th Vehicular Technology Conference (VTC2024-Fall)*, pages 1–6, 2024. doi: 10.1109/VTC2024-Fall63153.2024.10757965.

[248] 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`.

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

[250] Yulong Lu and Pierre Monmarché. Convergence of time-averaged mean field gradient descent dynamics for continuous multi-player zero-sum games. *arXiv preprint arXiv:2505.07642*, 2025.

[251] 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, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 20580–20603. PMLR, 17–23 Jul 2022. URL `https://proceedings.mlr.press/v162/su22b.html`.

[252] 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`.

[253] 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, page 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`.

[254] 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`.

[255] 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*, pages 1–7, 2009. doi: 10.1109/ICAT.2009.5348437.

[256] 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`.

[257] 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.

[258] 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.

[259] 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, page 1028–1037, Richland, SC, 2022. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9781450392136.

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

[261] Saeed Hadikhanloo. Learning in anonymous nonatomic games with applications to first-order mean field games. *arXiv preprint arXiv:1704.00378*, 2017.

[262] Jincun Liu, Yinjie Ren, Yang Liu, Yan Meng, Dong An, and Yaoguang Wei. Achievement of fish school milling motion based on distributed multi-agent reinforcement learning. *Journal of Bionic Engineering*, pages 1–19, 2025.

[263] Breno Cunha Queiroz and Daniel MacRae. Occlusion-based object transportation around obstacles with a swarm of miniature robots. *Swarm Intelligence*, pages 1–29, 2024.

[264] Mohammad Amin Tajeddini, Hamed Kebriaei, and Luigi Glielmo. Robust decentralised mean field control in leader following multi-agent systems. *IET Control Theory & Applications*, 11(16):2707–2715, 2017.

[265] Hesam Farzaneh, Mohammad Shokri, Hamed Kebriaei, and Farrokh Aminifar. Robust Energy Management of Residential Nanogrids via Decentralized Mean Field Control. *IEEE Transactions on Sustainable Energy*, 11(3):1995–2002, 2020. doi: 10.1109/TSTE.2019.2949016.

[266] 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.

[267] Changling Li and Ying Li. Scaling up Energy-Aware Multi-Agent Reinforcement Learning for Mission-Oriented Drone Networks With Individual Reward. *IEEE Internet of Things Journal*, pages 1–1, 2024. doi: 10.1109/JIOT.2024.3511253.

[268] Leo Cazenille, Maxime Toquebiau, Nicolas Lobato-Dauzier, Alessia Loi, Loona Macabre, Nathanaël Aubert-Kato, Anthony J Genot, and Nicolas Bredeche. Signalling and social learning in swarms of robots. *Philosophical Transactions A*, 383(2289):20240148, 2025.

[269] 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*, pages 1–12, 2024. doi: 10.1109/TSIPN.2024.3384817.

[270] Swadhin Agrawal, Jitesh Jhawar, Andreagiovanni Reina, Sujit P Baliyarasimhuni, Heiko Hamann, and Liang Li. Impact of Individual Defection on Collective Motion. In *International Conference on Swarm Intelligence*, pages 127–140. Springer, 2024.

[271] 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.