Networked Communication for Mean-Field Games with Function Approximation and Empirical Mean-Field Estimation

Patrick Benjamin
University of Oxford
Oxford, United Kingdom
phlbenjamin@gmail.com

Alessandro Abate
University of Oxford
Oxford, United Kingdom
alessandro.abate@cs.ox.ac.uk

ABSTRACT

Recent algorithms allow decentralised agents, possibly connected via a communication network, to learn equilibria in mean-field games from a non-episodic run of the empirical system. However, these algorithms are for tabular settings: this computationally limits the size of agents' observation space, meaning the algorithms cannot handle anything but small state spaces, nor generalise beyond policies depending only on the agent's local state to so-called 'population-dependent' policies. We address this limitation by introducing function approximation to the existing setting, drawing on the Munchausen Online Mirror Descent method that has previously been employed only in finite-horizon, episodic, centralised settings. While this permits us to include the mean field in the observation for players' policies, it is unrealistic to assume decentralised agents have access to this global information: we therefore also provide new algorithms allowing agents to locally estimate the global empirical distribution, and to improve this estimate via inter-agent communication. We prove theoretically that exchanging policy information helps networked agents outperform both independent and even centralised agents in function-approximation settings. Our experiments demonstrate this happening empirically, and show that the communication network allows decentralised agents to estimate the mean field for population-dependent policies.[1]

KEYWORDS

Mean-Field Games; Communication Networks; RL

ACM Reference Format:
Patrick Benjamin and Alessandro Abate. 2026. Networked Communication for Mean-Field Games with Function Approximation and Empirical Mean-Field Estimation. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 20 pages. https://doi.org/10.65109/WWYK6345

1 INTRODUCTION

The mean-field game (MFG) framework [17, 19] can be used to circumvent the difficulty faced by multi-agent reinforcement learning regarding computational scalability as the number of agents grows [45, 48]. It models a representative agent as interacting not with other individual agents in the population on a per-agent basis, but instead with a distribution over the other agents, called the *mean field*. The MFG framework analyses the limiting case when the population consists of an infinite number of symmetric and anonymous agents, that is, they have identical reward and transition functions which depend on the mean-field distribution rather than on the actions of specific other players. The solution to this game is the mean-field Nash equilibrium (MFNE), which can be used as an approximation for the Nash equilibrium (NE) in a finite-agent game (which is harder to solve in itself), with the error in the solution reducing as the number of agents N tends to infinity [3, 8, 16, 32, 38, 44]. MFGs have thus been applied to a wide range of real-world problems: see Laurière et al. [21] for examples.

Recent works argue that classical algorithms for solving MFGs rely on assumptions and methods that are likely to be undesirable in real-world applications (e.g. swarm robotics, autonomous vehicles), emphasising that desirable qualities for practical MFG algorithms include: learning from the empirical distribution of N agents (i.e. this distribution is generated only by the policies of the agents, rather than being updated by the algorithm itself or an external oracle/simulator); learning online from a single, non-episodic system run (i.e. similar to above, the population cannot be arbitrarily reset by an external controller); model-free learning; decentralised learning; and fast practical convergence [5, 43]. While these works address these desiderata, they do 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 their approaches' scalability.

Moreover, in those works, 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 ('stationary MFGs') [3, 5, 21, 41, 43, 47]. However, in reality there are numerous reasons why agents may benefit from being able to respond to the current distribution (discussed further in Appx. B). Recent work has thus increasingly focused on these more general settings where it is necessary for agents to have so-called 'master policies' (a.k.a. population-dependent policies) which depend on both the mean-field distribution and their local state [6, 7, 21, 22, 28, 40].

The distribution is a large, high-dimensional observation object, taking a continuum of values. Therefore a population-dependent Q-function cannot be represented in a table and must be approximated. To address these limitations while maintaining the desiderata for real-world applications given in recent works, 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. To overcome the difficulties of training non-linear approximators in this context, we use the so-called 'Munchausen' trick, introduced first

[1] For our appendices, please see https://arxiv.org/abs/2408.11607.

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/WWYK6345

for single-agent RL [39], extended to MFGs [22], and then to MFGs with population-dependent policies [40].

We particularly explore this in the context of networked communication between decentralised agents. Almost all prior work relies on a centralised node to learn on behalf of all the agents. In this context ‘centralised’ does not 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. Therefore to reduce confusion, we sometimes refer to ‘central-agent learning’ instead of ‘centralised learning’ in contrast to prior works. More recent works have recognised that the assumption of a central learner might be be unrealistic in the real world, as well as a computational bottleneck and a vulnerable single point of failure [5, 43].

We demonstrate that networked communication brings two specific benefits over the purely independent setting, 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. Allowing networked agents to propagate better performing policies through the population leads to faster learning than in the purely independent case and very often even than in the central-agent case, as we show both theoretically and empirically (this method is reminiscent of the use of fitness functions in distributed evolutionary algorithms [12, 15], and similarly of ‘population-based training’ [18]). 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.

We focus on ‘coordination’ games, where agents can increase their individual rewards by following the same strategy as others and therefore have an incentive to communicate policies, even if the MFG setting itself is non-cooperative. Thus our work can be applied to problems in e.g. traffic signal control, robotic swarm formation control, vehicle platooning, and consensus and synchronisation in sensor networks [33].[2] In summary, our contributions 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 (cf. finite-horizon [22, 40]).
  - This constitutes the first use of function approximation for solving MFGs from a single, non-episodic run of the empirical system (cf. tabular settings [5, 43]).
- 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.
- Instead of 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 than even 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 architectures.

The paper is structured as follows. Related work is given in Sec. 2. We give preliminaries in Sec. 3 and our core learning and policy-improvement algorithm in Sec. 4. We present our mean-field estimation and communication algorithms in Sec. 5, theoretical results in Sec. 6 and experiments in Sec. 7.

2 RELATED WORK

We refer the reader to Benjamin and Abate [5] for detailed discussion of networked communication in MFGs, and to Laurière et al. [21] for a broader survey of MFGs. Our work is most closely related to Benjamin and Abate [5], which introduced networked communication to the infinite-horizon MFG setting. Their work focuses only on tabular settings rather than using function approximation as in ours, and only addresses population-independent policies.

Laurière et al. [22] uses Munchausen Online Mirror Descent (MOMD), similar to our method for learning with neural networks, but in a different setting: their 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, in a finite-horizon, episodic setting. Wu et al. [40] 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. [28] 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. [49] 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. [46] 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 the closely related but distinct area of mean-field RL, Subramanian et al. [36] does estimate the empirical mean-field distribution from

[2] We further pre-empt concerns about communication in competitive settings by wondering whether self-interested agents would be any more likely to want to obey a central learner as has usually been assumed. Moreover we show that self-interested communicating agents can obtain higher returns than independent agents even in non-coordination games (Fig. 6), indicating that they do have incentive to communicate.

the local neighbourhood, however agents are seeking to estimate the mean action rather than the mean-field 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.

3 PRELIMINARIES

3.1 Mean-field games

We use the following notation. N is the number of agents in a population, with $\mathcal{S}$ and $\mathcal{A}$ representing the finite state and common action spaces, respectively. The 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. For agent $i \in 1\ldots N$, 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, 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 $\mathcal{Q} = \{q : \mathcal{S}\times\Delta_{\mathcal{S}}\times\mathcal{A} \to \mathbb{R}\}$.

Definition 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.:

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

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 so-called 'mean-field flow' $\boldsymbol{\mu}$ is given by the infinite sequence of mean-field distributions s.t. $\boldsymbol{\mu} = (\mu_t)_{t\geq0}$.

Definition 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. We also give the following definitions.

Definition 3.3 (Mean-field discounted return)**.** In a MFG where all agents follow policy π giving a mean-field flow $\boldsymbol{\mu} = (\mu_t)_{t\geq0}$, the expected discounted return of the representative agent is given by

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

Definition 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 $BR(\boldsymbol{\mu})$:

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

Definition 3.5 (Mean-field Nash equilibrium (MFNE))**.** A pair $(\pi^*,\boldsymbol{\mu}^*)$ is a mean-field Nash equilibrium (MFNE) 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 MFNE policy for any initial distribution μ_0, it is a 'master policy'.

Previous works have shown 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 MFNE using only the empirical distribution $\hat{\mu}_t$, rather than the exactly calculated infinite flow $\boldsymbol{\mu}$ [5, 43]. This MFNE 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. In this work we provide algorithms to perform this process in non-tabular and population-dependent settings, and demonstrate them empirically.

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. $\pi_{k+1}(\cdot|s,\mu) = softmax\left(\frac{1}{\tau_q}\sum_{\kappa=0}^{k} Q_\kappa(s,\mu,\cdot)\right)$. τ_q is a temperature parameter that scales the entropy in Munchausen RL [39]; note that this is a different temperature to the one agents use when selecting which communicated parameters to adopt, denoted τ_k^{comm} (Sec. 4.2).

If the Q-function is approximated non-linearly using neural networks, it is difficult to compute this weighted sum. The so-called '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} [39]. Using this reparametrisation gives Munchausen OMD (MOMD), detailed further in Sec. 4.1 [22, 40]. MOMD does not bias the MFNE, and has the same convergence guarantees as OMD [14, 26, 40].

3.3 Networks

We conceive of the finite population as exhibiting two time-varying networks. The basic definition of such a network is:

Definition 3.6 (Time-varying network)**.** The 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 representing agents $i \in \{1, \ldots, N\}$, 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 t. The *diameter* $d_{\mathcal{G}_t}$ is the maximum of the shortest path length between any pair of nodes.

One of these networks $\mathcal{G}_t^{comm}$ defines which agents can communicate information to each other at time t. The second network $\mathcal{G}_t^{obs}$ is a graph defining which agents can observe each other's states, which we use in general settings for estimating the mean-field distribution 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. This visibility graph is:

Definition 3.7 (Time-varying state-visibility graph)**.** The 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 say an agent in s can obtain a count of the number of agents in s' if s' is visible to s. The benefit of $\mathcal{G}_t^{vis}$ 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 $\mathcal{G}_t^{vis}$ applies (see Sec. 5).

Algorithm 1 Networked learning with function approximation

Require: loop parameters K, M, L, E, C_p; learning parameters $\gamma, \tau_q, |B|, cl, \nu, \{\tau_k^{comm}\}_{k\in\{0,\ldots,K-1\}}$; initial states $\{s_0^i\}_{i=1}^N$; $t \leftarrow 0$

1: $\forall i$: Randomly initialise parameters θ_0^i of Q-networks $\check{Q}_{\theta_0^i}(o,\cdot)$; set $\pi_0^i(a|o) = \text{softmax}\left(\frac{1}{\tau_q}\check{Q}_{\theta_0^i}(o,\cdot)\right)(a)$ and $\check{Q}_{\theta_{0,0}^{i,\prime}} \leftarrow \check{Q}_{\theta_0^i}(o,\cdot)$
2: **for** $k = 0, \ldots, K-1$ **do**
3: $\quad\forall i$: Empty i's buffer
4: $\quad$**for** $m = 0, \ldots, M-1$ **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** $l = 0, \ldots, L-1$ **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. 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: $\quad\forall i : \pi_{k+1}^i(a|o) \leftarrow \text{softmax}\left(\frac{1}{\tau_q}\check{Q}_{\theta_{k+1}^i}(o,\cdot)\right)(a)$
15: $\quad\forall i : \sigma_{k+1}^i \leftarrow 0$
16: $\quad$**for** $e = 0, \ldots, E-1$ 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 $\text{adopted}^i \sim \Pr(\text{adopted}^i = j) = \frac{\exp(\sigma_{k+1}^j/\tau_k^{comm})}{\sum_{x\in J_t^i}\exp(\sigma_{k+1}^x/\tau_k^{comm})}\ \forall j \in J_t^i$
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 $\{\pi_K^i\}_{i=1}^N$

4 LEARNING AND POLICY IMPROVEMENT

4.1 Q-network and update

Lines 1-14 of our novel Alg. 1 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 $\pi_{\theta_k^i}(a|o) = \text{softmax}\left(\frac{1}{\tau_q}\check{Q}_{\theta_k^i}(o,\cdot)\right)(a)$. 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 $\left(o_t^i, a_t^i, r_t^i, o_{t+1}^i\right)$. 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 $\check{Q}_{\theta_{k,l}^{i,\prime}}$ 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 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 empirical loss (Def. 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 [39, 40]:

Definition 4.1 (Empirical loss for Q-network)**.** The loss is:

$$\hat{\mathcal{L}}(\theta,\theta') = \frac{1}{|B|} \sum_{transition \in B^i_{k,l}} \left| \check{Q}_{\theta^i_{k,l}}(o_t, a_t) - T \right|^2, \quad \text{for the target } T:$$

$$T = r_t + [\tau_q \ln \pi_{\theta^{i,\prime}_{k,l}}(a_t|o_t)]^0_{cl} + \gamma \sum_{a\in\mathcal{A}} \pi_{\theta^{i,\prime}_{k,l}}(a|o_{t+1}) \left(\check{Q}_{\theta^{i,\prime}_{k,l}}(o_{t+1}, a) - \tau_q \ln \pi_{\theta^{i,\prime}_{k,l}}(a|o_{t+1}) \right).$$

4.2 Communication and adoption of parameters

We use the communication network $\mathcal{G}^{comm}_t$ to share two types of information at different points in Alg 1. One is used to improve local estimates of the mean field (Sec. 5). The other, described here, 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.

We adapt Benjamin and Abate [5] for the function-approximation case, where in our work agents broadcast the parameters of the Q-network that defines their policy, rather than the Q-function table. At each iteration k, after independently updating their Q-network and policy (Lines 3-14), agents *estimate* the infinite discounted return (Def. 3.3) of their new policies by collecting rewards for E steps, 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. 6.

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. 3.3). 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. 1, 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. Both versions of the algorithm are subject to implicit assumptions, which we discuss methods for addressing in Appx. C.

5.1 Algorithm for the general setting

In this setting, our method (Alg. 2) assumes each agent is associated with a unique ID to avoid the same agents being counted multiple times. Each agent maintains a 'count' vector $\hat{v}^i_t$ 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 $\hat{v}^i_t$ is initialised as full of $\emptyset$ ('no count') markers

Algorithm 2 Mean-field estimation in general settings

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

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

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 $\hat{v}^i_t[s]$ 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 $\tilde{\hat{\mu}}^i_t$ is then obtained from $\hat{v}^i_t$ 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 $\hat{v}^i_t$ 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 $\tilde{\hat{\mu}}^i_t$, replacing the $\emptyset$ marker for states for which no agents have been observed (Line 10).

5.2 Algorithm for visibility-based environments

We explain now the differences in our estimation algorithm (Alg. 3) for the subclass of environments where $\mathcal{G}^{vis}_t$ applies in place of $\mathcal{G}^{obs}_t$, i.e. the mutual observability of agents depends in turn on the mutual visibility of states. The benefit of $\mathcal{G}^{vis}_t$ over $\mathcal{G}^{obs}_t$ 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

Algorithm 3 Mean-field estimation 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 $\{\tilde{\hat{\mu}}_t^i\}_{i=1}^N$

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 setting.[3]

6 THEORETICAL RESULTS

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 previous MFG works [5, 43], in the central-agent setting, the Q-network updates of arbitrary agent $i = 1$ are automatically pushed to all other agents, and 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 often learn faster than central-agent ones in our experiments. To indicate how this is possible while allowing simplicity of the theory, we give a proof for a special case with relatively strong assumptions that give conditions under which networked populations *do* outperform central-agent ones. Nevertheless the intuition provided by Thm. 6.3's proof suggests why networked agents can learn faster even without enforcing the assumptions, and we discuss loosening them subsequently.

Recall that at each iteration k of Alg. 1, after independently updating their policies in Line 14, the population has the policies $\{\pi_{k+1}^i\}_{i=1}^N$. 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 $\{V(\pi_{k+1}^i, I(\pi_{k+1}^i))\}_{i=1}^N$ (Def. 3.3) 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 $\{\pi_{k+1}^i\}_{i=1}^N$ (i.e. they do not at this stage all follow a single identical policy). We can characterise the approximation as $\{\sigma_{k+1}^i\}_{i=1}^N = \{\widehat{V}(\pi_{k+1}^i, I(\pi_{k+1}^i))\}_{i=1}^N$. We now assume the following:

Assumption 6.1. After the C_p rounds in Lines 21-27 in which agents exchange and adopt policies from neighbours, the population is left with a single policy such that $\forall i, j \in \{1, \dots, N\}$ $\pi_{k+1}^i = \pi_{k+1}^j$.[4]

Assumption 6.2. 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. $\forall i, j \in \{1, \dots, N\}$:

$$\sigma_{k+1}^i > \sigma_{k+1}^j \iff V(\pi_{k+1}^i, I(\pi_{k+1}^i)) > V(\pi_{k+1}^j, I(\pi_{k+1}^j)).$$

Call the network consensus policy π_{k+1}^{net}, and its associated finitely estimated return $\sigma_{k+1}^{\text{net}}$. 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 $\{\pi_{k+1}^i\}_{i=1}^N$. Call this policy *arbitrarily* given to the whole population π_{k+1}^{cent}, and its associated finitely estimated return $\sigma_{k+1}^{\text{cent}}$. Now we can say:

Theorem 6.3. *Given Assumptions 6.1 and 6.2,*

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

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

Proof. Recall that before the communication rounds in Line 21 (Alg. 1), the randomly updated policies $\{\pi_{k+1}^i\}_{i=1}^N$ have associated estimated returns $\{\sigma_{k+1}^i\}_{i=1}^N$. Call the mean and maximum of this set $\sigma_{k+1}^{\text{mean}}$ and $\sigma_{k+1}^{\text{max}}$ respectively. Since π_{k+1}^{cent} is chosen arbitrarily from $\{\pi_{k+1}^i\}_{i=1}^N$, it will obey $\mathbb{E}[\sigma_{k+1}^{\text{cent}}] = \sigma_{k+1}^{\text{mean}}$ $\forall k$, though there will be high variance. Conversely, the softmax adoption probability (Line 24, Alg. 1) for the networked case means by definition that policies with higher σ_{k+1}^i are more likely to be adopted at each communication round. Thus the π_{k+1}^{net} that is adopted by the whole networked population will obey $\mathbb{E}[\sigma_{k+1}^{\text{net}}] > \sigma_{k+1}^{\text{mean}}$ (if τ_{k+1}^{comm} is a positive value

[3]In our Algs. 2 and 3, 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 is only then 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 Appx. C 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 rounds.

[4]Most simply Assumption 6.1 holds if 1) τ_k^{comm} is a positive constant sufficiently close to zero 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 larger than 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 σ_{k+1}^{max} value (and hence the unique associated π_{k+1}^{max}) within a number of rounds equal to the diameter $d_{\mathcal{G}_t^{comm}}$ [5, 25]. However, policy consensus can be achieved even outside of these conditions, including if the network is dynamic and not connected at every step, given appropriate values for C_p and $\tau_{k+1}^{comm} \in \mathbb{R}_{>0}$.

near zero, it will obey $\mathbb{E}[\sigma^{\text{net}}_{k+1}] = \sigma^{\max}_{k+1}\ \forall k$). So $\mathbb{E}[\sigma^{\text{net}}_{k+1}] > \mathbb{E}[\sigma^{\text{cent}}_{k+1}]$, which by Assumption 6.2 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 6.2), 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.[5,6] Similar logic can be applied to understand why networked agents outperform independent ones, coupled 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 [5, 43].

Significantly, our communication scheme not only avoids the undesirable assumption of a central node, but even outperforms it. Moreover, the benefit of the scheme over central-agent learning is greater with our function approximation than in the tabular case (cf. [5]), perhaps due to greater variance in the quality of Q-function estimates in our case. This shows that networked communication facilitates greater scalability than the central-agent paradigm.

7 EXPERIMENTS

We provide two sets of experiments. The first shows that our function-approximation algorithm (Alg. 1) can scale to large state spaces for population-independent policies, and that in such settings networked, communicating populations can outperform purely-independent and even central-agent populations. The second set demonstrates that Alg. 1 can handle population-dependent policies, as well as the ability of Alg. 3 to estimate the mean field locally.

7.1 Experimental set-up

For the types of game 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 [2, 5, 9, 20, 22, 40, 47]. We present results from four tasks defined by the agents' reward/transition functions, all of which are *coordination* tasks - see Appx. A.1 for full technical descriptions, and also a fifth, non-coordination task. In all cases, rewards are normalised in [0,1] after they are computed. The first two tasks are those used with population-independent policies in Benjamin and Abate [5], but while they show results for an 8×8 and a 'larger' 16×16 grid, our results are for 100×100 and 50×50 grids:

- **Cluster.** Agents are rewarded for gathering but given no indication where to do so, agreeing it over time.
- **Target agreement.** Agents are rewarded for visiting any of a given number of targets, but the reward is proportional to the number of other agents co-located at the target. Agents must coordinate on which single target they will all meet at to maximise their individual rewards.

We also show our algorithms handling more complex tasks:

- **Evade shark in shoal.** At each t, a 'shark' takes a step towards the grid point containing the most agents according to the empirical mean field. The shark's position is part of agents' local states in addition to their own position. Agents are rewarded more for being further from the shark and for clustering with other agents. As well as having a non-stationary distribution, we add 'common noise', with the shark taking a random step with probability 0.01. 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 [21].
- **Push object to edge.** Agents are rewarded for how close an 'object' is to the grid's edge. The object's position forms part of agents' local states in addition to their own position. The object moves in a direction with a probability proportional to the number of agents on its opposite side, i.e. agents must coordinate on which side of the object from which to 'push' it, to ensure it moves toward the edge of the grid.

We evaluate our experiments via two metrics. *Exploitability* is the most common metric in MFG works, and is a measure of proximity to the MFNE. It quantifies how much an agent can benefit by deviating from the set of policies that generate the current mean field, with a lower exploitability meaning the population is closer to the MFNE. However, there are several issues with this metric in our setting, particularly for our coordination tasks where competitive agents benefit from aligning behaviours, such that it may give limited or noisy information (discussed further in Appx. A.2.1). We thus also give a second metric, as in Benjamin and Abate [5]: the population's *average discounted return*. 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 MFNEs to which populations converge.

7.2 Results and discussion

In our spatial environments, the physical distance from i determines the communication graph $\mathcal{G}^{comm}_t$ and the visibility graph $\mathcal{G}^{vis}_t$. Our plots show various radii, given 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. Other hyperparameters are detailed in Appx. A.3.

7.2.1 Population-independent policies in large state-spaces. Figs. 1 and 2 (for 100 × 100 and grids; reproduced larger in Figs. 3 and 4, Appx. A), and Figs. 5 and 7 (for 50×50 grids; Appx. A) 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. By contrast, the tabular algorithms in Benjamin and Abate [5] appear only just to converge

[5]Assumption 6.1 may not hold if C_p is not large enough or if parts of the population remain isolated. Thus in our experiments, where we use $C_p = 1$ to show the benefit of even one communication round, networked populations with smaller broadcast radii outperform central-agent populations by a smaller margin. Nevertheless the intuition provided by Thm. 6.3's proof indicates how the former are still able to perform better even **if Assumption 6.1 is loosened**.

[6]Even **if Assumption 6.2 does not strictly hold**, the softmax parameter τ^{comm}_k allows a smooth degradation as the ordering of the approximations worsens with respect to the ordering of the true values. I.e. 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 the best policy may still be adopted even if it is not evaluated as the best.

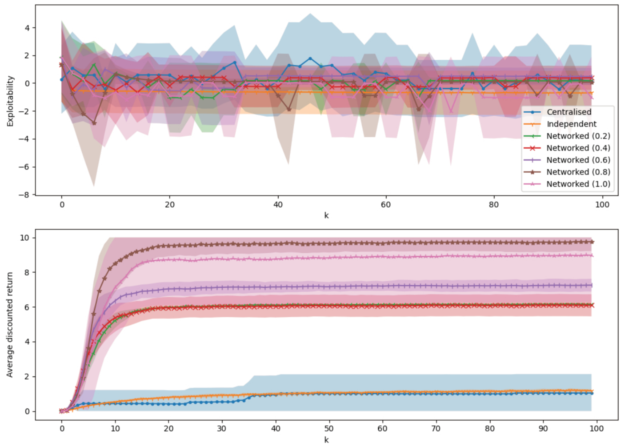

Figure 1: 'Target agreement', pop.-independent, 100×100 **grid. All networked populations significantly outperform central-agent and independent populations wrt. average return, where the latter two cases hardly appear to learn at all.**

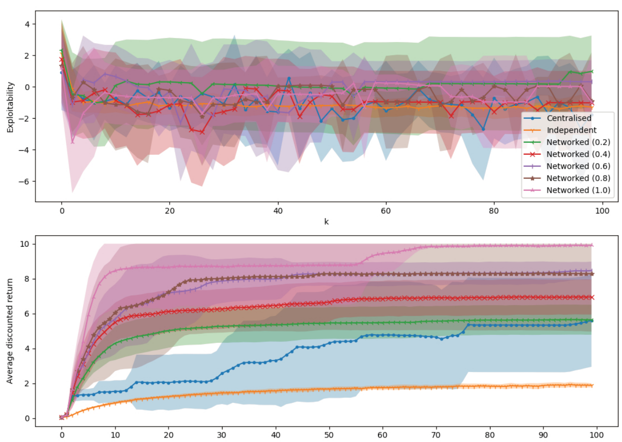

Figure 2: 'Cluster', pop.-independent, 100×100 **grid. All networked populations outperform central-agent and independent populations wrt. return; the latter hardly learn at all.**

by $k = 200$ for the same tasks on the larger of their two grids, which is only 16×16.

In Figs. 1, 2 and 5, no populations appear to have significantly different exploitability to each other, while in Fig. 7 the central-agent population may have lower exploitability, but not significantly so. As discussed in Appx. A.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. 1 and 5. There is therefore little a 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 improve their returns and thus significantly outperform the stagnant independent agents in Figs. 1, 2, 5 and 7 and the stagnant central-agent populations in Figs. 1 and 5. This indicates that our communication scheme helps agents to reach 'preferable' equilibria, even when exploitability is similar (for reasons related to how networked populations can increase their returns faster as per Sec. 6). While central-agent populations do appear to increase their returns in the 'cluster' task in Figs. 2 and 7, they do so more slowly and reaching a lower final value than all networked agents in the 100×100 grid case, and than all networked agents apart from those with the smallest broadcast radii in the 50×50 grid case. Indeed in the 100×100 grids in Figs. 1 and 2, the central-agent populations appear to perform less well than they do in the 50×50 grids in Figs. 5 and 7, 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 Benjamin and Abate [5], the central-agent populations generally perform similarly to the networked populations, indicating that our networked architecture is more robust than the central-agent alternative when moving to non-tabular settings.

7.2.2 Population-dependent policies in complex environments. We also show the ability of our algorithms to handle more complex tasks, using population-dependent policies and estimated mean-field observations. Figs. 8a and 9a (Appx. A), where agents estimate the mean field via Alg. 3, differ minimally from Figs. 8b and 9b, where agents directly receive the global mean field. 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. 8a, and there only with the smaller broadcast radii, do agents slightly underperform the returns of agents in the global observability case in Fig. 8b, as might be expected.

For the reasons given in Appx. A.2.1 regarding coordination games, the exploitability metric gives limited information in the 'push object' and 'evade' tasks in Figs. 8 and 9: 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.

8 CONCLUSION

We introduced function approximation to the online-learning setting for empirical MFGs, and also contributed novel algorithms for locally estimating the empirical mean field for population-dependent policies. We proved theoretically that our networked communication algorithms can learn faster than even central-agent architectures in this function approximation setting, and showed empirically that they can handle large state spaces and estimate the mean field. See Appx. C for future work.

REFERENCES

[1] Swadhin Agrawal, Jitesh Jhawar, Andreagiovanni Reina, Sujit P Baliyarasimhuni, Heiko Hamann, and Liang Li. 2024. Impact of Individual Defection on Collective Motion. In *International Conference on Swarm Intelligence*. Springer, 127–140.

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

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

[4] Dirk Becherer and Stefanie Hesse. 2024. Common Noise by Random Measures: Mean-Field Equilibria for Competitive Investment and Hedging. *arXiv preprint arXiv:2408.01175* (2024).

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

[6] Pierre Cardaliaguet, François Delarue, Jean-Michel Lasry, and Pierre-Louis Lions. 2015. The master equation and the convergence problem in mean field games. arXiv:1509.02505 [math.AP] https://arxiv.org/abs/1509.02505

[7] René Carmona, François Delarue, and Daniel Lacker. 2016. Mean Field Games with Common Noise. *The Annals of Probability* 44, 6 (2016), 3740–3803. http://www.jstor.org/stable/44072057

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

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

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

[11] Breno Cunha Queiroz and Daniel MacRae. 2024. Occlusion-based object transportation around obstacles with a swarm of miniature robots. *Swarm Intelligence* (2024), 1–29.

[12] Agoston Eiben and James Smith. 2015. *What Is an Evolutionary Algorithm?* Springer Berlin Heidelberg, Berlin, Heidelberg, 25–48. https://doi.org/10.1007/978-3-662-44874-8_3

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

[14] Saeed Hadikhanloo. 2017. Learning in anonymous nonatomic games with applications to first-order mean field games. *arXiv preprint arXiv:1704.00378* (2017).

[15] Emma Hart, Andreas Steyven, and Ben Paechter. 2015. 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* (Madrid, Spain) *(GECCO '15)*. Association for Computing Machinery, New York, NY, USA, 169–176. https://doi.org/10.1145/2739480.2754688

[16] Anran Hu and Junzi Zhang. 2024. MF-OML: Online Mean-Field Reinforcement Learning with Occupation Measures for Large Population Games. *arXiv preprint arXiv:2405.00282* (2024). https://arxiv.org/abs/2405.00282

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

[18] Max Jaderberg, Valentin Dalibard, Simon Osindero, Wojciech M Czarnecki, Jeff Donahue, Ali Razavi, Oriol Vinyals, Tim Green, Iain Dunning, Karen Simonyan, et al. 2017. Population based training of neural networks. *arXiv preprint arXiv:1711.09846* (2017).

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

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

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

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

[23] Min Li, Tianyang Nie, Shujun Wang, and Ke Yan. 2024. Incomplete Information Mean-Field Games and Related Riccati Equations. *Journal of Optimization Theory and Applications* (2024), 1–22.

[24] Jincun Liu, Yinjie Ren, Yang Liu, Yan Meng, Dong An, and Yaoguang Wei. 2025. Achievement of Fish School Milling Motion Based on Distributed Multi-agent Reinforcement Learning. *Journal of Bionic Engineering* (2025), 1–19.

[25] Behrang Monajemi Nejad, Sid Ahmed Attia, and Jorg Raisch. 2009. 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*. 1–7. https://doi.org/10.1109/ICAT.2009.5348437

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

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

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

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

[30] Sarah Perrin, Julien Pérolat, Mathieu Laurière, Matthieu Geist, Romuald Elie, and Olivier Pietquin. 2020. Fictitious Play for Mean Field Games: Continuous Time Analysis and Applications. In *Proceedings of the 34th International Conference on Neural Information Processing Systems* (Vancouver, BC, Canada) *(NIPS'20)*. Curran Associates Inc., Red Hook, NY, USA, Article 1107, 15 pages.

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

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

[33] Javad Soleimani, Reza Farhangi, and Gunes Karabulut Kurt. 2024. Distributed Critic-Based Neuro-Fuzzy Learning in Swarm Autonomous Vehicles. In *2024 IEEE 100th Vehicular Technology Conference (VTC2024-Fall)*. 1–6. https://doi.org/10.1109/VTC2024-Fall63153.2024.10757965

[34] Jayakumar Subramanian and Aditya Mahajan. 2019. Reinforcement Learning in Stationary Mean-Field Games. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems* (Montreal QC, Canada) *(AAMAS '19)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 251–259.

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

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

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

[38] Noureddine Toumi, Roland Malhame, and Jerome Le Ny. 2024. A mean field game approach for a class of linear quadratic discrete choice problems with congestion avoidance. *Automatica* 160 (2024), 111420. https://doi.org/10.1016/j.automatica.2023.111420

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

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

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

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

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

[44] Batuhan Yardim, Artur Goldman, and Niao He. 2024. When is Mean-Field Reinforcement Learning Tractable and Relevant? *arXiv preprint arXiv:2402.05757* (2024).
[45] Batuhan Yardim and Niao He. 2024. Exploiting Approximate Symmetry for Efficient Multi-Agent Reinforcement Learning. *arXiv preprint arXiv:2408.15173* (2024).
[46] Bora Yongacoglu, Gürdal Arslan, and Serdar Yüksel. 2024. Mean-field games with finitely many players: independent learning and subjectivity. *Journal of Machine Learning Research* 25, 419 (2024), 1–69.
[47] Muhammad Aneeq Uz Zaman, Alec Koppel, Sujay Bhatt, and Tamer Basar. 2023. Oracle-free Reinforcement Learning in Mean-Field Games along a Single Sample Path. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 10178–10206.
[48] Sihan Zeng, Sujay Bhatt, Alec Koppel, and Sumitra Ganesh. 2024. A Single-Loop Finite-Time Convergent Policy Optimization Algorithm for Mean Field Games (and Average-Reward Markov Decision Processes). *arXiv e-prints* (2024), arXiv–2408.
[49] Chenyu Zhang, Xu Chen, and Xuan Di. 2024. Stochastic Semi-Gradient Descent for Learning Mean Field Games with Population-Aware Function Approximation. *arXiv preprint arXiv:2408.08192* (2024).