Model-Free Mean-Field Reinforcement Learning: Mean-Field MDP and Mean-Field Q-Learning

René Carmona[†] **Mathieu Laurière**[†] **Zongjun Tan**[†]

Abstract

We develop a general reinforcement learning framework for mean field control (MFC) problems. Such problems arise for instance as the limit of collaborative multi-agent control problems when the number of agents is very large. The asymptotic problem can be phrased as the optimal control of a non-linear dynamics. This can also be viewed as a Markov decision process (MDP) but the key difference with the usual RL setup is that the dynamics and the reward now depend on the state's probability distribution itself. Alternatively, it can be recast as a MDP on the Wasserstein space of measures. In this work, we introduce generic model-free algorithms based on the state-action value function at the mean field level and we prove convergence for a prototypical Q-learning method. We then implement an actor-critic method and report numerical results on two archetypal problems: a finite space model motivated by a cyber security application and a continuous space model motivated by an application to swarm motion.

1 Introduction

Typical *reinforcement learning* (RL) applications involve the search for a procedure to learn by trial and error the optimal behavior so as to maximize a reward. While similar in spirit to optimal control applications, a key difference is that in the latter, the model is assumed to be known to the controller. This is in contrast with RL for which the environment has to be explored, and the reward cannot be predicted with certainty. Still, the RL paradigm has generated numerous theoretical developments and found plenty practical applications. As a matter of fact, bidirectional links with the optimal control literature have been unveiled as common tools lie at the heart of many studies. *Mean field control* (MFC), also called *optimal control of McKean-Vlasov* (MKV) dynamics, is an extension of stochastic control which has recently attracted a surge of interest (see e.g. [5, 11, 12]). From a theoretical standpoint, the main peculiarity of this type of problems is that the transition and reward functions not only involve the state and the action of the controller, but also the distribution of the state (and possibly of the control). Practically speaking, these problems appear as the asymptotic limits for the control of a large number of collaborative agents. They can also be introduced as single agent problems whose evolution and costs depend upon the distribution of her state (and potentially of her control). Such problems have found a wide range of applications in distributed robotics, energy, drone fleet management, risk management, finance, etc. Although they are *bona fide* control problems for which a dynamic programming principle can be formulated, they generally lead to Bellman equations on the infinite dimensional space of measures, which are extremely difficult to solve ([6, 32, 35]). Figure 1 contains a schematic diagram of the relationships between optimal control (i.e., planning with a model) and how the paradigms of MFC and RL are combined in mean-field reinforcement learning (MFRL).

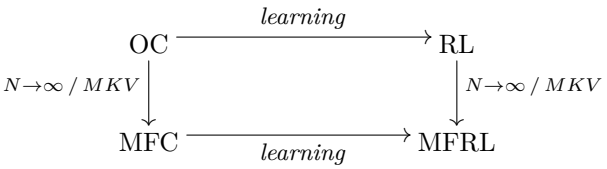


Figure 1: Relationships between optimal control (OC), mean-field control (MFC), reinforcement learning (RL) and mean-field reinforcement learning (MFRL). Horizontal arrows: extension in the direction of model-free learning. Vertical arrows: generalization by letting the number of agents grow to infinity or by controlling a MKV dynamics.

[†]Department of Operations Research and Financial Engineering, Princeton University

Main contributions. Our first contribution is conceptual and consists in introducing a general framework of MFC problems in discrete time, with infinite horizon, discount and common noise. We argue that this setup, which has not been covered in the classical literature on MFC problems, is particularly relevant to develop a theory of reinforcement learning for mean field problems. We then rephrase the problem as a Markov decision process (MDP) on the space of measures. This point of view leads to the introduction of a state value function and a state-action value function as well as their associated Bellman equations on the Wasserstein space of measures. Our second contribution is to propose two model-free methods to learn an approximation of the state-action value function by trial and error. The first method relies on a discretization of the simplex and a tabular version of Q-learning, for which we prove a convergence result. The second method is based on an actor-critic method (Deep Deterministic Policy Gradient). Last, our third contribution is to implement the latter method and assess its convergence numerically. Numerical tests are conducted on two prototypical examples drawn from the mean-field literature: a finite state model motivated by a cyber security application and a continuous state and action model motivated by an application to swarm motion.

2 Mean Field Control

In this section, we keep the discussion at an informal level in order to encompass both finite and continuous state spaces. Specific methods and examples for each setting are presented in Sections 3 and 4.

Definition of the problem. We denote by $\mathcal{S}$ and $\mathcal{A}$ respectively the state space and the action space. Typically, we have in mind the finite space case where both are finite sets, say $\mathcal{S} = \{1, \dots, |\mathcal{S}|\}$ and $\mathcal{A} = \{1, \dots, |\mathcal{A}|\}$, or the continuous case, where $\mathcal{S} = \mathbb{R}^d$ and $\mathcal{A} = \mathbb{R}^k$. A generic discrete time, infinite horizon, discounted mean field control (MFC) problem (or control of McKean-Vlasov dynamics) takes the following form: *Maximize over the control process (or policy) π the reward functional*

$$J(\mu_0, \pi) = \mathbb{E}\left[\sum_{t=0}^{+\infty} \gamma^t f(x_t^{\pi,\mu_0}, \mu_t^{\pi,\mu_0}, \pi_t)\right] \qquad (1)$$

where the state process has initial distribution μ_0 and dynamics

$$\mathbb{P}\left(x_{t+1}^{\pi,\mu_0} \in \cdot \,|\, x_t^{\pi,\mu_0}, \mu_t^{\pi,\mu_0}, \pi_t, \epsilon_t^0\right) = p_{x_t^{\pi,\mu_0}, \mu_t^{\pi,\mu_0}, \pi_t, \epsilon_t^0}(\cdot). \qquad (2)$$

Here $p : \mathcal{S} \times \mathcal{P}(\mathcal{S}) \times \mathcal{A} \times \mathcal{S} \times \mathcal{E}^0 \to \mathbb{R}$ encodes the transition. The mean-field nature of the model stems from the presence of $\mu_t^{\pi,\mu_0} = \mathcal{L}(x_t^{\pi,\mu_0}|\epsilon^0) \in \mathcal{P}(\mathcal{S})$, which is the law of x_t^{π,μ_0} conditioned on the realization of ϵ^0 up to time $t-1$, where $(\epsilon_t^0)_{t\geq 0}$ is a stochastic process taking values in a set $\mathcal{E}^0$. For simplicity we assume that the ϵ_t^0 are i.i.d. They play the role of a so-called *common* noise affecting the state transitions. Although the presence of this noise is not necessary for the model to be meaningful and we postpone the rigorous mathematical framework to future work, we believe that its presence is important for applications. Motivations and examples for this type of noise are provided in the sequel. Randomness in the rewards as well as interactions through the control's distribution could also be handled, but for the sake of simplicity of the presentation we limit ourselves to a reward f which is a deterministic function of $(x, \mu, a) \in \mathcal{S} \times \mathcal{P}(\mathcal{S}) \times \mathcal{A}$ and to the interaction through the conditional distribution of the state only.

Remark 1. *Note that J depends on μ_0. Although this dependence is usually omitted in the MFC literature, it is important to remember that the solution of the MFC problem changes if we let this initial distribution vary.*

In the finite case, the above dynamics take the following form, where we identify $\mathcal{P}(\mathcal{S})$ with the $|\mathcal{S}|$-simplex denoted by $\mathbb{S}$: for every $(x, \mu, a, s, e^0) \in \mathcal{S} \times \mathbb{S} \times \mathcal{A} \times \mathcal{S} \times \mathcal{E}^0$, $p_{x,\mu,a,e^0}(x')$ corresponds to

$$\mathbb{P}\left(x_{t+1}^{\pi,\mu_0} = x' \,|\, (x_t^{\pi,\mu_0}, \mu_t^{\pi,\mu_0}, \pi_t, \epsilon_t^0) = (x, \mu, a, e^0)\right).$$

Such an evolution can be interpreted in terms of a transition rate matrix, and the common noise can for instance affect the coefficients of this matrix. In the continuous case, (2) can come from a continuous time model, for example a stochastic differential equation of the McKean Vlasov type (MKV SDE) via an Euler scheme [8], in which case:

$$x_{t+1}^{\pi,\mu_0} = x_t^{\pi,\mu_0} + b(x_t^{\pi,\mu_0}, \mu_t^{\pi,\mu_0}, \pi_t) + \epsilon_{t+1} + \epsilon_{t+1}^0, \quad (3)$$

where the random variables $\epsilon_t, \epsilon_t^0, t \geq 1$ are independent (e.g. with Gaussian distributions) and are interpreted as sources of noise. This type of setting has been studied in [16] with a linear dynamics and a quadratic cost.

When there is no ambiguity from the context, we will drop the superscripts π and μ_0. Intuitively, (1) is the limiting problem, as N grows to infinity, of the following control problem for N agents: *Maximize over* $(\pi^1, \dots, \pi^N)$ *the social average reward*

$$J^N(\mu_0, \pi^1, \dots, \pi^N) = \frac{1}{N} \sum_{i=1}^{N} \mathbb{E}\left[\sum_{t=0}^{+\infty} \gamma^t f(x_t^i, \bar{\mu}_t^N, \pi_t^i) \right],$$

where $\bar{\mu}_t^N = \sum_{j=1}^N \delta_{x_t^j}/N$ *is the empirical distribution,* x_0^i *are i.i.d. with distribution* μ_0 *and the dynamics are* $\mathbb{P}\left(x_{t+1}^i \in \cdot \,|\, x_t^i, \bar{\mu}_t^N, \pi_t^i, \epsilon_t^0\right) = p_{x_t^i, \bar{\mu}_t^N, \pi_t^i, \epsilon_t^0}(\cdot)$. Note that the same ϵ_t^0 appears in the transitions of all $(x_t^i)_{i=1,\dots,N}$. In other words, the system is affected by two sources of noise: one perturbs each state x^i independently, and one affects all the agents. The first noise is *idiosyncratic* to each agent whereas the second one is *common* to the whole population. This latter type of noise is important in many applications, see e.g. [13, 2] for models of systemic risk or energy management in the context of mean field games. Since, in this N-agent problem, the goal is to maximize J^N, the problem can be interpreted e.g. as a large collaborative game (i.e. a Pareto optimum, rather than a Nash equilibrium as in mean field games), or as the problem of a central planner trying to find the best way to control a large group of robots.

Reformulation as an MDP on the Wasserstein space of measures. We reformulate the MFC problem (1) as the optimal control of a Markov decision process in which the state space is the space of measures, in the spirit of e.g. [23]. We restrict our attention to controls which are stationary feedback functions of $(\mathcal{L}(x_t), x_t)$, namely processes π for which there exists a (deterministic) function $a : \mathcal{P}(\mathcal{S}) \times \mathcal{S} \to \mathcal{A}$ such that for all t

$$\pi_t = a\Big(\mathcal{L}(x_t^{\pi,\mu_0}), x_t^{\pi,\mu_0}\Big).$$

In this situation, a typical agent takes her decision at each time step based on only two pieces of information: her current state and the distribution of the population's states. For such a and π, we will write $J(a)$ instead of $J(\pi)$. We denote by $\mathbb{A}$ the set of such functions a and it will sometimes be convenient to view them as functions from $\mathcal{P}(\mathcal{S})$ to the set $\tilde{\mathbb{A}} = \{\tilde{a} : \mathcal{S} \to \mathcal{A}\}$. The initial MFC problem (1) can be recast as an optimal control problem on the distribution flow, namely,

$$J(\mu_0, a) = \mathbb{E}_{(\epsilon_t^0)_{t\geq 0}} \sum_{t=0}^{+\infty} \gamma^t \tilde{f}\Big(\mu_t^{\mu_0,a}, a(\mu_t^{\mu_0,a})\Big) \qquad (4)$$

under the constraint: $\mu_t^{\mu_0,a} = \mathcal{L}(x_t^{\mu_0,a}|\epsilon^0)$, where $x^{\mu_0,a}$ solves (2). Here $\tilde{f} : \mathcal{P}(\mathcal{S}) \times \tilde{\mathbb{A}} \to \mathbb{R}$ is defined by

$$\tilde{f}(\mu, \tilde{a}) = \mathbb{E}_{x\sim\mu}[f(x, \mu, \tilde{a}(x))],$$

and the evolution of the distribution is given by

$$\mu_{t+1}^{\mu_0,a} = \Phi^{a(\mu_t^{\mu_0,a},\cdot),\epsilon_t^0}(\mu_t^{\mu_0,a}) \qquad (5)$$

where $\Phi : \tilde{\mathbb{A}} \times \mathcal{E}^0 \times \mathcal{P}(\mathcal{S}) \to \mathcal{P}(\mathcal{S}), (\tilde{a}, e^0, \mu) \mapsto \Phi^{\tilde{a},e^0}(\mu')$, formalizing the transition (2) in our new set of notations. Note that Φ depends (possibly in a non-linear way) on the distribution at time t, in accordance with the idea of MKV dynamics. If Φ is constant with respect to the common noise, then the evolution of the distribution is deterministic and the expectation symbol in (4) is superfluous. To alleviate the presentation, we sometimes omit the dependence on ϵ^0 (since it is now the only source of randomness) and see Φ as a stochastic map from $\tilde{\mathbb{A}} \times \mathcal{P}(\mathcal{S})$ to $\mathcal{P}(\mathcal{S})$.

We are facing an MDP over the space $\mathcal{P}(\mathcal{S})$ of probability measures on the underlying state $\mathcal{S}$. Note that $\mathcal{P}(\mathcal{S})$ is always continuous and infinite dimensional unless $\mathcal{S}$ is finite. If $\mathcal{S}$ is finite, the distribution μ_t can be viewed as a vector in $\mathbb{R}^{|\mathcal{S}|}$ whose dynamics can be written as

$$\mu_{t+1}^{\mu_0,a} = P^{\mu_t^{\mu_0,a}, a(\mu_t^{\mu_0,a}), \epsilon_t^0} \mu_t^{\mu_0,a}, \qquad (6)$$

where $P^{\mu_t^{\mu_0,a}, a(\mu_t^{\mu_0,a}), \epsilon_t^0} \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$ is the transition matrix of the distribution, which can be interpreted in terms of the transition rate matrix of a typical player.

Dynamic programming. Let V be the value function associated to the above problem (4), defined by: for $\mu \in \mathcal{P}(\mathcal{S})$,

$$V(\mu) = \sup_{a\in\mathbb{A}} \mathbb{E} \sum_{t=0}^{+\infty} \gamma^t \tilde{f}\Big(\mu_t^{\mu,a}, a(\mu_t^{\mu,a})\Big) \qquad (7)$$

under the constraint: $(\mu_t^{\mu,a})_{t\geq 0}$ solves (5) with initial condition μ. One can check, at least formally, that a dynamic programming principle holds in the sense that V solves the following Bellman equation:

$$\begin{aligned} V(\mu) &= \sup_{a\in\mathbb{A}} \left\{ \tilde{f}(\mu, a(\mu)) + \gamma \mathbb{E} V\left(\Phi^{a(\mu)}(\mu)\right) \right\} \\ &= \sup_{\tilde{a}\in\tilde{\mathbb{A}}} \left\{ \tilde{f}(\mu, \tilde{a}) + \gamma \mathbb{E} V\left(\Phi^{\tilde{a}}(\mu)\right) \right\}, \end{aligned} \qquad (8)$$

where the expectation is over the randomness of Φ, which stems from the common noise.

Moreover, under suitable conditions, we expect a verification theorem to hold, namely, any function satisfying the above Bellman equation (8) coincides with the value function of the MFC problem: $V(\mu_0) = \sup_{a\in\mathbb{A}} J(\mu_0, a)$ for any initial distribution μ_0, and the optimal control is given by the maximizer in (8). The value function and its connection to the so-called Master equation and to the mean-field PDE system has been a very active research direction in recent years, see e.g. [5, 11, 12].

3 Mean-Field Q-Learning

We now turn our attention to the question of *learning* the solution of the MFC problem in a model-free setting, i.e., assuming the model is unknown but one has access to a simulator which can provide samples of trajectories and rewards.

State-action value function. From now on, we see the distribution as the state in $\mathcal{P}(\mathcal{S})$ and the action taken by the central planner given this distribution as an element of $\tilde{\mathbb{A}}$. We then introduce the following state-action value function $Q : \mathcal{P}(\mathcal{S}) \times \tilde{\mathbb{A}} \to \mathbb{R}$ defined, for every $(\mu, \tilde{a}) \in \mathcal{P}(\mathcal{S}) \times \tilde{\mathbb{A}}$ by

$$Q(\mu, \tilde{a}) = \tilde{f}(\mu, \tilde{a}) + \gamma \mathbb{E} \max_{\tilde{a}' \in \tilde{\mathbb{A}}} Q\left(\Phi^{\tilde{a}}(\mu), \tilde{a}'\right), \qquad (9)$$

where we recall that we dropped ϵ^0 from the notation in the expectation and in Φ. Under suitable conditions, V and Q are well defined and $V(\mu) = \sup_{\tilde{a}} Q(\mu, \tilde{a})$. Note that finding one maximizer $\tilde{a}_\mu \in \tilde{\mathbb{A}}$ for each $\mu \in \mathcal{P}(\mathcal{S})$ amounts to find a maximizer in $\mathbb{A}$ for (7).

In the rest of this section, we propose two model-free algorithms relying on this state-action value function. We first focus on the case where $\mathcal{S}$ and $\mathcal{A}$ are finite and we explain later on how to adapt these techniques to the case of continuous spaces.

First method: Tabular MFQ-learning with projection. Note that even when $\mathcal{S}$ is finite, $\mu \in \mathcal{P}(\mathcal{S})$ is an element of the $|\mathcal{S}|$-simplex $\mathbb{S}$, which is a continuous space, and hence a tabular version of Q-learning can not be applied directly. One possible workaround is to first replace $\mathbb{S}$ by a finite subset $\check{\mathbb{S}}$ and then project μ_t on $\check{\mathbb{S}}$ at each time step t, thus letting the mean-field term take only a finite number of values. We can then approximate the function Q by a function $\check{Q} : \check{\mathbb{S}} \times \tilde{\mathbb{A}} \to \mathbb{R}$, which can be represented by a matrix (or a "table") in $\mathbb{R}^{|\check{\mathbb{S}}|\times|\mathcal{A}|^{|\mathcal{S}|}}$ (since $\tilde{\mathbb{A}}$ is the set of functions from $\mathcal{S}$ to $\mathcal{A}$). We introduce the following "projected MFC problem": *Maximize over* $\check{a} \in \check{\mathbb{A}} = \{\check{a} : \check{\mathbb{S}} \times \mathcal{S} \to \mathcal{A}\}$

$$\check{J}(\mu_0, \check{a}) = \mathbb{E} \sum_{t=0}^{+\infty} \gamma^t \tilde{f}\Big(\check{\mu}_t^{\mu_0,\check{a}}, \check{a}(\check{\mu}_t^{\mu_0,\check{a}})\Big) \qquad (10)$$

under the constraint: $\check{\mu}_{t+1}^{\mu_0,\check{a}} = \check{\Phi}^{\check{a}(\check{\mu}_t^{\mu_0,\check{a}}),\epsilon_t^0}(\check{\mu}_t^{\mu_0,\check{a}})$, *where* $\check{\Phi} : \tilde{\mathbb{A}} \times \times \mathcal{E}^0\check{\mathbb{S}} \to \check{\mathbb{S}}$, $(\tilde{a}, e^0, \check{\mu}) \mapsto \mathrm{Proj}_{\check{\mathbb{S}}}\left(\Phi^{\tilde{a},e^0}(\check{\mu})\right)$. This problem is still an optimal control problem of some sequence of distributions, except that at each time step the distribution is pushed forward by $\check{\Phi}^{\tilde{a}} = \mathrm{Proj}_{\check{\mathbb{S}}} \circ \Phi^{\tilde{a}}$ (instead of $\Phi^{\tilde{a}}$) when using control $\tilde{a}$. In this case, a straightforward adaptation of the tabular Q-learning algorithm leads to Algorithm 1. Note that, even in the absence of common noise, this algorithm is possibly stochastic since at each episode, the order in which the state-action pairs are picked is potentially random. In practice, the order could be fixed in advance or stem from a sampled trajectory.

Algorithm 1: Mean-Field Q-learning (MFQ)

Data: A number of episodes N_{epi}; a sequence of learning rates $(\alpha_t)_{t=0,\dots,N_{epi}-1}$.
Result: An approximation of Q on $\check{\mathbb{S}} \times \tilde{\mathbb{A}}$.
begin
  Initialize table $\check{Q}_0 \equiv 0 \in \mathbb{R}^{|\check{\mathbb{S}}|\times|\mathcal{A}|^{|\mathcal{S}|}}$
  for *episode* $t = 0, 1, \dots N_{epi} - 1$ **do**
    Set $\check{Q}_{t+1} \leftarrow \check{Q}_t$
    for $(\check{\mu}, \tilde{a}) \in \check{\mathbb{S}} \times \tilde{\mathbb{A}}$ **do**
      Execute action $\tilde{a}$, observe $\check{\mu}' = \check{\Phi}^{\tilde{a}}(\check{\mu})$ and reward $\tilde{f} = \tilde{f}(\check{\mu}, \tilde{a})$
      Replace $\check{Q}_{t+1}(\check{\mu}, \tilde{a})$ by

$$(1 - \alpha_t(\check{\mu}, \tilde{a}))\check{Q}_t(\check{\mu}, \tilde{a}) + \alpha_t(\check{\mu}, \tilde{a}) \left(\tilde{f} + \gamma \max_{\tilde{a}' \in \tilde{\mathbb{A}}} \check{Q}_t(\check{\mu}', \tilde{a}')\right)$$

  return $\check{Q}_{N_{epi}}$

For this elementary algorithm, as a proof of concept we provide a convergence result for the approximation of the Q-function of the original MFC problem by the table returned at the end of Algorithm 1. To this end, in order to keep the paper at a reasonable length, we will make the following simplifying assumptions.

We endow the simplex $\mathbb{S}$ seen as a subset of $\mathbb{R}^{|\mathcal{S}|}$ with the Euclidean distance denoted by $d_{\mathbb{S}}$.

(A1) Regularity of the data: $\tilde{f}$ is bounded and Lipschitz continuous with respect to $(\mu, \tilde{a})$ with constant $L_{\tilde{f}}$ and Φ is Lipschitz continuous with respect to μ with constant L_Φ, uniformly in $\tilde{a}$ in expectation over the randomness of the common noise, namely: for every $\tilde{a} \in \tilde{\mathbb{A}}, \mu_1, \mu_2 \in \mathbb{S}$,

$$\mathbb{E}_{e^0}\left[d_{\mathbb{S}}\left(\Phi^{\tilde{a},e^0}(\mu_1), \Phi^{\tilde{a},e^0}(\mu_2)\right)\right] \leq L_\Phi d_{\mathbb{S}}(\mu_1, \mu_2).$$

(A2) Regularity of the value function: V is Lipschitz continuous wrt μ with constant L_V.

(A3) Simplex discretization: There exists $\epsilon_S > 0$ s.t.: $\forall \mu \in \mathbb{S}, \exists \check{\mu} \in \check{\mathbb{S}}$ s.t. $d_{\mathbb{S}}(\mu, \check{\mu}) \leq \epsilon_S$.

(A4) Covering time: There exists a finite T_{cov} such that with probability $1/2$ (over the randomness of the common noise and of Algorithm 1) the following holds: For every starting point in $\check{\mathbb{S}} \times \tilde{\mathbb{A}}$, every element of $\check{\mathbb{S}} \times \tilde{\mathbb{A}}$ has been visited before time T_{cov} during the execution of Algorithm 1.

(A5) Learning rates: There exists $\kappa \in (1/2, 1)$ such that for every $(\check{\mu}, \tilde{a}) \in \check{\mathbb{S}} \times \tilde{\mathbb{A}}$, $\alpha_t(\check{\mu}, \tilde{a}) = 1/\big(1 + n(\check{\mu}, \tilde{a}, t)\big)^\kappa$ for each $t \geq 0$, where $n(\check{\mu}, \tilde{a}, t)$ is the number of times up to t that the pair $(\check{\mu}, \tilde{a})$ has been visited in Algorithm 1.

The regularity of V in **(A2)** can typically be ensured through suitable conditions on the data of the problem, as e.g. in [17, 9, 12], while to obtain **(A3)**, one can consider an ϵ_S-net as in [24]. Assumption **(A4)** holds for instance either by using exploring starts (if the learner can query an oracle which simulates transitions from any $(\mu, \tilde{a})$), or by following a long enough sampled trajectory (provided some form of irreducibility or ergodicity of the dynamics, ensuring full exploration). Note that the boundedness of the running reward $\tilde{f}$ from Assumption **(A1)** together with the fact that $\gamma \in (0, 1)$ ensures the existence of a finite upper bound $\check{V}_{max}$ for the value function of the projected MFC problem. We denote by $\beta = (1-\gamma)/2$ the horizon of the MDP corresponding to the projected MFC problem, and for $\delta \in (0, 1)$, we let $T_{cov}(\delta) = \lceil T_{cov} \log_2(1/(2\delta)) \rceil$.

Theorem 2. *Let $\delta \in (0, 1)$ and $\epsilon > 0$. Under Assumptions **(A1)**–**(A5)**, if the number of episodes N_{epi} is of order*

$$\Omega\left(\left(\frac{(T_{cov}(\delta))^{1+3\kappa}\check{V}_{max}^2 \ln\left(|\check{\mathbb{S}}|\,|\mathcal{A}|^{|\mathcal{S}|}\check{V}_{max}/(2\delta\beta\epsilon)\right)}{\beta^2\epsilon^2}\right)^{\frac{1}{\kappa}} + \left(\frac{(T_{cov}(\delta))}{\beta}\ln\left(\frac{\check{V}_{max}}{\epsilon}\right)\right)^{\frac{1}{1-\kappa}}\right), \tag{11}$$

then with probability $1-\delta$, for all $(\mu, \tilde{a}) \in \mathbb{S} \times \tilde{\mathbb{A}}$,

$$|\check{Q}_{N_{epi}}(\mathrm{Proj}_{\check{\mathbb{S}}}(\mu), \tilde{a}) - Q(\mu, \tilde{a})| \leq \epsilon',$$

where $\epsilon' = \epsilon + \left[\frac{\gamma(2-\gamma)}{1-\gamma} L_V(1 + L_\Phi) + L_{\tilde{f}}\right]\epsilon_S$.

Note that ϵ can be chosen as small as desired provided N_{epi} is large enough. The second term in the error ϵ' is proportional to ϵ_S, which is somehow unavoidable in general due to the projection on $\check{\mathbb{S}}$. However, this error vanishes as $\epsilon_S \to 0$, i.e., as $\check{\mathbb{S}}$ is better and better an approximation of $\mathbb{S}$.

The proof is deferred to Section C of the appendix. It relies on the following three steps: **(1)** Since N_{epi} is large enough, then $\check{Q}_{N_{epi}} \approx \check{Q}$ on $\check{\mathbb{S}} \times \tilde{\mathbb{A}}$; **(2)** $\check{Q} \approx Q$ on $\check{\mathbb{S}} \times \tilde{\mathbb{A}}$; **(3)** For every $(\mu, \tilde{a}) \in \mathbb{S} \times \tilde{\mathbb{A}}$, $Q(\mathrm{Proj}_{\check{\mathbb{S}}}(\mu), \tilde{a}) \approx Q(\mu, \tilde{a})$. The first step relies on standard Q-learning convergence results [19], while the two other steps stem from the regularity assumptions and the approximation of $\mathbb{S}$ by $\check{\mathbb{S}}$.

Let us now derive a consequence in terms of the control function. We will use the following additional assumption on the gap between the values of the best and second-best actions, which is rather standard in approximation algorithms based on tabular Q-functions [20, 4].

(B) Action gap: There exists $K_A > 0$ such that: For every $\check{\mu} \in \check{\mathbb{S}}$ and $\tilde{a} \in \tilde{\mathbb{A}} \setminus \arg\max Q(\check{\mu}, \cdot)$, $\max_{\tilde{\mathbb{A}}} Q(\check{\mu}, \cdot) - Q(\check{\mu}, \tilde{a}) \geq K_A$.

For $\tau > 0$ and $x \in \mathbb{R}^n$, we use the notation

$$\mathrm{softmax}_\tau(x) = (e^{\tau x_1}, \ldots, e^{\tau x_n}) / \sum_j e^{\tau x_j},$$

and

$$\mathrm{argmaxe}(x) = (\mathbf{1}_{i \in \arg\max(x)})_{i=1,\ldots,n} / |\arg\max(x)|.$$

Corollary 3. *In the setting of Theorem 2, if in addition Assumption **(B)** holds, then for every $\check{\mu} \in \check{\mathbb{S}}$,*

$$\|\text{softmax}_\tau\left(\check{Q}_{N_{epi}}(\check{\mu},\cdot)\right) - \text{argmaxe}\left(Q(\check{\mu},\cdot)\right)\|_2 \leq \tau\epsilon' + 2|\tilde{\mathbb{A}}|e^{-\tau K_A},$$

where $\check{Q}_{N_{epi}}$ is the table returned by Algorithm 1 and ϵ' is the error term appearing in Theorem 2.

The proof is deferred to Section C of the appendix. The argmaxe in the second term is here in case there are several optimal controls. The softmax regularizes the best action predicted by the estimation $\check{Q}_{N_{epi}}$ of the function Q.

Second method: DDPG for MFC. Although the elementary structure of the above method allows us to analyze it, one drawback is that it requires the computation of a projection on a discretization of the simplex at each time step, which can be quite cumbersome and computationally costly when the number of states in $\mathcal{S}$ is large. For this reason, we propose a different RL method based on an approximation of the Q-function by neural networks which can deal with inputs and outputs in continuous spaces. Still in the case of finite $\mathcal{S}$ and $\mathcal{A}$, since the state of distributions $\mathcal{P}(\mathcal{S})$ and the set of actions $\mathcal{A}^{\mathcal{S}}$ are finite dimensional, we can try to approximate the Q-function by a neural network taking inputs in $\mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|^{|\mathcal{S}|}}$ and outputting a real number. For the learning procedure, we employ the Deep Deterministic Policy Gradient (DDPG) proposed in [33]. It relies on two neural networks, one for the Q-function (the critic) and one for the policy (the actor). The heart of the algorithm consists in updating alternatively the critic by minimizing an empirical square error and the actor by making one step of gradient descent. To improve exploration, a Gaussian noise is added to the action prescribed by the actor, and for more stability, target networks are also added. The algorithm is summarized in the appendix (see Algorithm 2).

Adaptation to continuous spaces. When the underlying state space $\mathcal{S}$ is continuous, the distribution μ_t is an element of the infinite-dimensional space $\mathcal{P}(\mathcal{S})$. In order to develop a reinforcement learning method, we thus need some kind of finite-dimensional approximation. Motivated by applications to physical models such as swarm of robots or drones in which the underlying space $\mathcal{S}$ is in low dimension (typically $\mathcal{S} \subset \mathbb{R}^d$ with $d = 1, 2$ or 3), we propose to represent μ_t by a histogram of its values at a finite number of points. In other words, we consider a finite number N_p of points in $\mathcal{S}$ and let $M_t \in \mathbb{R}^{N_p}$ be a vector which approximates μ_t. Similarly, an action $a \in \tilde{\mathbb{A}}$ can be approximated by its values on the N_p points chosen in $\mathcal{S}$ and can hence be represented by a vector in $\mathbb{R}^{N_p}$. The problem then reduces to the finite-state case discussed above.

4 Numerical Examples

General setup. We present numerical results obtained using the second method introduced above. We assumed that the central planner has access to an oracle which can simulate the evolution of M_t as discussed at the end of the previous section. In the numerical implementation, we provided the learning algorithm with a black-box which computes transitions of M_t. This oracle has been implemented using a finite-difference scheme for Kolmorov-Fokker-Planck equations, in line with recent research on numerical methods for mean field games [1]. The actor and critic networks have been implemented using a feedforward fully connected architecture with 2 hidden layers of width less than 300 neurons. We used random initial states at each episode, and the noise used on the action is a Gaussian noise with mean 0 and variance 0.02. We used Adam optimizer with initial learning rate 0.0001 and minibatches of size 16. For the sake of clarity and in order to benchmark the aforementioned method, we provide illustrations on examples without common noise but analogous results can also be obtained in the presence of a common noise.

Example 1: Cyber security model. For a first testbed, we start with a finite state problem. We revisit the cyber security example introduced in [28], but here from the point of view of a central planner (such as a large company or a state) trying to protect its computers against the attacks of a hacker. The situation can hence be phrased as a mean field control problem.

In this model, the population consists of a large group of computers which can be either defended (D) or undefended (U) and either infected (I) or susceptible (S) of infection. The set $\mathcal{S}$ has hence four elements corresponding to the four possible combinations: DI, DS, UI, US. The action set is $\mathcal{A} = \{0, 1\}$, where 0 is interpreted as the fact that the central planner is satisfied with the

current level of protection (D or U) of the computer under consideration, whereas 1 means that she wants to change this level of protection. In the latter case, the update occurs at a (fixed) rate $\lambda > 0$. At each of the four states, all the computers are indistinguishable and hence the central planner only chooses one action per state and applies it to all the computers at that state. When infected, each computer may recover at rate q^D_{rec} or q^U_{rec} depending on whether it is defended or not. On the other hand, a computer may be infected either directly by a hacker, at rate $v_H q^D_{inf}$ (resp. $v_H q^U_{inf}$) if it is defended (resp. undefended), or by undefended infected computers, at rate $\beta_{UU}\mu(\{UI\})$ (resp. $\beta_{UD}\mu(\{UI\})$) if it is undefended (resp. defended), or by defended infected computers, at rate $\beta_{DU}\mu(\{DI\})$ (resp. $\beta_{DD}\mu(\{DI\})$) if it is undefended (resp. defended).

The transition matrix from (6) is given by

$$P^{\mu,a} = \begin{pmatrix} \dots & P^{\mu,a}_{DS\to DI} & \lambda a & 0 \\ q^D_{rec} & \dots & 0 & \lambda a \\ \lambda a & 0 & \dots & P^{\mu,a}_{US\to UI} \\ 0 & \lambda a & q^U_{rec} & \dots \end{pmatrix}$$

where

$$P^{\mu,a}_{DS\to DI} = v_H q^D_{inf} + \beta_{DD}\mu(\{DI\}) + \beta_{UD}\mu(\{UI\}),$$
$$P^{\mu,a}_{US\to UI} = v_H q^U_{inf} + \beta_{UU}\mu(\{UI\}) + \beta_{DU}\mu(\{DI\}),$$

and all the instances of $\dots$ should be replaced by the negative of the sum of the entries of the row in which $\dots$ appears on the diagonal. At each time step, the central planner pays a protection cost $k_D > 0$ for each defended computer, and a penalty $k_I > 0$ for each infected computer. The total reward at time t is hence $\tilde{f}_t = -\left[k_D\mu_t(\{DI, DS\}) + k_I\mu_t(\{DI, UI\})\right]$.

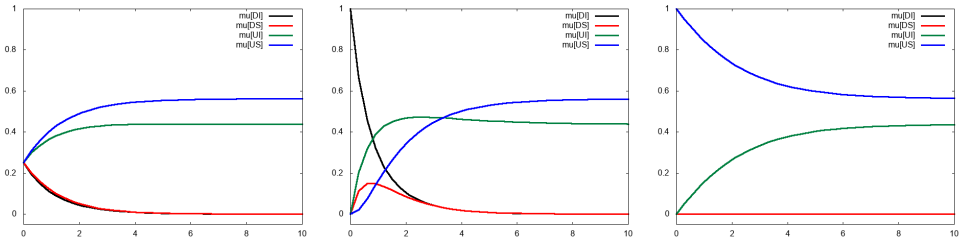
Although the underlying state space $\mathcal{S}$ is finite, instead of using MFQ (see Algorithm 1), we use the second method introduced above based on DDPG. In the present case, this approach has the advantage to avoid discretizing $\mathcal{P}(\mathcal{S})$ since we instead deal directly with the distribution as a vector in dimension 4. The DDPG method learns a control, that we then apply to the three reference cases studied in [11, Section 7.2.3] (for the sake of brevity, we do not reproduce here the values of the parameters). The resulting distribution's evolution are shown in Figure 2. We can see that we recover the same evolution for the three initial distributions considered, namely $(0.25, 0.25, 0.25, 0.25)$, $(1, 0, 0, 0)$ and $(0, 0, 0, 1)$. In particular, at $T = 10$, we obtain in all three simulations the distribution $\mu_{10} = (0.0, 0.0, 0.4376, 0.5624)$, which is close to the values found in [11, Section 7.2] for the stationary distribution, namely $(0.0, 0.0, 0.44, 0.56)$.

Example 2: Swarm motion. We then turn our attention to a model in continuous space. More precisely, let us consider a model of swarm motion with aversion to crowded regions introduced in [3] (in the context of mean field games). Since here we simply want to provide a proof of principle for our method, we take (as in the aforementioned work) the interval $[0, 1]$ with periodic boundary condition (i.e. the unit torus) as the state space $\mathcal{S}$. The dynamics of a typical agent is driven by (3) with $b(x, \mu, a) = a$. In other words, the central planner chooses the velocity of each agent. The instantaneous reward (appearing in (1)) of a typical agent at location x and using action a while the population's state is μ, is defined as $f(x, \mu, a) = -\frac{1}{2}|a|^2 + \varphi(x) - \ln(\mu(x))$. Here, the first term penalizes a large velocity (it can be interpreted as a kind of cost proportional to the kinetic energy of the agent), φ encodes a preference for certain positions in space, and the last term models crowd aversion since it penalizes the fact of being at a location where the density of agents is high. By choosing

$$\varphi(x) = -2\pi^2\left[-\sin(2\pi x) + |\cos(2\pi x)|^2\right] + 2\sin(2\pi x),$$

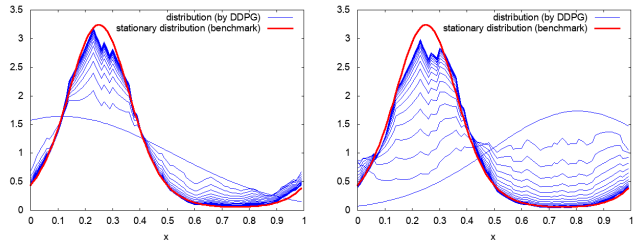
we obtain a model for which, when ϵ_t have Gaussian distribution and $\epsilon^0 \equiv 0$, the MFC admits an explicit ergodic solution that we can use as a benchmark. Indeed, in this case the optimal ergodic control is given by $\tilde{a}(x) = 2\pi\cos(2\pi x)$ and the ergodic distribution of the corresponding MKV dynamics has density $\mu(x) = e^{2\sin(2\pi x)}/\int e^{2\sin(2\pi x')}dx'$.

To implement the second RL method described above, we discretize $[0, 1]$ with a mesh of N_p points and use a finite difference scheme to simulate the evolution of the dynamics. The DDPG method uses this as a black-box and, for a given action $\tilde{a} \in \mathbb{R}^{N_p}$, can only access the resulting new distribution and the associated reward. Figure 3 presents results obtained using this method after 3000 episodes. The system has been trained on initial distributions which are Gaussian with random mean and random variance. As illustrated in the figures, the system has learnt how to drive this type of initial distributions towards the analytical stationary distribution and then how to use an approximation of the stationary optimal control in order to keep the system in the stationary regime.

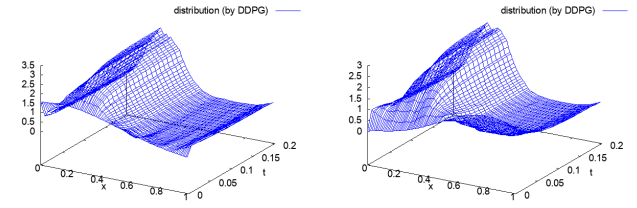


(a) Test 1 (b) Test 2 (c) Test 3

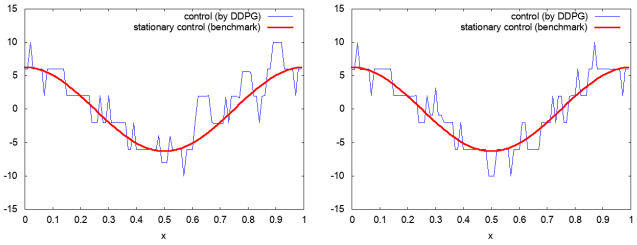
Figure 2: Cyber security example: Evolution of the distribution when applying the control learnt by DDPG.



(a) μ_t at several t (b) μ_t at several t



(c) Evolution of μ_t (d) Evolution of μ_t



(e) Control learnt (f) Control learnt

Figure 3: Swarm motion: Evolution of the distribution and control learnt for two different initial distributions.

5 Conclusion and future research

In this work, we explored the central role played by MKV dynamics in multi-agent RL. We developed a framework and model-free methods to learn mean field optimal control. As a proof of principle, we established a rate of convergence for a Q-learning method and our numerical tests assess empirical convergence of an actor-critic method on examples from the literature. An important feature of our model is the presence of common noise, whose impact had to be controlled.

Our results can be extended in several directions. The analysis and the numerical implementations could be applied to other mean-field problems such as mean field games or mean field control problems with several populations, with important applications to multi-agent sweeping and tracking. The proof of convergence of the actor-critic method is also postponed for future work. Last, it would be interesting to investigate other types of simulators, such as Monte-Carlo simulators based on samples of a finite population.

Related work. Our work is at the intersection of RL and MFC. The latter has recently attracted a lot of attention, particularly since the introduction of mean field games (MFG) by Lasry and Lions [2006a, 2006b, 2007] and by Caines, Huang and Malhamé [2006, 2007]. MFGs correspond to the asymptotic limit of Nash equilibria for games with mean field interactions. They are defined through a fixed point procedure and hence, differ both conceptually and numerically, from MFC problems which correspond to social optima, see e.g. [11] for details. Most works on learning in the presence of mean field interactions have focused on MFGs, see e.g. [40, 10] for "learning" (or rather *solving*) MFGs based on the full knowledge of the model, and [27, 38, 39, 24, 34, 18, 37, 21] for RL based methods.

In contrast, our work focuses on MFC problems. Along these lines, we have studied policy gradient methods for MFC in [16]. However, this work was restricted to linear-quadratic models. While completing the present work, we became aware of the very recent work [36], which studies MFC with policy gradient methods too. However, their work is restricted to finite state and action spaces whereas we also consider continuous spaces. Furthermore, we provide a rate of convergence (see Theorem 2). We also stress that although some tools are common, our work differs significantly from [24, 18] because the latter works deal with a mean field game. Their learning procedure is embedded in a fixed point on the distribution and, for this reason, the Q-learning step is only needed to solve a classical control problem and not a mean field one. Here, the key novelty is that our learning methods are designed for MDPs on the space of measures. Last, we would also like to mention that the first two authors have recently proposed machine learning methods for *solving* mean field control problems and mean field games, see [14, 15]. These methods are based on the knowledge of the model, since one relies on it to compute gradients of the cost functional and implement a stochastic gradient descent. The present work can be viewed as an extension of these methods, where one tries to be free from the model and *learn* the solution by trial and error.

Acknowledgements

M.L. is grateful to Matthieu Geist and Julien Pérolat for helpful discussions on the DDPG algorithm.

References

[1] Achdou, Y. and Capuzzo-Dolcetta, I. (2010). Mean field games: numerical methods. *SIAM J. Numer. Anal.*, 48(3):1136–1162.

[2] Alasseur, C., Ben Tahar, I., and Matoussi, A. (2017). An extended mean field game for storage in smart grids. *arXiv preprint arXiv:1710.08991*.

[3] Almulla, N., Ferreira, R., and Gomes, D. (2017). Two numerical approaches to stationary mean-field games. *Dyn. Games Appl.*, 7(4):657–682.

[4] Bellemare, M. G., Ostrovski, G., Guez, A., Thomas, P. S., and Munos, R. (2016). Increasing the action gap: New operators for reinforcement learning. In *Thirtieth AAAI Conference on Artificial Intelligence*.

[5] Bensoussan, A., Frehse, J., and Yam, S. C. P. (2013). *Mean field games and mean field type control theory*. Springer Briefs in Mathematics. Springer, New York.

[6] Bensoussan, A., Frehse, J., and Yam, S. C. P. (2015). The master equation in mean field theory. *J. Math. Pures Appl. (9)*, 103(6):1441–1474.

[7] Bertsekas, D. P. (2012). *Dynamic programming and optimal control. Vol. II. Approximate dynamic programming*. Athena Scientific, Belmont, MA, fourth edition.

[8] Bossy, M. and Talay, D. (1997). A stochastic particle method for the McKean-Vlasov and the Burgers equation. *Math. Comp.*, 66(217):157–192.

[9] Cardaliaguet, P., Delarue, F., Lasry, J.-M., and Lions, P.-L. (2019). *The master equation and the convergence problem in mean field games*, volume 201 of *Annals of Mathematics Studies*. Princeton University Press, Princeton, NJ.

[10] Cardaliaguet, P. and Hadikhanloo, S. (2017). Learning in mean field games: the fictitious play. *ESAIM: Control, Optimisation and Calculus of Variations*, 23(2):569–591.

[11] Carmona, R. and Delarue, F. (2018a). *Probabilistic theory of mean field games with applications. I*, volume 83 of *Probability Theory and Stochastic Modelling*. Springer, Cham. Mean field FBSDEs, control, and games.

[12] Carmona, R. and Delarue, F. (2018b). *Probabilistic theory of mean field games with applications. II*, volume 84 of *Probability Theory and Stochastic Modelling*. Springer, Cham. Mean field games with common noise and master equations.

[13] Carmona, R., Fouque, J.-P., and Sun, L.-H. (2015). Mean field games and systemic risk. *Commun. Math. Sci.*, 13(4):911–933.

[14] Carmona, R. and Laurière, M. (2019a). Convergence analysis of machine learning algorithms for the numerical solution of mean field control and games: I - the ergodic case. *arXiv preprint arXiv:1907.05980*.

[15] Carmona, R. and Laurière, M. (2019b). Convergence analysis of machine learning algorithms for the numerical solution of mean field control and games: II - the finite horizon case. *arXiv preprint arXiv:1908.01613*.

[16] Carmona, R., Laurière, M., and Tan, Z. (2019). Linear-quadratic mean-field reinforcement learning: Convergence of policy gradient methods. *arXiv preprint arXiv:1910.04295*.

[17] Chassagneux, J.-F., Crisan, D., and Delarue, F. (2014). A probabilistic approach to classical solutions of the master equation for large population equilibria. arXiv:1411.3009.

[18] Elie, R., Pérolat, J., Laurière, M., Geist, M., and Pietquin, O. (2019). Approximate fictitious play for mean field games. *arXiv preprint arXiv:1907.02633*.

[19] Even-Dar, E. and Mansour, Y. (2003). Learning rates for Q-learning. *J. Mach. Learn. Res.*, 5:1–25.

[20] Farahmand, A.-m. (2011). Action-gap phenomenon in reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 172–180.

[21] Fu, Z., Yang, Z., Chen, Y., and Wang, Z. (2019). Actor-critic provably finds nash equilibria of linear-quadratic mean-field games. *arXiv preprint arXiv:1910.07498*.

[22] Gao, B. and Pavel, L. (2017). On the properties of the softmax function with application in game theory and reinforcement learning. *arXiv preprint arXiv:1704.00805*.

[23] Gast, N., Gaujal, B., and Le Boudec, J.-Y. (2012). Mean field for markov decision processes: from discrete to continuous optimization. *IEEE Transactions on Automatic Control*, 57(9):2266–2280.

[24] Guo, X., Hu, A., Xu, R., and Zhang, J. (2019). Learning mean-field games. *arXiv preprint arXiv:1901.09585*.

[25] Huang, M., Caines, P. E., and Malhamé, R. P. (2007). Large-population cost-coupled LQG problems with nonuniform agents: individual-mass behavior and decentralized ϵ-Nash equilibria. *IEEE Trans. Automat. Control*, 52(9):1560–1571.

[26] Huang, M., Malhamé, R. P., and Caines, P. E. (2006). Large population stochastic dynamic games: closed-loop McKean-Vlasov systems and the Nash certainty equivalence principle. *Commun. Inf. Syst.*, 6(3):221–251.

[27] Iyer, K., Johari, R., and Sundararajan, M. (2014). Mean field equilibria of dynamic auctions with learning. *Management Science*, 60(12):2949–2970.

[28] Kolokoltsov, V. N. and Bensoussan, A. (2016). Mean-field-game model for botnet defense in cyber-security. *Appl. Math. Optim.*, 74(3):669–692.

[29] Lasry, J.-M. and Lions, P.-L. (2006a). Jeux à champ moyen. I. Le cas stationnaire. *C. R. Math. Acad. Sci. Paris*, 343(9):619–625.

[30] Lasry, J.-M. and Lions, P.-L. (2006b). Jeux à champ moyen. II. Horizon fini et contrôle optimal. *C. R. Math. Acad. Sci. Paris*, 343(10):679–684.

[31] Lasry, J.-M. and Lions, P.-L. (2007). Mean field games. *Jpn. J. Math.*, 2(1):229–260.

[32] Laurière, M. and Pironneau, O. (2016). Dynamic programming for mean-field type control. *J. Optim. Theory Appl.*, 169(3):902–924.

[33] Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. (2016). Continuous control with deep reinforcement learning. In *Proceedings of the International Conference on Learning Representations (ICLR 2016)*.

[34] Mguni, D., Jennings, J., and de Cote, E. M. (2018). Decentralised learning in systems with many, many strategic agents. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[35] Pham, H. and Wei, X. (2017). Dynamic programming for optimal control of stochastic McKean-Vlasov dynamics. *SIAM J. Control Optim.*, 55(2):1069–1101.

[36] Subramanian, J. and Mahajan, A. (2019). Reinforcement learning in stationary mean-field games. In *Proceedings. 18th International Conference on Autonomous Agents and Multiagent Systems*.

[37] Tiwari, N., Ghosh, A., and Aggarwal, V. (2019). Reinforcement learning for mean field game. *arXiv preprint arXiv:1905.13357*.

[38] Yang, J., Ye, X., Trivedi, R., Xu, H., and Zha, H. (2018a). Deep mean field games for learning optimal behavior policy of large populations. In *International Conference on Learning Representations*.

[39] Yang, Y., Luo, R., Li, M., Zhou, M., Zhang, W., and Wang, J. (2018b). Mean field multi-agent reinforcement learning. In *International Conference on Machine Learning*, pages 5567–5576.

[40] Yin, H., Mehta, P. G., Meyn, S. P., and Shanbhag, U. V. (2013). Learning in mean-field games. *IEEE Transactions on Automatic Control*, 59(3):629–644.

A Derivation of Bellman equation

Although the derivation of the Bellman equation (7) is rather standard, see e.g. [7], we include it for the sake of completeness.

To alleviate the presentation, we assume that the running reward f is bounded by a constant C_f and we focus on stationary controls. To avoid measurability issues, we assume that the set $\mathcal{E}^0$ is countable. More general settings could be considered at the expense of technicalities which are beyond the scope of the present work. We rewrite the value function as

$$V(\mu) = \lim_{n\to+\infty} \sup_{a\in\mathbb{A}} \mathbb{E} \sum_{t=0}^{n} \gamma^t \tilde{f}(\mu_t, a(\mu_t)),$$

which leads us to introduce the following dynamic programming operators defined, for a bounded function v and $a \in \mathbb{A}$, by

$$\begin{aligned}(\mathcal{T}_a v)(\mu) &= \tilde{f}(\mu, a(\mu)) + \gamma \mathbb{E} v(\Phi^{a(\mu)}(\mu)),\\ (\mathcal{T} v)(\mu) &= \sup_{a\in\mathbb{A}} \{\mathcal{T}_a v\}.\end{aligned}$$

The above expression for V then rewrites

$$V(\mu) = \lim_{n\to+\infty} (\mathcal{T}^n 0)(\mu),$$

where $(\mathcal{T}^n 0)$ represents the result of $\mathcal{T}$ composed n times and applied to the function which is identically 0 on $\mathcal{P}(\mathcal{S})$. Thanks to the bound on f, one can check that for every $\mu \in \mathcal{P}(\mathcal{S})$ and every integer n,

$$V(\mu) - \frac{\gamma^n}{1-\gamma} C_f \le (\mathcal{T}^n 0)(\mu) \le V(\mu) + \frac{\gamma^n}{1-\gamma} C_f,$$

Applying $\mathcal{T}$ to the above relation yields

$$(\mathcal{T}V)(\mu) - \frac{\gamma^{n+1}}{1-\gamma} C_f \le (\mathcal{T}^{n+1} 0)(\mu) \le (\mathcal{T}V)(\mu) + \frac{\gamma^{n+1}}{1-\gamma} C_f.$$

By taking $n \to +\infty$, we deduce

$$(\mathcal{T}V)(\mu) = \lim_{n\to+\infty} (\mathcal{T}^{n+1}0)(\mu) = V(\mu),$$

which is the desired relation. One can also check that the solution to (7) is unique. Similarly, denoting $\tilde{J}(a)(\mu) = J(\mu, a)$, we have for any $a \in \mathbb{A}$ and $\mu \in \mathcal{P}(\mathcal{S})$, $\tilde{J}(a)$ is the only fixed point of $\mathcal{T}_a$.

In addition, let us consider an stationary control a^* which is optimal in the sense that, for any $\mu_0 \in \mathcal{P}(\mathcal{S})$, it achieves the supremum over $a \in \mathbb{A}$ in (4), i.e., $\tilde{J}(a^*) = V$. Then, using the Bellman equations for V and $\tilde{J}(a^*)$, we have

$$\begin{aligned}(\mathcal{T}V)(\mu_0) &= V(\mu_0)\\ &= \tilde{J}(a^*)(\mu_0)\\ &= (\mathcal{T}_{a^*}\tilde{J}(a^*))(\mu_0)\\ &= (\mathcal{T}_{a^*}V)(\mu_0).\end{aligned}$$

Conversely, assume a is such that for every μ_0,

$$(\mathcal{T}V)(\mu_0) = (\mathcal{T}_a V)(\mu_0).$$

Then, since $(\mathcal{T}V)(\mu_0) = V(\mu_0)$, we obtain that $V(\mu_0) = (\mathcal{T}_a V)(\mu_0)$. By uniqueness of the fixed point of $\mathcal{T}_a$, we have that $V = \tilde{J}(a)$ and hence a is optimal.

B Link with MDPs arising in Mean Field Games

Here, we clarify the difference between the Mean Field MDPs studied in the present work and the ones arising in the context of finite state Mean Field Games [38, 24]. Although both problems involve the term "mean field", we argue that in a MFG, the MDP is a rather standard one.

In [38], the authors make the following point (at the beginning of their Section 4): *given the evolution* of the population (i.e., the mean-field term), an infinitesimal player solves a (standard) optimal control problem, to which corresponds a (standard) MDP. In other words, the population's distribution appears as a given parameter in the MDP and not as the state over which the optimization is performed, as in our case. To emphasize, in our notation, the difference between their setting and ours, let us consider the evolution (6) for a finite state space $\mathcal{S} = \{1, 2, \dots, |\mathcal{S}|\}$. As in [38], let us assume that the players control directly their transition probabilities. In other words, the action space is the set of probability distributions on $\mathcal{S}$, and all the players in a state $x \in \mathcal{S}$ control the probability with which they will go to each other state $x' \in \mathcal{S}$. In this case, each element of $\mathcal{A}$ can be identified with a vector of length $|\mathcal{S}|$ representing a probability distribution on $\mathcal{S}$, and each element of $\tilde{a} \in \tilde{\mathbb{A}}$ can be identified with a matrix $P^{\tilde{a}}$ such that $P^{\tilde{a}}_{x',x} = \tilde{a}(x)(x')$. In [38], the initial distribution μ_0 is fixed (hence we omit to denote explicitly the dependence on μ_0), and there is no common noise. In this case, one can look for feedback controls which are functions of time and x only. Then the MFC problem (4) rewrites: *Find* $\tilde{\boldsymbol{a}} = (\tilde{a}_t)_{t\geq 0}, a_t \in \mathbb{A}$ *maximizing*

$$\begin{aligned}J(\tilde{\boldsymbol{a}}) &= \sum_{t=0}^{+\infty} \gamma^t \mathbb{E}_{x\sim\mu_t^{\tilde{\boldsymbol{a}}}}\left[f\Big(x, \mu_t^{\tilde{\boldsymbol{a}}}, \tilde{a}_t(x)\Big)\right]\\ &= \sum_{t=0}^{+\infty} \gamma^t \sum_{x\in\mathcal{S}} \mu_t^{\tilde{\boldsymbol{a}}}(x) f\Big(x, \mu_t^{\tilde{\boldsymbol{a}}}, \tilde{a}_t(x)\Big),\end{aligned}$$

under the constraint: $\mu_0^{\tilde{\boldsymbol{a}}} = \mu_0$ *and for* $t \geq 0$,

$$\mu_{t+1}^{\tilde{\boldsymbol{a}}} = P^{\tilde{a}_t}\mu_t^{\tilde{\boldsymbol{a}}}, \tag{12}$$

where $P^{\cdot}$ *is as defined above.* The corresponding MFG, analogous to the one studied in [38], can be formulated as follows: *Find a sequence of distributions* $\boldsymbol{m} = (m_t)_{t\geq 0}$ *and a sequence of controls* $\tilde{\boldsymbol{a}} = (\tilde{a}_t), \tilde{a}_t \in \mathbb{A}$ *such that: (1)* $\tilde{\boldsymbol{a}}$ *maximizes*

$$J^{MFG}(\tilde{\boldsymbol{a}}; \boldsymbol{m}) = \sum_{t=0}^{+\infty} \gamma^t \sum_{x\in\mathcal{S}} \mu_t^{\tilde{\boldsymbol{a}},\boldsymbol{m}}(x) f\Big(x, m_t, \tilde{a}_t(x)\Big),$$

under the constraint: $\mu_0^{\tilde{\boldsymbol{a}},\boldsymbol{m}} = \mu_0$ *and for* $t \geq 0$,

$$\mu_{t+1}^{\tilde{\boldsymbol{a}},\boldsymbol{m}} = P^{\tilde{a}_t}\mu_t^{\tilde{\boldsymbol{a}},\boldsymbol{m}}, \tag{13}$$

and (2) for every $t \geq 0$, $m_t = \mu_0^{\tilde{\boldsymbol{a}},\boldsymbol{m}}$. Step (1) above corresponds to the problem faced by a typical agent, when the evolution of the population is given by $\boldsymbol{m}$. The dynamics (13) can be viewed as an MDP on the space of measures, but the evolution is purely linear, in the sense that at time t, the state of the MDP, namely $\mu_t^{\tilde{\boldsymbol{a}},\boldsymbol{m}}$, is not involved in the transition matrix, namely $P^{\tilde{a}_t}$. In contrast, the dynamics (6) is in general non-linear.

In [24], the authors build the MDP upon the same insight in a different way. To stress the difference with our notion of MDP, let us go back to the MFC problem (1) and consider again a finite state space $\mathcal{S}$ and no common noise. The corresponding MFG would be: *Find a flow of distributions* $\boldsymbol{m} = (m_t)_{t\geq 0}$ *and a policy* π *such that: (1)* π *maximizes*

$$J^{MFG}(\pi; \boldsymbol{m}) = \mathbb{E}\left[\sum_{t=0}^{+\infty} \gamma^t f(x_t^{\pi,\boldsymbol{m}}, m_t, \pi_t)\right]$$

where the state process x *has initial distribution* μ_0 *and dynamics*

$$\mathbb{P}\left(x_{t+1}^{\pi,\boldsymbol{m}} = x' \,|\, x_t^{\pi,\boldsymbol{m}}, m_t, \pi_t\right) = p_{x_t^{\pi,\boldsymbol{m}}, m_t, \pi_t}(x'),$$

and (2) For every $t \geq 0$, m_t *is the distribution of* $x_t^{\pi,\boldsymbol{m}}$. As mentioned above, in the optimization problem of a typical player, the flow of distributions $\boldsymbol{m}$ appears as a given parameter. The corresponding MDP is hence parameterized by

this $\boldsymbol{m}$ but evolves in the finite state space $\mathcal{S}$. Furthermore, in [24], although they consider interactions through the control's distributions (omitted here to alleviate the notations), the authors focus on a stationary Nash equilibrium. In other words, they look for a solution of the following type of problems: *Find $m \in \mathcal{P}(\mathcal{S})$ and $\pi = (\pi_t)_{t\geq 0}$ such that, letting $\boldsymbol{m}^\infty = (m, m, \dots)$, we have: (1) π maximizes*

$$J^{MFG}(\pi; \boldsymbol{m}^\infty) = \mathbb{E}\left[\sum_{t=0}^{+\infty} \gamma^t f(x_t^{\pi,m}, m, \pi_t)\right]$$

where the state process has initial distribution m and dynamics

$$\mathbb{P}\left(x_{t+1}^{\pi,\boldsymbol{m}^\infty} = x' \,|\, x_t^{\pi,\boldsymbol{m}^\infty}, m, \pi_t, \epsilon_t^0\right) = p_{x_t^{\pi,\boldsymbol{m}^\infty}, m, \pi_t, \epsilon_t^0}(x'),$$

and (2) For every $t \geq 0$, m is the distribution of $x_t^{\pi,\boldsymbol{m}^\infty}$. In this case, the rewards and dynamics of an infinitesimal player is parameterized by a single distribution m and the same remark holds for the corresponding MDP.

C Proof of the convergence results

Proof of Theorem 2. Let us denote by $\check{V}$ and $\check{Q}$ respectively the state value function and the state-action value function of the projected MFC problem. We split the proof into several steps.

Step 1. If N_{epi} is large enough, then for every $(\check{\mu}, \tilde{a}) \in \check{\mathbb{S}} \times \tilde{\mathbb{A}}$,

$$\check{Q}_{N_{epi}}(\check{\mu}, \tilde{a}) \approx \check{Q}(\check{\mu}, \tilde{a}).$$

This comes from standard convergence results on Q-learning for finite state-action spaces. More precisely, under Assumptions **(A1)**, **(A4)** and **(A5)**, we can apply Theorem 4 and Corollary 34 in [19] for asynchronous Q-learning and polynomial learning rates, and we obtain that, with probability at least $1-\delta$, $\|\check{Q}_{N_{epi}} - \check{Q}\|_\infty \leq \epsilon$, given that N_{epi} is of order (11).

Step 2. For every $(\check{\mu}, \tilde{a}) \in \check{\mathbb{S}} \times \tilde{\mathbb{A}}$,

$$\check{Q}(\check{\mu}, \tilde{a}) \approx Q(\check{\mu}, \tilde{a}).$$

This amounts to say that the projection on $\check{\mathbb{S}}$ realized at each step does not perturb too much the value function. Let us start by noting that, for every $\check{\mu} \in \check{\mathbb{S}}$ and $\tilde{a} \in \tilde{\mathbb{A}}$,

$$\begin{aligned}
&\left|\check{Q}(\check{\mu}, \tilde{a}) - Q(\check{\mu}, \tilde{a})\right| \\
&= \gamma \left| \mathbb{E}\left[\max_{\tilde{a}'} \check{Q}(\check{\Phi}^{\tilde{a}}(\check{\mu}), \tilde{a}') - \max_{\tilde{a}'} Q(\Phi^{\tilde{a}}(\check{\mu}), \tilde{a}')\right] \right| \\
&\leq \gamma \mathbb{E}\left|\check{V}(\check{\Phi}^{\tilde{a}}(\check{\mu})) - V(\Phi^{\tilde{a}}(\check{\mu}))\right| \\
&\leq \gamma \mathbb{E}\left|\check{V}(\check{\Phi}^{\tilde{a}}(\check{\mu})) - V(\check{\Phi}^{\tilde{a}}(\check{\mu}))\right| \\
&\qquad + \gamma \mathbb{E}\left|V(\check{\Phi}^{\tilde{a}}(\check{\mu})) - V(\Phi^{\tilde{a}}(\check{\mu}))\right| \\
&\leq \gamma \|\check{Q} - Q\|_\infty + \gamma L_V \mathbb{E}\left[d_{\mathbb{S}}\left(\check{\Phi}^{\tilde{a}}(\check{\mu}), \Phi^{\tilde{a}}(\check{\mu})\right)\right],
\end{aligned}$$

where the last inequality holds by Lipschitz continuity of V, see Assumption **(A2)**, and because for every $\check{\mu}' \in \check{\mathbb{S}}$,

$$\begin{aligned}
&\left|\check{V}(\check{\mu}') - V(\check{\mu}')\right| \\
&= \left|\sup_{\tilde{a}'} \check{Q}\left(\check{\mu}', \tilde{a}'\right) - \sup_{\tilde{a}'} Q\left(\check{\mu}', \tilde{a}'\right)\right| \\
&\leq \sup_{\check{\mu}'', \tilde{a}'} \left|\check{Q}\left(\check{\mu}'', \tilde{a}'\right) - Q\left(\check{\mu}'', \tilde{a}'\right)\right| = \|\check{Q} - Q\|_\infty.
\end{aligned}$$

To conclude, we use Assumptions **(A3)** and **(A1)**, and we obtain:

$$\begin{aligned}
&\mathbb{E}_{e^0}\left[d_{\mathbb{S}}\left(\check{\Phi}^{\tilde{a},e^0}(\check{\mu}), \Phi^{\tilde{a},e^0}(\mu)\right)\right] \\
&\leq \epsilon_S + \mathbb{E}_{e^0}\left[d_{\mathbb{S}}\left(\Phi^{\tilde{a},e^0}(\check{\mu}), \Phi^{\tilde{a},e^0}(\mu)\right)\right] \\
&\leq \epsilon_S + L_\Phi d_{\mathbb{S}}\left(\check{\mu}, \mu\right) \\
&\leq (1 + L_\Phi)\epsilon_S.
\end{aligned}$$

Combining the above bounds yields

$$\|\check{Q} - Q\|_\infty \leq \frac{\gamma}{1-\gamma} L_V (1 + L_\Phi)\epsilon_S.$$

Step 3. For every $(\mu, \tilde{a}) \in \mathbb{S} \times \tilde{\mathbb{A}}$,

$$Q(\mathrm{Proj}_{\check{\mathbb{S}}}(\mu), \tilde{a}) \approx Q(\mu, \tilde{a}).$$

Indeed, for every $\mu \in \check{\mathbb{S}}$ and $\tilde{a} \in \tilde{\mathbb{A}}$, letting $\check{\mu} = \mathrm{Proj}_{\check{\mathbb{S}}}(\mu)$ to alleviate the notation, we have

$$\begin{aligned}
&|Q(\check{\mu}, \tilde{a}) - Q(\mu, \tilde{a})| \\
&\leq \left|\tilde{f}(\check{\mu}, \tilde{a}) - \tilde{f}(\mu, \tilde{a})\right| + \\
&\quad + \gamma \mathbb{E}\left|\max_{\tilde{a}'} Q(\Phi^{\tilde{a}}(\check{\mu}), \tilde{a}') - \max_{\tilde{a}'} Q(\Phi^{\tilde{a}}(\mu), \tilde{a}')\right| \\
&\leq L_{\tilde{f}} d_{\mathbb{S}}(\check{\mu}, \mu) + \gamma \mathbb{E}\left|V(\Phi^{\tilde{a}}(\check{\mu})) - V(\Phi^{\tilde{a}}(\mu))\right| \\
&\leq \left(L_{\tilde{f}} + \gamma L_V L_\Phi\right) d_{\mathbb{S}}(\check{\mu}, \mu) \\
&\leq \left(L_{\tilde{f}} + \gamma L_V L_\Phi\right) \epsilon_S,
\end{aligned}$$

where we used the Lipschitz continuity of $\tilde{f}, V, \Phi$ and the assumption on $\check{\mathbb{S}}$, see Assumptions **(A1)**, **(A2)** and **(A3)**. □

Proof of Corollary 3. We use Proposition 4 of [22], which states that $\mathrm{softmax}_\tau$ is τ-Lipschitz and Lemma 7 [24], which states that for $(x_i)_{i=1,\dots,n}$,

$$\|\mathrm{softmax}_\tau(x) - \mathrm{argmaxe}(x)\|_2 \leq 2ne^{-\tau\delta},$$

where $\delta = \max(x) - \max_{x_i < \max(x)} x_i$, and $\delta = \infty$ if all x_i are equal. We can apply this latter result to $Q(\check{\mu}, \cdot)$ thanks to our Assumption **(B)** on the action gap, with $n = |\tilde{\mathbb{A}}|$ and

$\delta = K_A$. Combining this with the result of Theorem 2, we have, for every $\check{\mu}$,

$$\begin{aligned}
&\|\text{softmax}_\tau\, \check{Q}(\check{\mu},\cdot) - \text{argmaxe}\, Q(\check{\mu},\cdot)\|_2 \\
&\le \|\text{softmax}_\tau\, \check{Q}(\check{\mu},\cdot) - \text{softmax}_\tau Q(\check{\mu},\cdot)\|_2 \\
&\qquad + \|\text{softmax}_\tau Q(\check{\mu},\cdot) - \text{argmaxe}\, Q(\check{\mu},\cdot)\|_2 \\
&\le \tau \|\check{Q}(\check{\mu},\cdot) - Q(\check{\mu},\cdot)\|_{\ell^\infty(\tilde{\mathbb{A}})} + 2|\tilde{\mathbb{A}}| e^{-\tau K_A} \\
&\le \tau\epsilon' + 2|\tilde{\mathbb{A}}| e^{-\tau K_A}.
\end{aligned}$$

□

D Deep Deterministic Policy Gradient Algorithm

In this section, we recall for the sake of completeness the Deep Deterministic Policy Gradient (DDPG) proposed in [33], adapted to our mean field control problem. See also [18] for how the same algorithm can be used for MFG. In the latter case, it is used to compute the (approximate) best response of a single infinitesimal player instead of the optimal control for the whole population as in MFC problems.

Algorithm 2: DDPG for MFC

Data: A number of episodes N_{epi}; a length T for each episode; a minibatch size N_{batch}; a learning rate τ.

Result: A control $a \in \mathbb{A}$.

begin

Randomly initialize critic network Q_{θ^Q} and actor network π_{θ^π} with parameters θ^Q and θ^π respectively

Randomly initialize target networks $Q'_{\theta^{Q'}}$ and network $\pi'_{\theta^{\pi'}}$ with $\theta^{Q'} \leftarrow \theta^Q$ and $\theta^{\pi'} \leftarrow \theta^\pi$

for *episode* $k = 0, 1, \dots N_{epi} - 1$ **do**

Initialize M_0

Initialize replay buffer R

for $t = 0, 1, \dots T-1$ **do**

Select an action $a_t = \pi_{\theta^\pi}(M_t) + \mathcal{N}_t \in \mathbb{R}^{N_p}$

Execute a_t, observe reward $\tilde{f}_t$ and M_{t+1}

Store transition $(M_t, a_t, \tilde{f}_t, M_{t+1})$ in R

Sample a random minibatch of N_{batch} transitions $(M_i, a_i, \tilde{f}_i, M_{i+1})$ from R

Set $y_i = \tilde{f}_i + \gamma Q'_{\theta^{Q'}}(M_{i+1}, \pi'_{\theta^{\pi'}}(M_{i+1}))$, for $i = 1 \dots, N_{batch}$

Update the critic by minimizing the loss: $L(\theta^Q) = \frac{1}{N_{batch}} \sum_i (y_i - Q_{\theta^Q}(x_i, a_i))^2$

Update the actor policy using the sampled policy gradient $\nabla_{\theta^\pi} J$ of

$$J(\theta^\pi) = \frac{1}{N_{batch}} \sum_i \nabla_a Q_{\theta^Q}(M_i, \pi_{\theta^\pi}(M_i))$$

Update target networks: $\theta^{Q'} \leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'}$ and $\theta^{\pi'} \leftarrow \tau\theta^\pi + (1-\tau)\theta^{\pi'}$

return π_{θ^π}