Journal of Machine Learning **ISSN**: 2790-2048(e), 2790-203X(p)

# Deep Reinforcement Learning for Infinite Horizon Mean Field Problems in Continuous Spaces

Andrea Angiuli * [1], Jean-Pierre Fouque † [2], Ruimeng Hu ‡ [2,3], and Alan Raydan § [3]

[1]Prime Machine Learning Team, Amazon, SEA83, Seattle, WA, 98109, USA.
[2]Department of Statistics and Applied Probability, University of California, Santa Barbara, CA 93106-3110, USA.
[3]Department of Mathematics, University of California, Santa Barbara, CA 93106-3080, USA.

**Abstract.** We present the development and analysis of a reinforcement learning algorithm designed to solve continuous-space mean field game (MFG) and mean field control (MFC) problems in a unified manner. The proposed approach pairs the actor-critic (AC) paradigm with a representation of the mean field distribution via a parameterized score function, which can be efficiently updated in an online fashion, and uses Langevin dynamics to obtain samples from the resulting distribution. The AC agent and the score function are updated iteratively to converge, either to the MFG equilibrium or the MFC optimum for a given mean field problem, depending on the choice of learning rates. A straightforward modification of the algorithm allows us to solve mixed mean field control games. The performance of our algorithm is evaluated using linear-quadratic benchmarks in the asymptotic infinite horizon framework.

**Keywords:**
Actor-critic, Linear-quadratic control, Mean field game, Mean field control, Mixed mean field control game, Score matching, Reinforcement learning, Timescales.

**Article Info.:**
Volume: 4
Number: 1
Pages: 11 - 47
Date: March/2025
doi.org/10.4208/jml.230919

**Article History:**
Received: 19/09/2023
Accepted: 08/11/2024

**Communicated by:**
Jiequn Han

## 1 Introduction

Mean field games and mean field control – collectively dubbed mean field problems – are mathematical frameworks used to model and analyze the behavior and optimization of large-scale, interacting agents in settings with varying degrees of cooperation. Since the early 2000s, with the seminal works [21, 28], MFGs have been used to study the equilibrium strategies of competitive agents in a large population, accounting for the aggregate behavior of the other agents. Alternately, MFC, which is equivalent to optimal control of McKean-Vlasov SDEs [31, 32], focuses on optimizing the behavior of a central decision-maker controlling the population in a cooperative fashion. Cast in the language

*aangiuli@amazon.com
†fouque@pstat.ucsb.edu
‡Corresponding author. rhu@ucsb.edu
§alanraydan@ucsb.edu

of stochastic optimal control, both frameworks center on finding an optimal control $\alpha_t$, which minimizes a cost functional objective $J(\alpha)$ subject to given state dynamics in the form of a stochastic differential equation. What distinguishes mean field problems from classical optimal control is the presence of the mean field distribution $\mu_t$, which may influence both the cost functional and the state dynamics. The mean field is characterized by a flow of probability measures that emulates the effect of a large number of participants whose individual states are negligible but whose influence appears in the aggregate. In this setting, the state process $X_t$ models a representative player from the crowd in the sense that the mean field should ultimately be the law of the state process: $\mu_t = \mathcal{L}(X_t)$. The distinction between MFG and MFC, a competitive game versus a cooperative governance, is made rigorous by precisely how we enforce the relationship between $\mu_t$ and $X_t$. We will address the details of the MFG/MFC dichotomy in greater depth in Section 2.

MFG and MFC theories have been instrumental in understanding and solving problems in a wide range of disciplines, such as economics, social sciences, biology, and engineering. In finance, mean field problems have been applied to model and analyze the behavior of investors and markets. For instance, MFG can be used to model the trading strategies of individual investors in a financial market, taking into account the impact of the overall market dynamics. Similarly, MFC can help optimize the management of large portfolios, where the central decision-maker seeks to maximize returns while considering the average behavior of other investors. For in-depth examples of mean field problems in finance, we refer the reader to [10–12].

Although traditional numerical methods for solving MFG and MFC problems have proceeded along two avenues, solving a pair of coupled partial differential equations (PDE) [13] or a forward-backward system of stochastic differential equations (FBSDE) [6], there has been growing interest in solving mean field problems in a model-free way [3, 4, 14, 19, 29, 34]. With this in mind, we turn to reinforcement learning (RL), an area of machine learning that trains an agent to make optimal decisions through interactions with a "black box" environment. RL can be employed to solve complex problems, such as those found in finance, traffic control, and energy management, in a model-free manner. A key feature of RL is its ability to learn from trial-and-error experiences, refining decision-making policies to maximize cumulative rewards. Temporal difference (TD) methods [37] are a class of RL algorithms that are particularly well-suited for this purpose. They estimate value functions by updating estimates based on differences between successive time steps, combining the benefits of both dynamic programming and Monte Carlo approaches for efficient learning without requiring a complete model of the environment. For a comprehensive overview of the foundations and numerous families of RL strategies, consult [38]. Actor-critic (AC) algorithms – the modern incarnations of which were introduced in [17] – are a popular subclass of TD methods where separate components, the actor and the critic, are used to update estimates of both a policy and a value function. The actor is responsible for selecting actions based on the current policy, while the critic evaluates the chosen actions and provides feedback to update the policy. By combining the strengths of both policy- and value-based approaches, AC algorithms achieve more stable and efficient learning.

The mean field term itself poses an interesting problem regarding how to numerically store and update a probability measure on a continuous space in an efficient manner. Some authors have chosen to discretize the continuous space, which leads to a vectorized representation as in [3,4], while others have looked towards deep learning and deep generative models [34]. We extend the latter avenue by considering a method of distributional learning known as score-matching [22] in which a probability distribution is represented by the gradient of its log density, i.e. its Stein score function. A parametric representation of the score function is updated using samples from the underlying distribution and allows us to compute new samples from the distribution using a discrete version of Langevin dynamics. We explain how to modify the score-matching procedure for our online regime in Section 4.1.

Building off of the work of [3,4], in which the authors adapt tabular Q-learning [43] to solve discrete-space MFG and MFC problems, the main contributions of this papers are:

1. We address continuous-state and -action space mean field problems in a unified manner by developing an AC algorithm inspired by advantage actor-critic [33]. This means that, for a given mean field problem, we use the same algorithm to converge to the MFG and MFC solutions simply by adjusting the relative learning rates of the parametric representations of the actor, critic, and mean field distribution. Compared to Q-learning, our proposed algorithm excels in handling continuous or large finite state- and action-spaces, thanks to our neural network implementation. Additionally, it has the advantage of directly learning a policy.

2. To effectively represent the mean field distribution in continuous space, we employ the score function, defined as the gradient of its log density. We parameterize the score function using neural networks and update it via score-matching techniques [22]. Our method then combines the mean field with the actor-critic paradigm by concurrently learning the score function of the mean field distribution alongside the optimal control, which we derive from the policy learned by the actor.

Note that our approach relies on a randomized policy for the actor while another direction of research utilizes entropy-regularization of the value function as in [16, 20, 42]. See also the comment in Section 5.5 describing the relationship between the two approaches. Other papers have concentrated on reinforcement learning for McKean-Vlasov control problems in continuous-time such as in [44], but our approach treats both MFG and MFC in the same algorithm. We also refer to [18] for learning MFC problems when the policy may depend additionally on the population distribution.

The rest of the paper is organized as follows. In Sections 2 and 3, we review the infinite horizon formulation for asymptotic mean field problems and recall the relevant background from RL, respectively. In Section 4 we modify the Markov decision process setting of RL to apply to mean field problems and present our central algorithm. Numerical results and comparisons with benchmark solutions are presented in Section 5. As a concluding application, we alter the algorithm in Section 6 to apply to mean field control games (MFCG), an extension of mean field problems combining both MFG and MFC to model multiple large homogeneous populations where interactions occur not only within each group, but also between groups.

# 2 Asymptotic infinite horizon mean field problems

In this section, we introduce the framework of mean field games and mean field control in the continuous-time infinite horizon setting. We further emphasize the mathematical distinction between the two classes of mean field problems, highlighting that they yield distinct solutions despite the apparent similarities in their formulation.

In both cases, the mathematical setting is a filtered probability space

$$\Omega, \mathcal{F}, \mathbb{F} = (\mathcal{F}_t)_{t\geq 0}, \mathbb{P},$$

satisfying the usual conditions which support an $m$-dimensional Brownian motion $(W_t)_{t\geq 0}$. The measurable function $f : \mathbb{R}^d \times \mathcal{P}_2(\mathbb{R}^d) \times \mathbb{R}^k \to \mathbb{R}$ is known as the running cost, where $\mathcal{P}(\mathbb{R}^d)$ denotes space of probability measures on $\mathbb{R}^d$ and $\mathcal{P}_s(\mathbb{R}^d)$ is the subspace of $\mathcal{P}(\mathbb{R}^d)$ with finite $s$-th-moments. The constant $\beta > 0$ is a discount factor, which measures the relative importance or value of future rewards or costs compared to immediate ones. For the state dynamics, we have drift $b : \mathbb{R}^d \times \mathcal{P}_2(\mathbb{R}^d) \times \mathbb{R}^k \to \mathbb{R}^d$ and volatility $\sigma : \mathbb{R}^d \times \mathcal{P}_2(\mathbb{R}^d) \times \mathbb{R}^k \to \mathbb{R}^{d\times m}$.

We will focus on the asymptotic formulation of the infinite horizon mean field problem. In this formulation, we seek a control in the feedback form that depends solely on the state process of the representative player $X : [0,\infty) \times \Omega \to \mathbb{R}^d$ with no explicit time dependency. In other words, the control function is of the form $\alpha : \mathbb{R}^d \to \mathbb{R}^k$, and the trajectory of the control will be given by $\alpha_t = \alpha(X_t)$. This choice allows us to frame the problem more naturally in terms of the Markov decision process setting of reinforcement learning (see Section 3), which is naturally formulated with time-independent policies.

It was shown in [3] that the asymptotic formulation of the MF problem is the limit of the traditional time-dependent problem as $t \to \infty$ in the following sense: the equilibrium (respectively optimal) control for the time-dependent MFG (respectively MFC) problem converges to the equilibrium (respectively optimal) control for the asymptotic problem as $t \to \infty$.

## 2.1 Mean field games

The solution of a mean field game, known as a mean field game equilibrium, is a control-mean field pair

$$(\hat{\alpha}, \hat{\mu}) \in \mathbb{A} \times \mathcal{P}(\mathbb{R}^d),$$

where $\mathbb{A}$ is the set of measurable functions $\alpha : \mathbb{R}^d \to \mathbb{R}^k$, satisfying the following conditions:

1. $\hat{\alpha}$ solves the stochastic optimal control problem

$$\inf_{\alpha\in\mathbb{A}} J_{\hat{\mu}}(\alpha) = \inf_{\alpha\in\mathbb{A}} \mathbb{E}\left[\int_0^\infty e^{-\beta t} f\big(X_t^{\alpha,\hat{\mu}}, \hat{\mu}, \alpha\big(X_t^{\alpha,\hat{\mu}}\big)\big)\,\mathrm{d}t\right], \quad \beta > 0 \tag{2.1}$$

subject to

$$\mathrm{d}X_t^{\alpha,\hat{\mu}} = b\big(X_t^{\alpha,\hat{\mu}}, \hat{\mu}, \alpha\big(X_t^{\alpha,\hat{\mu}}\big)\big)\,\mathrm{d}t + \sigma\big(X_t^{\alpha,\hat{\mu}}, \hat{\mu}, \alpha\big(X_t^{\alpha,\hat{\mu}}\big)\big)\,\mathrm{d}W_t, \quad X_0^{\alpha,\hat{\mu}} = \xi. \tag{2.2}$$

DOI https://doi.org/10.4208/jml.230919 | Generated on 2025-04-09 07:43:42

2. $\hat{\mu} = \lim_{t\to\infty} \mathcal{L}(X_t^{\hat{\alpha},\hat{\mu}})$, where $\mathcal{L}(X_t^{\hat{\alpha},\hat{\mu}})$ refers to the law of $X_t^{\hat{\alpha},\hat{\mu}}$.

This problem models a scenario in which an infinitesimal player seeks to integrate into a crowd of players already in the asymptotic regime as time tends toward infinity. The resulting stationary distribution represents a Nash equilibrium under the premise that any new player entering the crowd sees no benefit in diverging from this established asymptotic behavior.

## 2.2 Mean field control

A mean field control problem solution is a control $\alpha^* \in \mathbb{A}$ which satisfies an optimal control problem with McKean-Vlasov dynamics

$$\inf_{\alpha\in\mathbb{A}} J(\alpha) = \inf_{\alpha\in\mathbb{A}} \mathbb{E}\left[\int_0^\infty e^{-\beta t} f\big(X_t^\alpha, \mu^\alpha, \alpha\big(X_t^\alpha\big)\big)\,\mathrm{d}t\right] \tag{2.3}$$

subject to

$$\mathrm{d}X_t^\alpha = b\big(X_t^\alpha, \mu^\alpha, \alpha\big(X_t^\alpha\big)\big)\,\mathrm{d}t + \sigma\big(X_t^\alpha, \mu^\alpha, \alpha\big(X_t^\alpha\big)\big)\,\mathrm{d}W_t, \quad X_0^\alpha = \xi, \tag{2.4}$$

using the notation $\mu^\alpha = \lim_{t\to\infty} \mathcal{L}(X_t^\alpha)$. We will also adopt the notation $\mu^*$ to refer to $\mu^{\alpha^*}$ – the limiting distribution for the mean field distribution under the optimal control. In this alternate scenario, we are considering the perspective of a central organizer. Their objective is to identify the control which yields the best possible stationary distribution, ensuring that the societal costs incurred are the lowest possible when a new individual integrates into the group.

Although the initial distribution $\xi$ is specified in both cases, under suitable ergodicity assumptions, the optimal controls $\hat{\alpha}$ and $\alpha^*$ are independent of this initial distribution. For an in-depth treatment of infinite horizon mean field problems, with explicit solutions for the case of linear dynamics and quadratic cost, refer to [30].

## 2.3 Mean field game/control distinction

We summarize the crucial mathematical distinction between MFG and MFC. In the former, one must solve an optimal control problem depending on an arbitrary distribution $\mu$ and then recover the mean field $\hat{\mu}$, which yields the law of the optimal limiting state trajectory. If we consider the map

$$\Phi(\mu) = \lim_{t\to\infty} \mathcal{L}\big(X_t^{\tilde{\alpha},\mu}\big),$$

where $\tilde{\alpha} = \arg\min J_\mu(\alpha)$, then the MFG equilibrium arises as a fixed point of $\Phi$ in the sense that

$$\hat{\mu} = \Phi(\hat{\mu}).$$

In the latter case, the mean field is explicitly the law of the state process throughout the optimization and should be thought of as a pure control problem in which the law of the state process influences the state dynamics. Note that in the MFC case, the distribution $\mu^\alpha$

"moves" with the choice of control $\alpha$, while in the MFG case, it is "frozen" during the optimization step and then a fixed point problem is solved. These interpretations play a key role in guiding the development of this paper's central algorithm, detailed in Section 4.

Crucially, we conclude this section by noting that, in general,

$$(\hat{\alpha}, \hat{\mu}) \neq (\alpha^*, \mu^*) \tag{2.5}$$

for the same choice of running cost, discount factor, and state dynamics. Indeed, we will encounter examples of mean field problems with differing solutions when we test our algorithm against benchmark problems in Section 5.

## 3 Reinforcement learning and actor-critic algorithms

Reinforcement learning is a family of machine learning strategies aimed at choosing the sequence of actions which maximizes the long-term aggregate reward from an environment in a model-free way, i.e. assuming no explicit knowledge of the state dynamics or the reward function. Intuitively, one should imagine a black box environment in which an autonomous agent makes decisions in discrete time and receives immediate feedback in the form of a scalar reward signal.

At stage $n$, the agent is in a state $X_{t_n}$ from a given set of states $\mathcal{X}$ and selects an action $A_{t_n}$ from a set of actions $\mathcal{A}$. The environment responds by placing the agent in a new state $X_{t_{n+1}}$ and bestowing it with an immediate reward $r_{t_{n+1}} \in \mathbb{R}$. The agent continues choosing actions, encountering new states, and obtaining rewards in an attempt to maximize the total expected discounted return

$$\mathbb{E}\left[\sum_{n=0}^{\infty} \gamma^n r_{t_{n+1}}\right], \tag{3.1}$$

where $\gamma \in (0,1)$ is a discount factor specifying the degree to which the agent prioritizes immediate reward over long-term returns. The case in which we seek to minimize cost instead of maximize reward as in most financial applications, can be recast in the above setting by taking $r_{t_{n+1}} = -c_{t_{n+1}}$ where $c_{t_{n+1}}$ is the immediate cost incurred at the $n$-th time-step. The expectation in (3.1) refers to the stochastic transition from $X_{t_n}$ to $X_{t_{n+1}}$ and, eventually, to the randomness in the choice of $A_{t_n}$. When the new state $X_{t_{n+1}}$ and immediate reward $r_{t_{n+1}}$ only depend on the preceding state $X_{t_n}$ and action $A_{t_n}$, the above formulation is known as a Markov decision process (MDP).

The agent chooses its actions according to a policy $\pi : \mathcal{X} \to \mathcal{P}(\mathcal{A})$, which defines the probability that a certain action should be taken in a given state. The goal of the agent is then to find an optimal policy $\pi^*$ satisfying

$$\pi^* \in \arg\max_{\pi} \mathbb{E}_{\pi}\left[\sum_{n=0}^{\infty} \gamma^n r_{t_{n+1}}\right].$$

As the reward $r_{t_{n+1}} = r(X_{t_n}, A_{t_n})$ is a function of the current state and current action, the value to be maximized does indeed depend on the policy $\pi$.

For a given policy $\pi$, two quantities of interest in RL are the so-called state-value function $v_\pi : \mathcal{X} \to \mathbb{R}$ and the action-value function $q_\pi : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$ given by

$$v_\pi(x) = \mathbb{E}_\pi\left[\sum_{n=0}^{\infty} \gamma^n r(X_{t_n}, A_{t_n}) \mid X_{t_0} = x\right], \tag{3.2}$$

$$q_\pi(x,a) = \mathbb{E}_\pi\left[\sum_{n=0}^{\infty} \gamma^n r(X_{t_n}, A_{t_n}) \mid X_{t_0} = x, A_{t_0} = a\right]. \tag{3.3}$$

The state-value function defines the expected return obtained from beginning in an initial state $x$ and following the policy $\pi$ from the get-go, whereas the action-value function defines the expected return starting from $x$, taking an initial action $a$, and then proceeding according to $\pi$ after the first step. Moreover, $v_\pi$ and $q_\pi$ are related to each other via the following:

$$v_\pi(x) = \sum_{a \in \mathcal{A}} \pi(a \mid x) q_\pi(x,a).$$

The action-value function is integral to many RL algorithms since, assuming that the action-value function $q_*$ corresponding to an optimal policy is known, one can derive an optimal policy by taking the uniform distribution over the actions that maximize $q_*$

$$\pi^*(\cdot \mid x) = \text{unif}\left(\arg\max_{a \in \mathcal{A}} q_*(x,a)\right).$$

However, since this paper makes far more use of $v_\pi$ than $q_\pi$, we will henceforth refer to the former simply as the "value function" and the latter as the "Q-function" as is common in the literature. When referring to both $v_\pi$ and $q_\pi$, we may refer to them jointly as the value functions associated with $\pi$.

For our continuous-space setting, we randomize the policy according to a Gaussian distribution as explained in Section 3.2.

## 3.1 Temporal difference methods

In the search for an optimal policy, one often begins with an arbitrary policy, which is improved as the RL agent gains experience in the environment. A key factor in improving a policy is an accurate estimate of the associated value function since this allows us to quantify precisely how much better one policy is over another. The value function satisfies the celebrated Bellman equation, which relates the value of the current state to that of the successor state

$$v_\pi(x) = \mathbb{E}_\pi\big[r_{t_{n+1}} + \gamma v_\pi(X_{t_{n+1}}) \mid X_{t_n} = x\big]. \tag{3.4}$$

Since the transition from $X_{t_n}$ to $X_{t_{n+1}}$ is Markovian, (3.4) holds for all $n \geq 0$, not just the initial state. Importantly, the Bellman equation uniquely defines the value function for a given $\pi$, a fact which underlies all algorithms under the umbrella of dynamic programming. Solving the Bellman equation for $v_\pi$ is impossible without knowing the reward function and state transition dynamics, so an alternative strategy is needed for our model-free scenario.

Temporal difference methods center around iteratively updating an approximation $V$ to $v_\pi$ in order to drive to zero the TD error $\delta_n$,

$$\delta_n := r_{t_{n+1}} + \gamma V(X_{t_{n+1}}) - V(X_{t_n}) \tag{3.5}$$

at each timestep. TD methods use estimates of the value at future times to update the value at the current time, a strategy known as "bootstrapping". More importantly, the TD methods we reference here require only the immediate transition sequence $\{X_{t_n}, r_{t_{n+1}}, X_{t_{n+1}}\}$ and no information regarding the MDP model.

## 3.2 Actor-critic algorithms

Actor-critic algorithms form a subset of TD methods in which explicit representations of the policy (the actor) and the value function (the critic) are stored. Often, the representation of the policy is a parametric family of density functions in which the parameters are the outputs of another parametric family of functions, e.g. linear functions, polynomials, and neural networks. In the implementation discussed in Section 4, our actor is represented by a feedforward neural network which outputs the mean and standard deviation of a normal distribution. The action $A_{t_n}$ is then sampled according to this density. The benefit of a stochastic policy such as this is that it allows for more exploration of the environment so that the agent does not myopically converge to a suboptimal policy.

Since the value function simply outputs a scalar, it may be represented by any sufficiently rich family of real-valued functions. Let

$$\Pi_\psi \approx \pi, \quad V_\theta \approx v$$

be the parametric approximations of $\pi$ and $v$, both differentiable in their respective parameters $\psi$ and $\theta$. The goal for AC algorithms is to converge to an optimal policy by iteratively updating the actor to maximize the value function and updating the critic to satisfy the Bellman equation. For the critic, this suggests minimizing the following loss function at the $n$-th step:

$$L_V^{(n)}(\theta) := \left(r_{n+1} + \gamma V_\theta(X_{t_{n+1}}) - V_\theta(X_{t_n})\right)^2 =: \delta_n^2. \tag{3.6}$$

Note that the terms inside the square are precisely the TD error $\delta_n$ from (3.5).

The traditional gradient TD update treats the term $y_{t_{n+1}} = r_{n+1} + \gamma V_\theta(X_{t_{n+1}})$ – known as the TD target – as a constant and only considers the term $-V_\theta(X_{t_n})$ as a function of $\theta$. This yields faster convergence from gradient descent as opposed to treating both terms as variables in $\theta$ [40]. With this in mind, the gradient of $L_V$ is then

$$\nabla_\theta L_V^{(n)}(\theta) = -2\delta_n \nabla_\theta V_\theta(X_{t_n}), \tag{3.7}$$

meaning that, with some learning rate $\rho_V > 0$, we can update the parameters of the critic iteratively using gradient descent

$$\theta' = \theta + 2\rho_V \delta_n \nabla_\theta V_\theta(X_{t_n}). \tag{3.8}$$

Updating the actor, on the other hand, is not so obvious since updating $\psi$ to maximize $V_\theta$ requires somehow computing $\nabla_\psi V_\theta$. Since the connection between $\Pi_\psi$ and $V_\theta$ is

not explicit, it is not clear how to compute this gradient a priori. Thankfully, the desired relation comes in the form of the policy gradient theorem [39], which relates a parameterized policy and its value function via the following:

$$\nabla_\psi v_{\Pi_\psi}(x) \propto \mathbb{E}_{\Pi_\psi}\big[q_{\Pi_\psi}(X_{t_n}, A_{t_n})\nabla_\psi \log \Pi_\psi(A_{t_n} \mid X_{t_n})\big] \tag{3.9}$$

for any initial state $x \in \mathcal{X}$, where $v_{\Pi_\psi}$ and $q_{\Pi_\psi}$ are the true value functions associated with the parameterized policy $\Pi_\psi$.

As a result of the Bellman equation, we have the identity

$$q_\pi(x, a) = \mathbb{E}_\pi\big[r_{t_{n+1}} + \gamma v_\pi(X_{t_{n+1}})\big],$$

given that $r_{t_{n+1}}$ was the reward obtained by taking the action $a$ in the state $x$. Moreover, adding an arbitrary "baseline" value $\lambda$ to $q_{\Pi_\psi}(X_{t_n}, A_{t_n})$ does not alter the gradient in (3.9) as long as $\lambda$ does not depend on the action $A_{t_n}$. A common baseline value demonstrated to reduce variance and speed up convergence is $\lambda = -v_{\Pi_\psi}(X_{t_n})$ [40]. With this in mind, we can replace $q_{\Pi_\psi}(X_{t_n}, A_{t_n})$ in (3.9) with the TD error

$$\delta_n = r_{n+1} + \gamma v_{\Pi_\psi}(X_{t_{n+1}}) - v_{\Pi_\psi}(X_{t_n}),$$

which allows us to reuse $\delta_n$ from its role in updating the critic. As a whole, this suggests the following loss function for the actor:

$$L_\Pi^{(n)}(\psi) := -\delta_n \log \Pi_\psi(A_{t_n} \mid X_{t_n}). \tag{3.10}$$

For a learning rate $\rho_\Pi > 0$, the gradient descent step would then be

$$\psi' = \psi + \rho_\Pi \delta_n \nabla_\psi \log \Pi_\psi(A_{t_n} \mid X_{t_n}). \tag{3.11}$$

In practical applications, updating the actor and critic in the above fashion at each step generally yields convergence to an optimal policy and value function, respectively. While convergence has been proven in the case of linearly parameterized actor and critic [27], convergence in the general case is still an open problem.

### 3.2.1 Relative learning rates for actor and critic

Since the gradient descent learning rates play a crucial role in the development of our fundamental algorithm presented in Section 4, we briefly comment on the choice of learning rates in AC algorithms.

The AC framework alternates between two key steps: Refining the critic to accurately approximate the value function associated with the actor's policy – known as policy evaluation – and updating the actor to maximize the value returned by the critic – known as policy improvement. As the policy improvement step relies on the policy gradient theorem (3.9), a sufficiently precise critic is required for its success. Hence, the learning rates for the actor and critic are traditionally chosen such that

$$\rho_\Pi < \rho_V.$$

This constraint prompts the critic to learn at a quicker pace compared to the actor, thereby ensuring that the value function from the policy evaluation phase closely aligns with the policy's true value function.

# 4 Unified mean field actor-critic algorithm for infinite horizon

In this section, we introduce a novel infinite horizon mean field actor-critic (IH-MF-AC) algorithm for solving both MFG and MFC problems in continuous-time and continuous-space. Although there have been significant strides in recasting the MDP framework for continuous-time using the Hamiltonian of the associated continuous-time control problem as an analog of the Q-function [23–25, 42], we instead take the classical approach of first discretizing the continuous-time problem and then applying the MDP strategies discussed in Section 3. As our focus is aimed at identifying the stationary solution of the infinite horizon mean field problems, discretizing time does not meaningfully depart from the original continuous-time problem presented in Section 2. While in our ongoing work where we tackle the finite horizon regime, the time-discretization must be treated with more care since the mean field becomes a flow of probability distributions parameterized by time, and the optimal control also becomes time-dependent in this context. In the sequel, we will first recast the mean field setting from Section 2 as a discrete MDP parameterized by the distribution $\mu$ and then lay out the general procedure of the algorithm before addressing the continuous-space representation of $\mu$ via score functions in Section 4.1. Section 4.2 addresses the justification for alternating between the MFG and MFC solutions using the actor, critic, and mean field learning rates.

To begin, we fix a small step size $\Delta t > 0$ and consider the resulting time discretization $(t_0, t_1, t_2, \ldots)$ where $t_n = n\Delta t$. We then rewrite the cost objectives in (2.1) and (2.3) as the Riemann sum

$$\mathbb{E}\left[\sum_{n=0}^{\infty} e^{-\beta t_n} f(X_{t_n}, \mu, A_{t_n})\Delta t\right], \tag{4.1}$$

and the state dynamics in (2.2) and (2.4) as the Euler-Maruyama approximation

$$X_{t_{n+1}} = X_{t_n} + b(X_{t_n}, \mu, A_{t_n})\Delta t + \sigma(X_{t_n}, \mu, A_{t_n})\Delta W_n, \quad \Delta W_n \sim \mathcal{N}(0, \Delta t). \tag{4.2}$$

This reformulation is directly in correspondence with the MDP setting presented in Section 3 – albeit, parameterized by $\mu$. Observe that

$$r_{t_{n+1}} = -f(X_{t_n}, \mu, A_{t_n})\Delta t, \quad \gamma = e^{-\beta\Delta t},$$

and the state transition dynamics are given by (4.2). Note that the model and approximation approach used to derive the reward and next state dynamics are our own choice for simulation purposes but are not prescriptive. The agent sees the environment as a black box and has no knowledge of the details of the problem.

In the style of the AC method described in Section 3.2, our algorithm maintains and updates a policy $\Pi_\psi$ and a value function $V_\theta$ which are meant as stand-ins for the control $\hat{\alpha}$ (respectively $\alpha^*$) of the MFG (respectively MFC) and the cost functional $J$, respectively. Both are taken to be feedforward neural networks. The third component is the mean field distribution $\mu$, which is updated simultaneously with the actor and critic at each timestep to approximate the law of $X_t$. The procedure at the $n$-th step is as follows: the agent is in the state $X_{t_n}$ as a result of the dynamics in (4.2) where $\mu$ is replaced with the current estimate of the mean field $\mu_{n-1}$. The value of $X_{t_n}$ is then used to update the mean field,

yielding a new estimate $\mu_n$ (see section Section 4.1 for details). Using the actor's policy, the agent samples an action $A_{t_n} \sim \Pi_{\psi_n}(\cdot \mid X_{t_n})$ and executes it in the environment. The agent receives a reward which, unbeknownst to the agent, is given by

$$r_{t_{n+1}} = -f(X_{t_n}, \mu_n, A_{t_n})\Delta t$$

(the value $r_{t_{n+1}}$ is known to the agent but the form of $f$ is not). The environment places the agent in a new state $X_{t_{n+1}}$ according to (4.2) using the distribution $\mu_n$ and the action $A_{t_n}$. The previous steps encapsulate what is meant by model free in the sense that the agent has no explicit knowledge of the function $f$ nor the state transition dynamics given by $b, \sigma$; it relies only on the immediate information $r_{t_{n+1}}$ and $X_{t_{n+1}}$. Subsequently, $\Pi_{\psi_n}$ and $V_{\theta_n}$ are updated according to the update rules from Section 3.2. To mimic the infinite horizon regime, we iterate this procedure for a large number of steps until we achieve convergence to the limiting distribution $\hat{\mu}$ (respectively $\mu^*$) and the equilibrium (respectively optimal) control $\hat{\alpha}$ (respectively $\alpha^*$). The complete pseudocode is presented in Algorithm 1.

**Algorithm 1** IH-MF-AC: Infinite Horizon Mean Field Actor-Critic

**Require:** Number of time steps $N \gg 0$; discrete time step size $\Delta t$; neural network learning rates for actor $\rho_\Pi$, critic $\rho_V$, and score $\rho_\Sigma$; Langevin dynamics step size $\epsilon$.

1: Initialize neural networks:
**Actor** $\Pi_{\psi_0} : \mathbb{R}^d \to \mathcal{P}(\mathbb{R}^k)$.
**Critic** $V_{\theta_0} : \mathbb{R}^d \to \mathbb{R}$.
**Score** $\Sigma_{\varphi_0} : \mathbb{R}^d \to \mathbb{R}^d$.
2: Agent receives initial state $X_{t_0}$ from the Environment.
3: **for** $n = 0, \ldots, N-1$ **do**
4: Environment computes score loss: $L_\Sigma^{(n)}(\varphi_n) = \mathrm{tr}(\nabla_x \Sigma_{\varphi_n}(X_{t_n})) + \|\Sigma_{\varphi_n}(X_{t_n})\|_2^2/2$.
5: Environment updates score with SGD: $\varphi_{n+1} = \varphi_n - \rho_\Sigma \nabla_\varphi L_\Sigma^{(n)}(\varphi_n)$.
6: Environment generates mean field samples $S_{t_n} = (S_{t_n}^{(1)}, S_{t_n}^{(2)}, \ldots, S_{t_n}^{(k)})$ from $\Sigma_{\varphi_{n+1}}$ using Langevin dynamics (4.3) with step size $\epsilon$ and compute $\overline{\mu}_{S_{t_n}} := \sum_{i=1}^k \delta_{S_{t_n}^{(i)}}/k$.
7: Agent samples action: $A_{t_n} \sim \Pi_{\psi_n}(\cdot \mid X_{t_n})$.
8: Agent observes reward $r_{n+1}$ and next state $X_{t_{n+1}}$ generated from the environment based on its knowledge of $\overline{\mu}_{S_{t_n}}$.
9: Agent computes TD target: $y_{n+1} = r_{n+1} + e^{-\beta \Delta t} V_{\theta_n}(X_{t_{n+1}})$.
10: Agent computes TD error: $\delta_{\theta_n} = y_{n+1} - V_{\theta_n}(X_{t_n})$.
11: Agent computes critic loss: $L_V^{(n)}(\theta_n) = \delta_{\theta_n}^2$.
12: Agent updates critic with SGD: $\theta_{n+1} = \theta_n - \rho_V \nabla_\theta L_V^{(n)}(\theta_n)$.
13: Agent computes actor loss: $L_\Pi^{(n)}(\psi_n) = -\delta_{\theta_n} \log \Pi_{\psi_n}(A_{t_n} \mid X_{t_n})$.
14: Agent updates actor with SGD: $\psi_{n+1} = \psi_n - \rho_\Pi \nabla_\psi L_\Pi^{(n)}(\psi_n)$.
15: **end for**
16: **return** $(\Pi_{\psi_N}, \Sigma_{\varphi_N})$

### 4.1 Representation of $\mu$ via score-matching

The question of how to represent and update $\mu$ in the continuous-space setting deserves special consideration in this work. In [3, 4], the authors deal with the discrete-space mean field distribution in a natural way, using a normalized vector containing the probabilities of each state. Each individual state is modeled as a one-hot vector (a Dirac delta measure), and the approximation $\mu_n$ is updated at each step using an exponentially weighted update of the form $\mu_{n+1} = \mu_n + \rho_\mu(\delta_{X_{t_n}} - \mu_n)$ with the mean field learning rate $\rho_\mu > 0$. [18] uses a similar update in the context of an AC algorithm for solving only MFC problems, while focusing on a more in-depth treatment of the continuous-time aspect. The authors in [34] tackle continuous-state spaces for the MFG problem using the method of normalizing flows, which pushes forward a fixed latent distribution, such as a Gaussian, using a series of parameterized invertible maps [35]. There is reason to believe that other deep generative models, such as generative adversarial networks (GANs) or variational auto-encoders (VAEs), may yield successful representations of the population distribution with their own drawbacks and advantages.

In our case, partly due to its simplicity of implementation, we opt for the method known as score-matching [22], which has been successfully applied to generative modeling [36]. If $\mu$ has a density function $p_\mu : \mathbb{R}^d \to \mathbb{R}$, then its Stein score function is defined as

$$s_\mu(x) := \nabla \log p_\mu(x).$$

The score function is a useful proxy for $\mu$ in the sense that we can use $s_\mu$ to generate samples from $\mu$ using a Langevin Monte Carlo approach. Given an initial sample $x_0$ from an arbitrary distribution and a small step size $\epsilon > 0$, the sequence defined by

$$x_{m+1} = x_m + \frac{\epsilon}{2} s_\mu(x_m) + \sqrt{\epsilon}\, z_m, \quad z_m \sim \mathcal{N}(0,1) \tag{4.3}$$

converges to a sample from $\mu$ as $m \to \infty$.

From the standpoint of parametric approximation, if $(\Sigma_\varphi)_{\varphi\in\Phi}$ is a sufficiently rich family of functions from $\mathbb{R}^d \to \mathbb{R}^d$, the natural goal is to find the parameters $\varphi$ which minimize the residual $\mathbb{E}_{x\sim\mu}[\|\Sigma_\varphi(x) - s_\mu(x)\|_2^2]$. Although we do not know the true score function, a suitable application of integration by parts yields an expression that is proportional to the previous residual but independent of $s_\mu$

$$\mathbb{E}_{x\sim\mu}\left[\operatorname{tr}\left(\nabla_x \Sigma_\varphi(x)\right) + \frac{1}{2}\|\Sigma_\varphi(x)\|_2^2\right]. \tag{4.4}$$

We adapt the above expression for our online setting in the following way. At the $n$-th step, we have a sample $X_{t_n}$ of the state process and a score representation $\Sigma_{\varphi_n}$. We take the loss function for $\Sigma$ to be

$$L_\Sigma^{(n)}(\varphi_n) := \operatorname{tr}\left(\nabla_x \Sigma_{\varphi_n}(X_{t_n})\right) + \frac{1}{2}\|\Sigma_{\varphi_n}(X_{t_n})\|_2^2. \tag{4.5}$$

Assuming $\Sigma$ is differentiable with respect to $\varphi$, we then update the parameters using the gradient descent step

$$\varphi_{n+1} = \varphi_n - \rho_\Sigma \nabla_\varphi L_\Sigma^{(n)}(\varphi_n), \tag{4.6}$$

where $\rho_\Sigma > 0$ is the mean field learning rate. Now we can generate samples from $\Sigma_{\varphi_{n+1}}$ and take $\mu_n$ to be the empirical distribution of these samples. More concretely, let $S_{t_n} = (S_{t_n}^{(1)}, S_{t_n}^{(2)}, \dots, S_{t_n}^{(k)})$ be the $k$ samples generated from $\Sigma_{\varphi_{n+1}}$ using the Langevin Monte Carlo algorithm in (4.3), and let

$$\mu_n = \overline{\mu}_{S_{t_n}},$$

where the notation $\overline{\mu}_S := \sum_{i=1}^k \delta_{S^{(i)}}/k$ denotes the empirical distribution of the points $S = (S^{(1)}, S^{(2)}, \dots, S^{(k)})$. By the law of large numbers, $\overline{\mu}_{S_{t_n}}$ converges to the true distribution corresponding to $\Sigma_{\varphi_{n+1}}$ as $k \to \infty$.

In the context of generative modeling, the gradient descent update in (4.6) is usually evaluated with several mini-batches of independent samples all from a single distribution. This contrasts with our online approach in which each update is done with the current state $X_{t_n}$, which is generated from a different distribution than the previous state. We justify this as a form of bootstrapping in which we attempt to learn a target distribution that is continuously moving, but ultimately converging to the limiting distribution of the MFG or MFC. Since our updates depend on individual samples, we expect the loss $L_\Sigma$ to be a noisy estimate of the expectation in (4.4), which may slow down convergence. Rather than updating at every timestep, another option would be to perform a batch update after every $m > 1$ timesteps using all samples $(X_{t_n}, X_{t_{n+1}}, \dots, X_{t_{n+(m-1)}})$ generated along the state trajectory, which may accelerate convergence by reducing variance. It is important to acknowledge that the $m$ samples will come from different distributions, so the batch update will also introduce bias into the gradient estimate. This may be mitigated by instead running multiple trajectories in parallel and updating the score function at each step using the samples $(X_{t_n}^{(1)}, X_{t_n}^{(2)}, \dots, X_{t_n}^{(m)})$ from the same timestep.

## 4.2 Unifying mean field game and mean field control problems

Having laid out the general algorithm, we now address the issue of unifying the MFG and MFC formulations in the style of [3, 4].

The intuitions presented in Section 2 regarding the difference between MFG and MFC suggest that the interplay between the learning rates $\rho_\Pi, \rho_V$, and $\rho_\Sigma$ may be used to differentiate between the two solutions of the mean field problem. Taking $\rho_\Sigma < \min\{\rho_\Pi, \rho_V\}$ emulates the notion of solving the classical control problem corresponding to a fixed (frozen) $\mu$ – or, in this case, a slowly moving $\mu$ – and then updating the distribution to match the law of the state process in an iterative manner. This matches the strategy discussed in Section 2.1 for finding an MFG equilibrium. Conversely, taking $\rho_\Sigma > \max\{\rho_\Pi, \rho_V\}$ is more in keeping with simultaneous optimization of the mean field and the policy, which should yield the MFC solution as discussed in Section 2.2.

Borkar's stochastic multi-scale approximation framework [7, 8] has been applied in [3–5] to describe how the choice of learning rates is crucial to derive the solution of different mean field problems using a 2-time scale Q-learning algorithm. Following similar ideas, we show how our method can be expressed as a 3-time scale actor-critic algorithm converging to MFG or MFC solution depending on the choice of learning rates.

In particular, Algorithm 1 is characterized by the following system of numerical updates:

$$\begin{cases} \varphi_{n+1} = \varphi_n - \rho_n^{\Sigma} \nabla_{\varphi} L_{\Sigma}(\varphi_n), \\ \theta_{n+1} = \theta_n - \rho_n^{V} \nabla_{\theta} L_V(\theta_n), \\ \psi_{n+1} = \psi_n - \rho_n^{\Pi} \nabla_{\psi} L_{\Pi}(\psi_n), \end{cases} \tag{4.7}$$

where $L_\Sigma$, $L_V$, and $L_\Pi$ (each of which actually depend implicitly on all three of the parameter vectors) are the loss functions of the Stein score, the critic, and the actor networks, respectively, with $\rho_n^\Sigma, \rho_n^V$ and $\rho_n^\Pi$ being the corresponding learning rates satisfying usual Robbins–Monro type conditions, namely

$$\sum_{n=0}^{\infty} \rho_n^{\Sigma} = \sum_{n=0}^{\infty} \rho_n^{V} = \sum_{n=0}^{\infty} \rho_n^{\Pi} = \infty,$$
$$\sum_{n=0}^{\infty} \left(\rho_n^{\Sigma}\right)^2 = \sum_{n=0}^{\infty} \left(\rho_n^{V}\right)^2 = \sum_{n=0}^{\infty} \left(\rho_n^{\Pi}\right)^2 < \infty.$$

The relationship between learning rates is crucial to determine the convergence of our algorithm. As shown by [9], the actor-critic paradigm can be represented as a multi-scale stochastic approximation procedure in which the updates of the actor evolve at a much slower pace than the ones of the critic. Indeed, as discussed in Section 3.2.1, choosing $\rho_\Pi < \rho_V$ allows the actor $\Pi$ to be seen as frozen by the critic $V$ which can be learned accordingly. Our algorithm extends the traditional 2-time scale framework of actor-critic algorithms to approach mean field problems by adding an additional time scale for learning the mean field distributions.

**Three time scale approach for MFG.** If $\rho_n^\Sigma < \rho_n^\Pi < \rho_n^V$ so that

$$\frac{\rho_n^{\Sigma}}{\rho_n^{\Pi}} \mapsto 0, \quad \frac{\rho_n^{\Pi}}{\rho_n^{V}} \mapsto 0 \quad \text{as} \quad n \mapsto \infty,$$

the system (4.7) tracks the ODE system

$$\begin{cases} \dot{\varphi} = -\nabla_{\varphi} L_{\Sigma}(\varphi, \psi, \theta), \\ \dot{\psi} = -\dfrac{1}{\epsilon} \nabla_{\psi} L_{\Pi}(\varphi, \psi, \theta), \\ \dot{\theta} = -\dfrac{1}{\epsilon\tilde{\epsilon}} \nabla_{\theta} L_V(\varphi, \psi, \theta), \end{cases} \tag{4.8}$$

where $\rho_n^\Sigma/\rho_n^\Pi$ and $\rho_n^\Pi/\rho^V$ are thought of being of order $\epsilon \ll 1$ and $\tilde{\epsilon} \ll 1$, respectively. In the case $\varphi$ and $\psi$ are fixed, we assume the ODE

$$\dot{\theta} = -\frac{1}{\epsilon\tilde{\epsilon}} \nabla_{\theta} L_V(\varphi, \psi, \theta)$$

to have a stable equilibrium $(\varphi, \psi, \theta^{\varphi,\psi})$ corresponding to the local minimum of the operator $L_V$, meaning that the network approximates the value function $V$ corresponding to

the policy $\Pi$ and the population distribution $\mu$ described respectively by the parameters $\psi$ and $\varphi$. Similarly, given fixed $\varphi$, we assume the ODE

$$\dot{\psi} = -\frac{1}{\epsilon}\nabla_{\psi} L_{\Pi}(\varphi, \psi, \theta_{\varphi,\psi})$$

to reach a stable equilibrium $(\varphi, \psi_{\varphi}, \theta_{\varphi,\psi_{\varphi}})$ as the local minimum of the operator $L_{\Pi}$, meaning that the network approximates the optimal policy of the infinitesimal player facing the crowd distribution $\mu$ with Stein score function parameterized by $\varphi$. Finally, the ODE

$$\dot{\varphi} = -\nabla_{\varphi} L_{\Pi}(\varphi, \psi_{\varphi}, \theta_{\varphi,\psi_{\varphi}})$$

stabilizes at $(\varphi^*, \psi_{\varphi^*}, \theta_{\varphi^*,\psi_{\varphi^*}})$, the local minimum of $L_{\Sigma}$ which results the network approximating the Stein score function of the population at equilibrium of the MFG problem.

**Three time scale approach for MFC.** If $\rho_n^{\Pi} < \rho_n^{V} < \rho_n^{\Sigma}$ so that

$$\frac{\rho_n^{\Pi}}{\rho_n^{V}} \;\mapsto\; 0, \quad \frac{\rho_n^{V}}{\rho_n^{\Sigma}} \;\mapsto\; 0 \quad \text{as} \quad n \;\mapsto\; \infty,$$

the system (4.7) tracks the ODE system:

$$\begin{cases} \dot{\psi} = -\nabla_{\psi} L_{\Pi}(\varphi, \psi, \theta), \\ \dot{\theta} = -\dfrac{1}{\epsilon}\nabla_{\theta} L_{V}(\varphi, \psi, \theta), \\ \dot{\varphi} = -\dfrac{1}{\epsilon\tilde{\epsilon}}\nabla_{\varphi} L_{\Sigma}(\varphi, \psi, \theta), \end{cases} \tag{4.9}$$

where $\rho_n^{\Pi}/\rho_n^{V}$ and $\rho_n^{V}/\rho^{\Sigma}$ are thought of being of order $\epsilon \ll 1$ and $\tilde{\epsilon} \ll 1$ respectively. Considering $\psi$ and $\theta$ to be fixed, we assume the ODE

$$\dot{\varphi} = -\frac{1}{\epsilon\tilde{\epsilon}}\nabla_{\varphi} L_{\Sigma}(\varphi, \psi, \theta)$$

to have a stable equilibrium $(\varphi_{\psi,\theta}, \psi, \theta)$ corresponding to the local minimum of the operator $L_{\Sigma}$, meaning that the network approximates the population distribution induced by the policy and value function described respectively by the parameter $\psi$ and $\theta$. Similarly, fixed $\psi$, we assume the ODE

$$\dot{\theta} = -\frac{1}{\epsilon}\nabla_{\theta} L_{V}(\varphi_{\psi,\theta}, \psi, \theta)$$

to reach a stable equilibrium $(\varphi_{\psi,\theta_{\psi}}, \psi, \theta_{\psi})$ as the local minimum of the operator $L_{V}$, meaning that the network approximates the value function corresponding to the policy $\Pi$ described by $\psi$. Finally, the ODE

$$\dot{\psi} = -\nabla_{\psi} L_{\Pi}(\varphi_{\psi,\theta_{\psi}}, \psi, \theta_{\psi})$$

stabilizes at $(\varphi_{\psi^*,\theta_{\psi^*}}, \psi^*, \theta_{\psi^*})$, the local minimum of $L_{\Pi}$ which corresponds to the network approximating the optimal control strategy of the MFC problem.

Writing a rigorous proof of convergence with precise assumptions on these operators is extremely complicated and outside of the scope of this paper. It was recently achieved in the case of Q-learning with finite-state and -action spaces in [5].

# 5 Numerical results

## 5.1 A linear-quadratic benchmark

We test our algorithm on a linear-quadratic (LQ) mean field problem where we wish to optimize

$$\mathbb{E}\left[\int_0^\infty e^{-\beta t}\left(\frac{1}{2}\alpha_t^2 + c_1\left(X_t - c_2 m\right)^2 + c_3\left(X_t - c_4\right)^2 + c_5 m^2\right)\,\mathrm{d}t\right] \tag{5.1}$$

with state dynamics

$$\mathrm{d}X_t = \alpha_t\,\mathrm{d}t + \sigma\,\mathrm{d}W_t, \quad t \in [0,\infty), \tag{5.2}$$

where $m = \int x\,\mu(\mathrm{d}x)$ so that the mean field dependence is only through the first moment of the asymptotic distribution $\mu$. Note that the state dynamics depend only linearly on the control $\alpha$, and the running cost function depends on $\alpha$, $X$, and $m$ quadratically, hence the name linear-quadratic.

The various terms in (5.1) have the following interpretations: the first and last terms penalize $\alpha$ and $m$ from being too large, the second term addresses the relationship between the state process and the mean field distribution, which penalizes $X$ from deviating too far from $c_2 m$, and the third term penalizes $X$ for being far from $c_4$. The coefficients $c_1, c_3$, and $c_5$ determine the relative influence of each term on the total cost.

Both of the one-dimensional MFG and MFC problems corresponding to (5.1) and (5.2) have explicit analytic solutions, which we state now using the notation consistent with the derivations in [3]. We also test our algorithm on the equivalent multivariate problem for which we include the full derivation of the solutions in Appendix A.

## 5.2 Solution for asymptotic mean field game

Traditional methods for deriving the LQ problem solution begin with recovering the value function

$$v(x) := \inf_{\alpha\in\mathbb{A}} \mathbb{E}\left[\int_0^\infty e^{-\beta t}\left(\frac{1}{2}\alpha_t^2 + c_1(X_t - c_2 m)^2 + c_3(X_t - c_4)^2 + c_5 m^2\right)\mathrm{d}t \,\middle|\, X_0 = x\right]$$

as the solution of a Hamilton-Jacobi-Bellman equation. In the MFG case, we denote the optimal value function as $\hat{v}$ and an explicit form is given by

$$\hat{v}(x) = \hat{\Gamma}_2 x^2 + \hat{\Gamma}_1 x + \hat{\Gamma}_0,$$

where

$$\begin{aligned}
\hat{\Gamma}_2 &= \frac{-\beta + \sqrt{\beta^2 + 8(c_1 + c_3)}}{4},\\
\hat{\Gamma}_1 &= -\frac{2\hat{\Gamma}_2 c_3 c_4}{\hat{\Gamma}_2(\beta + 2\hat{\Gamma}_2) - c_1 c_2},\\
\hat{\Gamma}_0 &= \frac{c_5\hat{m}^2 + c_3 c_4^2 + c_1 c_2^2\hat{m}^2 + \sigma^2\hat{\Gamma}_2 - \hat{\Gamma}_1^2/2}{\beta}.
\end{aligned}$$

Then the optimal control for the MFG is

$$\hat{\alpha}(x) = -\hat{v}'(x) = -(2\hat{\Gamma}_2 x + \hat{\Gamma}_1). \tag{5.3}$$

Substituting (5.3) into (5.2) yields the Ornstein-Uhlenbeck process

$$\mathrm{d}\hat{X}_t = -(2\hat{\Gamma}_2 \hat{X}_t + \hat{\Gamma}_1)\,\mathrm{d}t + \sigma\,\mathrm{d}W_t,$$

whose limiting distribution $\hat{\mu} = \lim_{t\to\infty} \mathcal{L}(\hat{X}_t)$ is

$$\hat{\mu} = \mathcal{N}\left(-\frac{\hat{\Gamma}_1}{2\hat{\Gamma}_2}, \frac{\sigma^2}{4\hat{\Gamma}_2}\right). \tag{5.4}$$

Since the mean field interaction for the LQ problem is only through the mean $\hat{m} = \int x\hat{\mu}(\mathrm{d}x)$, we note that a simplified form of $\hat{m}$ is

$$\hat{m} = -\frac{\hat{\Gamma}_1}{2\hat{\Gamma}_2} = \frac{c_3 c_4}{c_1 + c_3 - c_1 c_2}. \tag{5.5}$$

### 5.3 Solution for asymptotic mean field control

Similarly as above, we denote the MFC value function by $v^*$ and claim that is has the form

$$v^*(x) = \Gamma_2^* x^2 + \Gamma_1^* x + \Gamma_0^*$$

with

$$\begin{aligned}
\Gamma_2^* &= \frac{-\beta + \sqrt{\beta^2 + 8(c_1 + c_3)}}{4}, \\
\Gamma_1^* &= -\frac{2\Gamma_2^* c_3 c_4}{\Gamma_2^*(\beta + 2\Gamma_2^*) + c_5 - c_1 c_2(2 - c_2)}, \\
\Gamma_0^* &= \frac{c_5 m^{*2} + c_3 c_4^2 + c_1 c_2^2 m^{*2} + \sigma^2 \Gamma_2^* - \Gamma_1^{*2}/2}{\beta}.
\end{aligned}$$

The optimal control for the MFC is

$$\alpha^*(x) = -v^{*\prime}(x) = -\left(2\Gamma_2^* x + \Gamma_1^*\right). \tag{5.6}$$

Substituting (5.6) into (5.2) yields the Ornstein-Uhlenbeck process

$$\mathrm{d}X_t^* = -\left(2\Gamma_2^* X_t^* + \Gamma_1^*\right)\mathrm{d}t + \sigma\,\mathrm{d}W_t,$$

whose limiting distribution $\mu^* = \lim_{t\to\infty} \mathcal{L}(X_t^*)$ is

$$\mu^* = \mathcal{N}\left(-\frac{\Gamma_1^*}{2\Gamma_2^*}, \frac{\sigma^2}{4\Gamma_2^*}\right). \tag{5.7}$$

Since the mean field interaction is only through the mean $m^* = \int x\,\mu^*(\mathrm{d}x)$, we note that an equation for $m^*$ which only depends explicitly on the running cost coefficients is

$$m^* = -\frac{\Gamma_1^*}{2\Gamma_2^*} = \frac{c_3 c_4}{c_1 + c_3 + c_5 - c_1 c_2(2 - c_2)}. \tag{5.8}$$

### 5.4 A multivariate linear-quadratic benchmark

We also test our algorithm on a higher-dimensional generalization of the scalar LQ problem in (5.2) and (5.1) with the cost functional

$$\mathbb{E}\left[\int_0^\infty e^{\beta t}\left(\frac{1}{2}\alpha_t^\top \alpha_t + (x - C_2 m)^\top C_1 (x - C_2 m) + (x - c_4)^\top C_3 (x - c_4) + m^\top C_5 m\right) \mathrm{d}t\right], \quad (5.9)$$

and state dynamics

$$\mathrm{d}X_t = \alpha_t \,\mathrm{d}t + \sigma \,\mathrm{d}W_t, \quad t \in [0, \infty). \quad (5.10)$$

In this case, $C_1, C_3, C_5 \in \mathbb{R}^{d\times d}$ are positive-definite matrices, $c_4 \in \mathbb{R}^d, C_2 \in \mathbb{R}^{d\times d}, \sigma \in \mathbb{R}^{d\times m}$, and $W_t$ is an $m$-dimensional Brownian motion. We test our algorithm on a benchmark problem for the case $d = m = 2$, and refer the reader to Appendix A for the derivations of the analytic solutions to both the MFG and MFC problems associated with the above system.

### 5.5 Hyperparameters and numerical specifics

For our numerical experiment, we test our algorithm on two different sets of values for the running cost coefficients and volatility $\sigma$ as listed in Tables 5.3 and 5.4 for the one-dimensional case and Table 5.7 for the 2-dimensional case. The discount factor is fixed in all cases to $\beta = 1$, and the continuous time is discretized using step size $\Delta t = 0.01$. The critic and score functions are both feedforward neural networks with one hidden layer of 128 neurons and a `tanh` activation function. For the one-dimensional examples, the actor is also a feedforward neural network that outputs the mean and standard deviation of a normal distribution from which an action is sampled. Its architecture consists of a shared hidden layer of size 64 neurons and a `tanh` activation followed by two separate layers of size 64 neurons for the mean and standard deviation. The standard deviation layer is bookended by a `softmax` activation function to ensure its output is positive. The actor is meant to converge to a deterministic policy – also known as a pure control – over time, so in order to ensure a minimal level of exploration, we add a baseline value of $10^{-5}$ to the output layer. This straightforwardly mimics the notion of entropy regularization detailed in [42]. The actor for the 2-dimensional example uses the same number of hidden layers and hidden neurons, and, correspondingly, outputs a mean vector and a diagonal covariance matrix to induce exploration. Refer to Table 5.1 for the learning rates used by the actor, critic, and score networks which were selected via grid search to give the best results. Table 5.2 summarizes the total parameter count for each neural network.

For the Langevin Monte Carlo iterations, we pick a step size $\epsilon = 5 \times 10^{-2}$ as shown in Table 5.1, and we run 200 iterations at each step using $k = 1000$ samples.

The results of the algorithm applied to the LQ benchmark problem after $N = 10^6$ iterations are displayed in Figs. 5.1, 5.3, 5.6, 5.8, 5.11, 5.13, 5.16 with different sets of parameters along with the corresponding analytic solutions. In addition to the plots of the learned solutions, we also provide plots for various error values to observe convergence with respect to the number of training steps. We consider the following error metrics for the MFG problem and their corresponding counterparts for the MFC problem:

- The absolute error in the learned population mean

$$e_{\hat{m}}(n) := \|m_{t_n} - \hat{m}\|_2,$$

- The expected absolute error in the learned control

$$e_{\hat{\alpha}}(n) := \mathbb{E}_{x\sim\hat{\mu}}\left[\|\alpha_{\psi_n}(x) - \hat{\alpha}(x)\|_2\right] \approx \frac{1}{\ell}\sum_{j=1}^{\ell}\|\alpha_{\psi_n}(x_j) - \hat{\alpha}(x_j)\|_2,$$

- The expected absolute error in the learned value function

$$e_{\hat{v}}(n) := \mathbb{E}_{x\sim\hat{\mu}}\left[\|V_{\theta_n}(x) - \hat{v}(x)\|_2\right] \approx \frac{1}{\ell}\sum_{j=1}^{\ell}\|V_{\theta_n}(x_j) - \hat{v}(x_j)\|_2,$$

  where $x_j$ are i.i.d. with distribution $\hat{\mu}$ for $j = 1, 2, \ldots, \ell$.

Table 5.1: Choice of learning rates used to obtain results in all figures seen in this work – $\rho_\Pi$ for the actor, $\rho_V$ for the critic, $\rho_\Sigma$ for the score function, and $\epsilon$ for the Langevin dynamics. The two left columns are the learning rates used for the case $d = 1$ and the right two for $d = 2$. Boldface values indicate a difference in the learning rate between the MFG and MFC regimes.

| | $d=1$ | | $d=2$ | |
|---|---|---|---|---|
| | MFG | MFC | MFG | MFC |
| $\rho_\Pi$ | $5\times10^{-6}$ | $5\times10^{-6}$ | $5\times10^{-6}$ | $5\times10^{-6}$ |
| $\rho_V$ | $10^{-5}$ | $10^{-5}$ | $10^{-5}$ | $10^{-5}$ |
| $\rho_\Sigma$ | $\mathbf{10^{-6}}$ | $\mathbf{5\times10^{-4}}$ | $\mathbf{10^{-6}}$ | $\mathbf{10^{-5}}$ |
| $\epsilon$ | $5\times10^{-2}$ | $5\times10^{-2}$ | $5\times10^{-2}$ | $5\times10^{-2}$ |

Table 5.2: Parameter counts and activation functions for the actor $\Pi_\psi$, critic $V_\theta$, and score $\Sigma_\varphi$ neural networks used to obtain results in all of the figures seen in this work.

| | Actor | Critic | Score |
|---|---|---|---|
| # parameters | 258 | 385 | 385 |
| activation | `tanh` | `ELU` | `tanh` |

Table 5.3: Running cost coefficients and volatility for (5.1) and (5.2). The results for this parameter set are displayed in Figs. 5.1-5.5.

| $c_1$ | $c_2$ | $c_3$ | $c_4$ | $c_5$ | $\sigma$ |
|---|---|---|---|---|---|
| 0.25 | 1.5 | 0.5 | 0.6 | 1.0 | 0.3 |

Table 5.4: Running cost coefficients and volatility for (5.1) and (5.2). The results for this parameter set are displayed in Figs. 5.6-5.10.

| $c_1$ | $c_2$ | $c_3$ | $c_4$ | $c_5$ | $\sigma$ |
|---|---|---|---|---|---|
| 0.15 | 1.0 | 0.25 | 1.0 | 2.0 | 0.5 |

DOI https://doi.org/10.4208/jml.230919 | Generated on 2025-04-09 07:43:42

It should be noted that taking larger values of the volatility $\sigma$ yields degrading results in the learning of the solution as a result of the lower signal-to-noise ratio in the state samples. Tables 5.5 and 5.6 illustrate the increase in several error metrics at the final time step $N$ for the 1-dimensional experiment with all other values fixed.

We observe many of the same insights alluded to by [3, 4] regarding the differences in recovering the MFG versus the MFC solution. Specifically, convergence to the MFG solution is more stable and faster than convergence to the MFC solution, as evidenced by the convergence plots in Figs. 5.2, 5.7, 5.12, 5.5, 5.10, 5.14, 5.4, 5.9, 5.15. Further, in all cases, there were certain runs in which instability was amplified by the AC algorithm, in which case we saw the weights of the neural networks diverge to numerical overflow. In order to combat this, we imposed a bound on the state space during the first $200,000$ iterations, truncating all states to the interval $[-5,5]$ for one-dimension case and the box $[-5,5]\times[-5,5]$ in the 2-dimensional case. We removed the artificial truncation following the initial iterations and were able to mitigate the instability issues leading to overflow.

Observe that the optimal control is particularly well-learned within the support of the learned distribution. We postulate that a more intricate exploration scheme, perhaps along the lines of entropy regularization [42], may aid in learning the control in a larger domain. We conclude by noting that for all the numerical results in this paper, the gradient descent updates of Algorithm 1 (steps 5, 13, and 15) were computed using the Adam opti-

Table 5.5: Error metrics for $\rho_\Sigma = 10^{-6}$. For increasing values of $\sigma$, we list the $L^2$ error in the learned mean $e_{\hat{m}}$, the expected $L^2$ error in the learned control $e_{\hat{\alpha}}$, and the expected $L^2$ error in the learned value function $e_{\hat{v}}$ after $N = 10^6$ times steps.

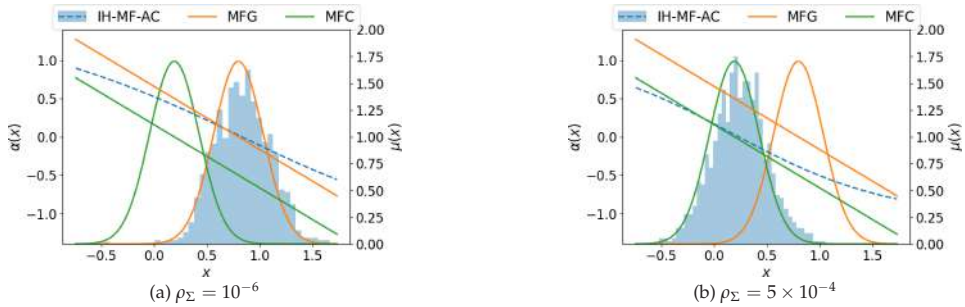
| $\sigma$ | 0.3 | 0.5 | 0.7 | 0.9 | 1.1 |
|---|---|---|---|---|---|
| $e_{\hat{m}}$ | 0.087 | 0.073 | 0.284 | 0.591 | 0.981 |
| $e_{\hat{\alpha}}$ | 0.267 | 0.242 | 0.693 | 0.802 | 1.819 |
| $e_{\hat{v}}$ | 0.178 | 0.394 | 0.522 | 0.673 | 2.005 |

Table 5.6: Error metrics for $\rho_\Sigma = 5\times10^{-4}$. For increasing values of $\sigma$, we list the $L^2$ error in the learned mean $e_{m^*}$, the expected $L^2$ error in the learned control $e_{\alpha^*}$, and the expected $L^2$ error in the learned value function $e_{v^*}$ after $N = 10^6$ time steps.

| $\sigma$ | 0.3 | 0.5 | 0.7 | 0.9 | 1.1 |
|---|---|---|---|---|---|
| $e_{m^*}$ | 0.115 | 0.299 | 1.876 | 2.701 | 3.415 |
| $e_{\alpha^*}$ | 0.583 | 0.441 | 1.106 | 4.556 | 6.946 |
| $e_{v^*}$ | 0.455 | 0.742 | 2.342 | 6.035 | 5.569 |

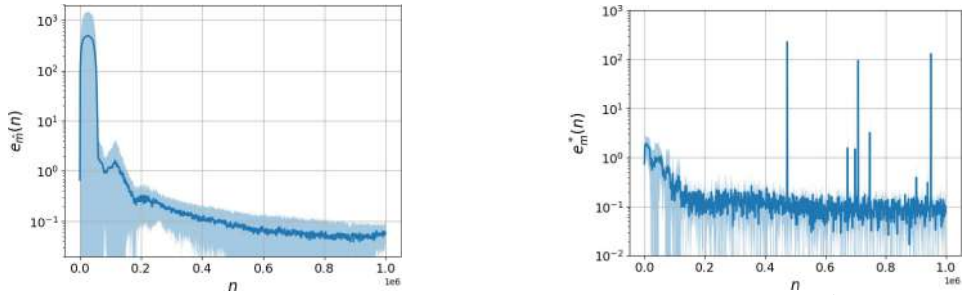
Table 5.7: Running cost coefficients and volatility for (5.9) and (5.10). The results for this parameter set are displayed in Figs. 5.11-5.16. The coefficient values below were randomly generated with the exception of $\sigma$.

| $C_1$ | $C_2$ | $C_3$ | $c_4$ | $C_5$ | $\sigma$ |
|---|---|---|---|---|---|
| $\begin{bmatrix}0.964 & 0.236\\ 0.236 & 0.076\end{bmatrix}$ | $\begin{bmatrix}0.677 & 0.937\\ 0.937 & 1.357\end{bmatrix}$ | $\begin{bmatrix}0.988 & 1.188\\ 1.188 & 1.483\end{bmatrix}$ | $\begin{bmatrix}0.810\\ 0.872\end{bmatrix}$ | $\begin{bmatrix}0.011 & 0.072\\ 0.072 & 0.520\end{bmatrix}$ | $\begin{bmatrix}0.3 & 0\\ 0 & 0.3\end{bmatrix}$ |



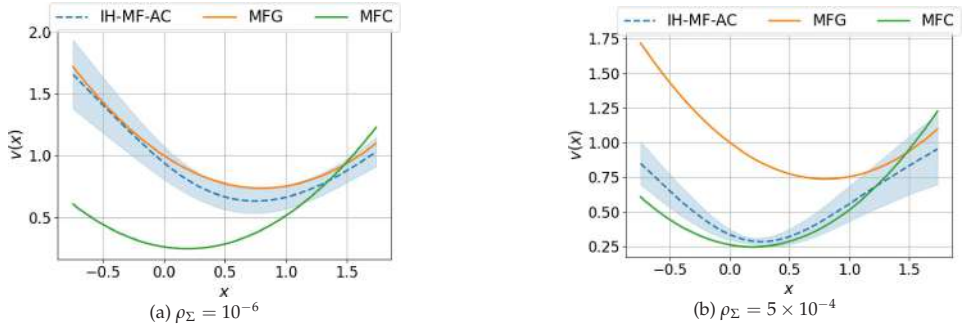

(a) $\rho_\Sigma = 10^{-6}$ (b) $\rho_\Sigma = 5 \times 10^{-4}$

Figure 5.1: The histogram (blue) is the learned asymptotic distribution using samples generated from the parameterized score function $\Sigma_{\varphi_N}$ and the dashed line (blue) is the learned feedback control after $N = 10^6$ iterations averaged over five runs with different initial samples. The green curves correspond to the optimal control and mean field distribution for MFC, while the orange curves are the equivalent for MFG. The bottom axis shows the state variable $x$, the left axis refers to the value of the control $\alpha(x)$, and the right axis represents the probability density of $\mu$.




(a) $\rho_\Sigma = 10^{-6}$ (b) $\rho_\Sigma = 5 \times 10^{-4}$

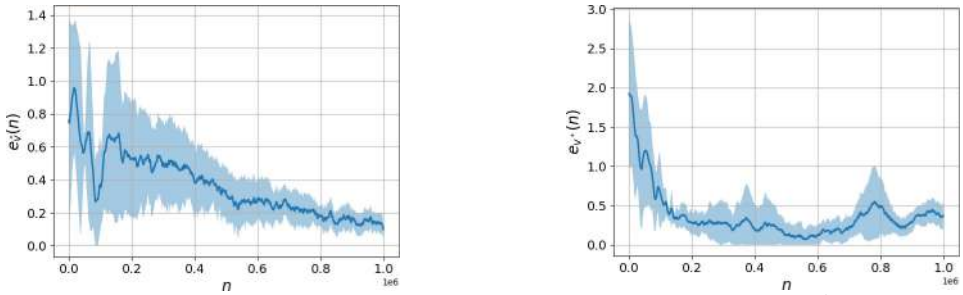
Figure 5.2: We plot the absolute error between the mean of samples produced from the parameterized score function $\Sigma_{\varphi_n}$ and the optimal mean $\hat{m}$ in the case of MFG (left) and $m^*$ in the case of MFC (right). These values were averaged over five runs each with different random initial samples with the standard deviation given by the light blue shaded region. Large jumps are due to random outliers which result from the stochasticity of our algorithm.




(a) $\rho_\Sigma = 10^{-6}$ (b) $\rho_\Sigma = 5 \times 10^{-4}$

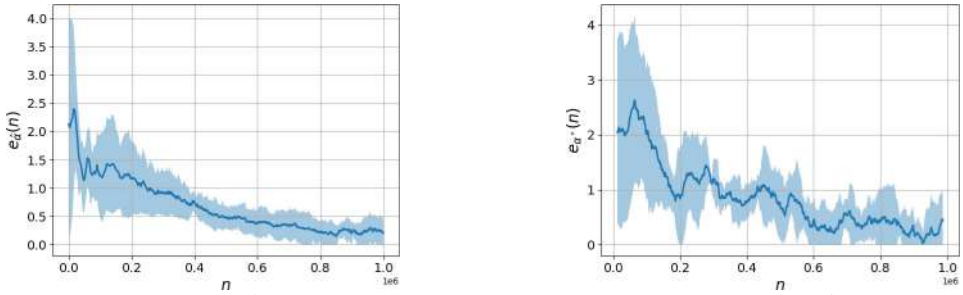
Figure 5.3: The orange and green curves are the optimal value functions for the MFG and MFC problem, respectively. The blue dashed line is the learned value function given by the negative of critic $V_{\theta_N}$ averaged over five runs with different initial samples after $N = 10^6$ iterations. Since the original optimization problem aims to minimize cost while our algorithm seeks to maximize reward, we take the negative of the critic function to make the problems equivalent. The light blue shaded region depicts one standard deviation from the learned value.

DOI https://doi.org/10.4208/jml.230919 | Generated on 2025-04-09 07:43:42



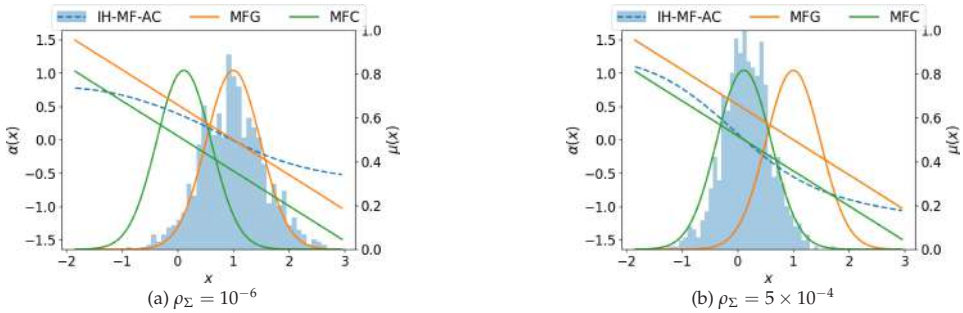

(a) $\rho_\Sigma = 10^{-6}$ (b) $\rho_\Sigma = 5 \times 10^{-4}$

Figure 5.4: We plot the absolute error between the expected error between the learned value function $V_{\theta_n}$ and the optimal value function $\hat{v}$ in the case of MFG (left) and $v^*$ in the case of MFC (right). These plots were averaged over five runs each with different random initial samples and with the standard deviation given by the light blue shaded region. Large jumps are due to random outliers which result from the stochasticity of our algorithm.




(a) $\rho_\Sigma = 10^{-6}$ (b) $\rho_\Sigma = 5 \times 10^{-4}$

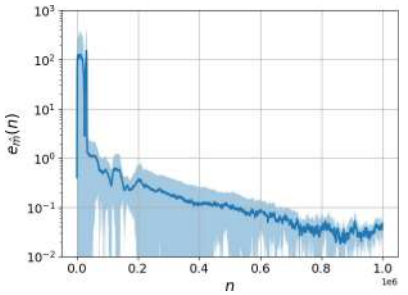
Figure 5.5: We plot the expected error between the learned control function $\alpha_n = \mathbb{E}[\Pi_{\psi_n}]$ and the optimal control $\hat{\alpha}$ in the case of MFG (left) and $\alpha^*$ in the case of MFC (right). These plots were averaged over five runs each with different random initial samples and with the standard deviation given by the light blue shaded region. Large jumps are due to random outliers which result from the stochasticity of our algorithm.



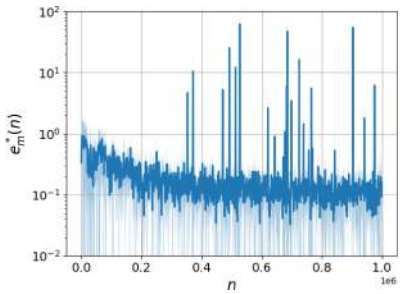

(a) $\rho_\Sigma = 10^{-6}$ (b) $\rho_\Sigma = 5 \times 10^{-4}$

Figure 5.6: The histogram (blue) is the learned asymptotic distribution using samples generated from $\Sigma_{\varphi_n}$ and the dashed line (blue) is the learned feedback control after $N = 10^6$ iterations averaged over five runs with different initial samples. The green curves correspond to the optimal control and mean field distribution for MFC, while the orange curves are the equivalent for MFG. The bottom axis shows the state variable $x$, the left axis refers to the value of the control $\alpha(x)$, and the right axis represents the probability density of $\mu$.

DOI https://doi.org/10.4208/jml.230919 | Generated on 2025-04-09 07:43:42

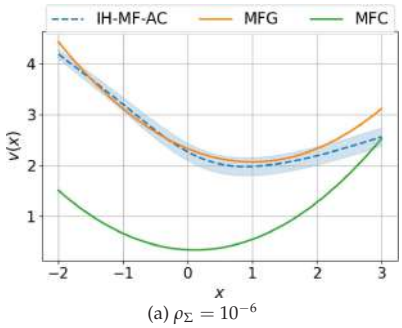



(a) $\rho_\Sigma = 10^{-6}$

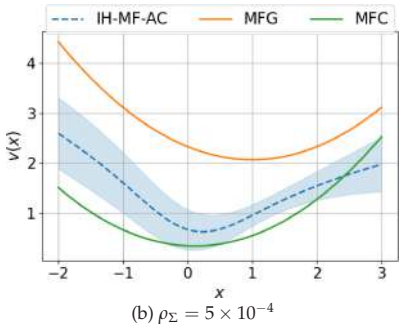



(b) $\rho_\Sigma = 5 \times 10^{-4}$

Figure 5.7: We plot the absolute error between the mean of samples produced from the parameterized score function $\Sigma_{\varphi_n}$ and the optimal mean $\hat{m}$ in the case of MFG (left) and $m^*$ in the case of MFC (right). These values were averaged over five runs each with different random initial samples with the standard deviation given by the light blue shaded region. Large jumps are due to random outliers which result from the stochasticity of our algorithm.

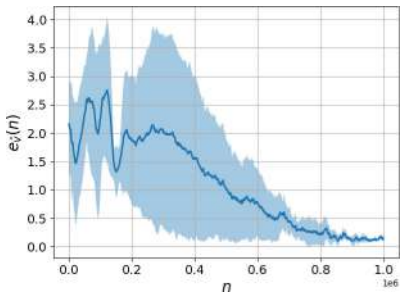



(a) $\rho_\Sigma = 10^{-6}$

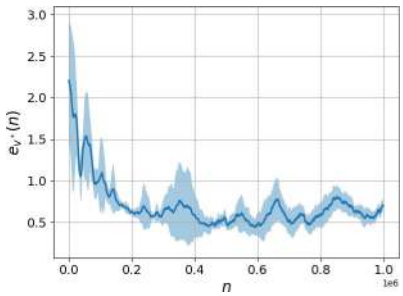



(b) $\rho_\Sigma = 5 \times 10^{-4}$

Figure 5.8: The orange and green curves are the optimal value functions for the MFG and MFC problem, respectively. The blue dashed line is the learned value function given by the negative of critic $V_{\theta_N}$ averaged over five runs with different initial samples after $N = 10^6$ iterations. Since the original optimization problem aims to minimize cost while our algorithm seeks to maximize reward, we take the negative of the critic function to make the problems equivalent. The light blue shaded region depicts one standard deviation from the learned value.



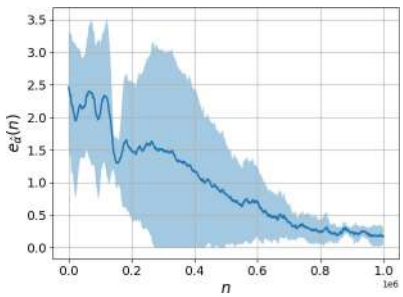

(a) $\rho_\Sigma = 10^{-6}$



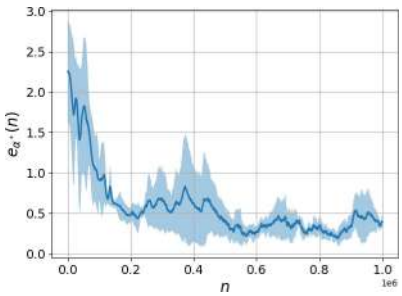

(b) $\rho_\Sigma = 5 \times 10^{-4}$

Figure 5.9: We plot the absolute error between the expected error between the learned value function $V_{\theta_n}$ and the optimal value function $\hat{v}$ in the case of MFG (left) and $v^*$ in the case of MFC (right). These plots were averaged over five runs each with different random initial samples and with the standard deviation given by the light blue shaded region. Large jumps are due to random outliers which result from the stochasticity of our algorithm.

DOI https://doi.org/10.4208/jml.230919 | Generated on 2025-04-09 07:43:42

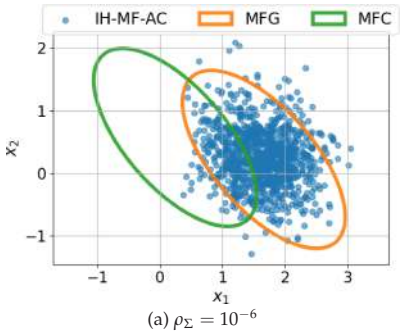



(a) $\rho_\Sigma = 10^{-6}$

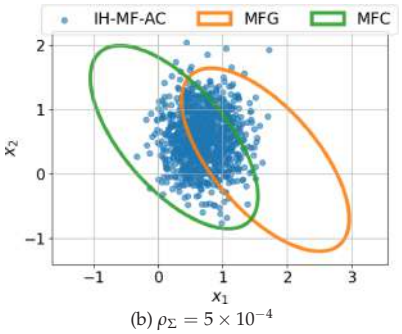



(b) $\rho_\Sigma = 5 \times 10^{-4}$

Figure 5.10: We plot the expected error between the learned control function $\alpha_n = \mathbb{E}[\Pi_{\psi_n}]$ and the optimal control $\hat{\alpha}$ in the case of MFG (left) and $\alpha^*$ in the case of MFC (right). These plots were averaged over five runs each with different random initial samples and with the standard deviation given by the light blue shaded region. Large jumps are due to random outliers which result from the stochasticity of our algorithm.

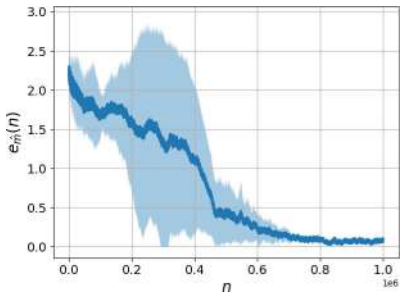



(a) $\rho_\Sigma = 10^{-6}$

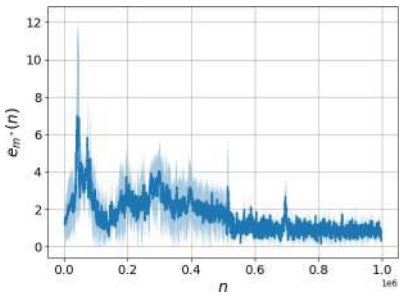



(b) $\rho_\Sigma = 5 \times 10^{-4}$

Figure 5.11: The scatter plot of points (blue) are the samples of the learned asymptotic distribution generated from the parameterized score function $\Sigma_{\varphi_n}$ after $N = 10^6$ iterations. The solid ellipses are the set of points with Mahalanobis distance three from the optimal mean field distributions in the case of MFG (orange) and MFC (green).



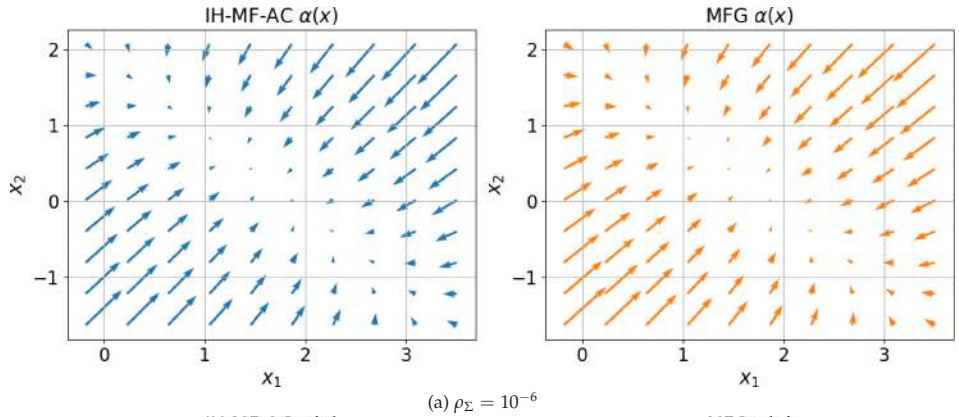

(a) $\rho_\Sigma = 10^{-6}$



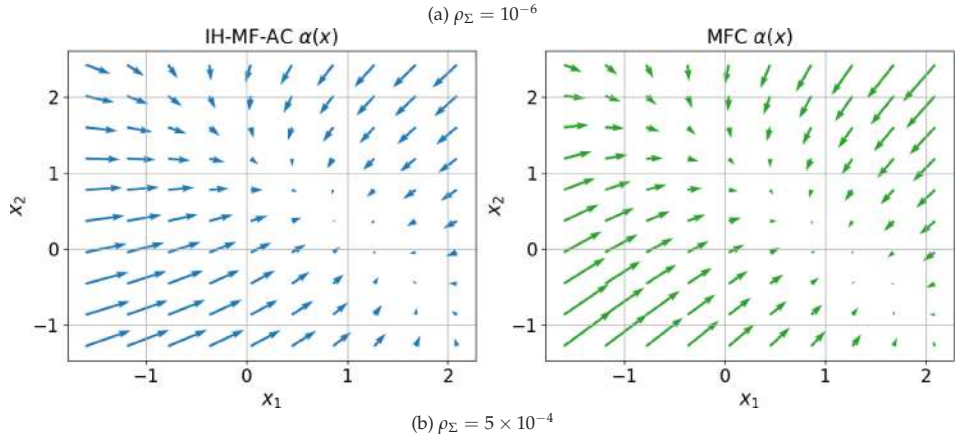

(b) $\rho_\Sigma = 5 \times 10^{-4}$

Figure 5.12: We plot the absolute error between the mean of samples produced from the parameterized score function $\Sigma_{\varphi_n}$ and the optimal mean $\hat{m}$ in the case of MFG (left) and $m^*$ in the case of MFC (right). These values were averaged over five runs each with different random initial samples with the standard deviation given by the light blue shaded region. Large jumps are due to random outliers which result from the stochasticity of our algorithm.

DOI https://doi.org/10.4208/jml.230919 | Generated on 2025-04-09 07:43:42

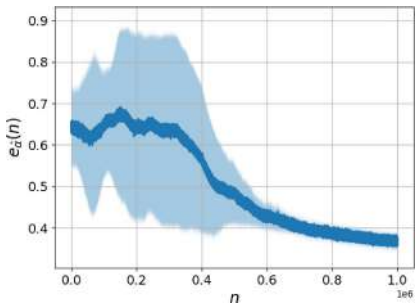



(a) $\rho_\Sigma = 10^{-6}$

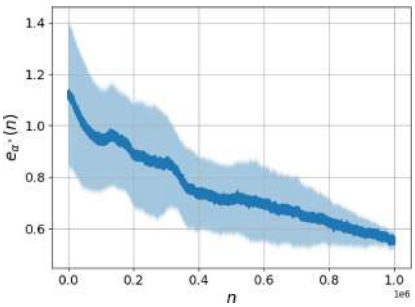



(b) $\rho_\Sigma = 5 \times 10^{-4}$

Figure 5.13: The orange and green vector fields (right) are the optimal controls for the MFG (top) and MFC (bottom) problems, respectively. The blue vector fields (left) show the learned feedback controls after $N = 10^6$ iterations averaged over five runs with different initial samples.



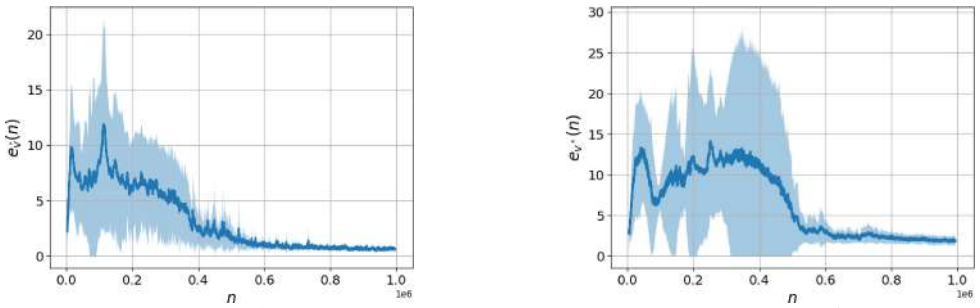

(a) $\rho_\Sigma = 10^{-6}$




(b) $\rho_\Sigma = 5 \times 10^{-4}$

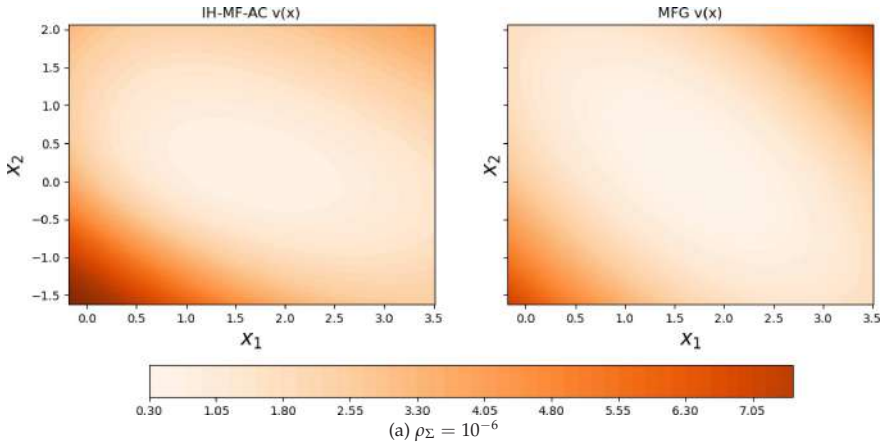
Figure 5.14: We plot the expected error between the learned control function $\alpha_n = \mathbb{E}[\Pi_{\psi_n}]$ and the optimal control $\hat{\alpha}$ in the case of MFG (left) and $\alpha^*$ in the case of MFC (right). These plots were averaged over five runs each with different random initial samples and with the standard deviation given by the light blue shaded region. Large jumps are due to random outliers which result from the stochasticity of our algorithm.

DOI https://doi.org/10.4208/jml.230919 | Generated on 2025-04-09 07:43:42

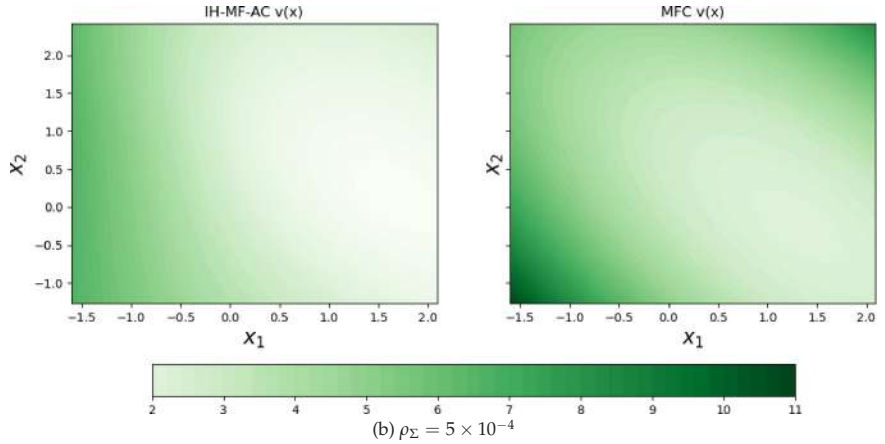



(a) $\rho_\Sigma = 10^{-6}$ (b) $\rho_\Sigma = 5 \times 10^{-4}$

Figure 5.15: We plot the absolute error between the expected error between the learned value function $V_{\theta_n}$ and the optimal value function $\hat{v}$ in the case of MFG (left) and $v^*$ in the case of MFC (right). These plots were averaged over five runs each with different random initial samples and with the standard deviation given by the light blue shaded region. Large jumps are due to random outliers which result from the stochasticity of our algorithm.




(a) $\rho_\Sigma = 10^{-6}$




(b) $\rho_\Sigma = 5 \times 10^{-4}$

Figure 5.16: The right-hand side plots are the optimal value functions for the MFG (top) and MFC (bottom) problems, respectively. The left-hand side plots show the learned functions after $N = 10^6$ iterations averaged over five runs with different initial samples.

DOI https://doi.org/10.4208/jml.230919 | Generated on 2025-04-09 07:43:42

mization update [26] rather than the stochastic gradient descent update suggested in the pseudocode.

We have shown that the same algorithm tested on a continuous-state space infinite horizon LQ problem, with hyperparameters shown in Table 5.1, recovers the MFG or MFC solution. However, our numerical experiments show that there is more stability for the MFG problem, as also observed in the case of the Q-learning algorithm in the context of infinite horizon and finite-space [3], and in [4] in the context of discrete-space finite horizon problems. In the following section, we see that we observe a similar phenomenon in the case of the mixed mean field control game. Note, however, that in our general algorithm we are not taking advantage of the fact that the numerical example is linear-quadratic. Doing so, we would know a priori that the value function is quadratic and that the control is linear, as is done, for instance, in [15].

## 6 Actor-critic algorithm for mean field control games

As observed in [2] in the case of tabular Q-learning, our IH-MF-AC algorithm (Algorithm 1) can easily be extended to the case of mixed mean field control game problems that involve two population distributions, a local one and a global one. This type of game corresponds to competitive games among a large number of large groups, where agents within each group collaborate. The local distribution represents the distribution within each "representative" agent's group, while the global distribution represents the distribution of the entire population.

Such games are motivated as follows: Consider a finite population of agents consisting of $M$ groups, each containing $N$ agents. An agent indexed by $(m, n)$ indicates that she is the $n$-th member of the $m$-th group. Agents collaborate within their respective groups (sharing the same first index) and compete with all agents from other groups. In other words, all $N$ agents in group $m$ collectively work to minimize the total cost of group $m$. The MFCG corresponds to the mean-field limit as $N$ and $M$ tend to infinity, and serves as a proxy for the solution in the finite player case. We refer to [1, 2] for further details on MFCG, including the limit from finite player games to infinite player games. Note that the solution gives an approximation of the Nash equilibrium between the competitive groups.

The solution of an infinite horizon mean field control game is a control-mean field pair $(\hat{\alpha}, \hat{\mu}) \in \mathbb{A} \times \mathcal{P}(\mathbb{R}^d)$ satisfying the following:

1. $\hat{\alpha}$ solves the McKean-Vlasov stochastic optimal control problem

$$\inf_{\alpha \in \mathbb{A}} J_{\hat{\mu}}(\alpha) = \inf_{\alpha \in \mathbb{A}} \mathbb{E}\left[\int_0^\infty e^{-\beta t} f\big(X_t^{\alpha,\hat{\mu}}, \hat{\mu}, \mu^{\alpha,\hat{\mu}}, \alpha\big(X_t^{\alpha,\hat{\mu}}\big)\big)\,\mathrm{d}t\right], \quad \beta > 0, \tag{6.1}$$

subject to

$$\mathrm{d}X_t^{\alpha,\hat{\mu}} = b\big(X_t^{\alpha,\hat{\mu}}, \hat{\mu}, \mu^{\alpha,\hat{\mu}}, \alpha\big(X_t^{\alpha,\hat{\mu}}\big)\big)\,\mathrm{d}t + \sigma\big(X_t^{\alpha,\hat{\mu}}, \hat{\mu}, \mu^{\alpha,\hat{\mu}}, \alpha\big(X_t^{\alpha,\hat{\mu}}\big)\big)\,\mathrm{d}W_t, \quad X_0^{\alpha,\hat{\mu}} = \xi, \tag{6.2}$$

where $\mu^{\alpha,\hat{\mu}} = \lim_{t\to\infty} \mathcal{L}(X_t^{\alpha,\hat{\mu}})$.

2. Fixed point condition $\hat{\mu} = \lim_{t\to\infty} \mathcal{L}(X_t^{\hat{\alpha},\hat{\mu}})$.

Note that conditions 1 and 2 above imply that $\hat{\mu} = \mu^{\hat{\alpha},\hat{\mu}}$.

We modify Algorithm 1 into our infinite horizon mean field control game actor-critic (IH-MFCG-AC) algorithm such that the global score function $\Sigma_\varphi$ represents the global distribution $\hat{\mu}$ and the local score function $\widetilde{\Sigma}_\xi$ represents the local distribution $\mu^{\alpha,\hat{\mu}}$. This is meant to mimic the parallel between the mean field game solution with the global distribution, and the mean field control solution with the local distribution. Following our intuition from Section 4.2, our choice of the now four learning rates will be chosen according to

$$\rho_\Sigma < \min\{\rho_\Pi, \rho_V\} < \max\{\rho_\Pi, \rho_V\} < \rho_{\widetilde{\Sigma}}. \tag{6.3}$$

Refer to Algorithm 2 for the complete pseudocode.

**Algorithm 2** IH-MFCG-AC: Infinite Horizon Mean Field Control Game Actor-Critic

**Require:** Number of time steps $N \gg 0$; discrete time step size $\Delta t$; neural network learning rates for actor $\rho_\Pi$, critic $\rho_V$, global score $\rho_\Sigma$, and local score $\rho_{\widetilde{\Sigma}}$; Langevin dynamics step size $\epsilon$.

Initialize neural networks:
**Actor** $\Pi_{\psi_0} : \mathbb{R}^d \to \mathcal{P}(\mathbb{R}^k)$.
**Critic** $V_{\theta_0} : \mathbb{R}^d \to \mathbb{R}$.
**Global Score** $\Sigma_{\varphi_0} : \mathbb{R}^d \to \mathbb{R}^d$.
**Local Score** $\widetilde{\Sigma}_{\xi_0} : \mathbb{R}^d \to \mathbb{R}^d$.
Agent receives initial state $X_{t_0}$ from the Environment.
**for** $n = 0, \ldots, N-1$ **do**
  Environment computes score loss for $\Sigma$: $L_\Sigma(\varphi_n) = \mathrm{tr}(\nabla_x \Sigma_{\varphi_n}(X_{t_n})) + \|\Sigma_{\varphi_n}(X_{t_n})\|_2^2/2$.
  Environment updates $\Sigma$ with SGD: $\varphi_{n+1} = \varphi_n - \rho_\Sigma \nabla_\varphi L_\Sigma(\varphi_n)$.
  Environment computes score loss for $\widetilde{\Sigma}$: $L_{\widetilde{\Sigma}}(\xi_n) = \mathrm{tr}(\nabla_x \widetilde{\Sigma}_{\xi_n}(X_{t_n})) + \|\widetilde{\Sigma}_{\xi_n}(X_{t_n})\|_2^2/2$.
  Environment updates $\widetilde{\Sigma}$ with SGD: $\xi_{n+1} = \xi_n - \rho_{\widetilde{\Sigma}} \nabla_\xi L_{\widetilde{\Sigma}}(\xi_n)$.
  Environment generates mean field samples $S_{t_n} = (S_{t_n}^{(1)}, S_{t_n}^{(2)}, \ldots, S_{t_n}^{(k)})$ from $\Sigma_{\varphi_{n+1}}$ and $\widetilde{S}_{t_n} = (\widetilde{S}_{t_n}^{(1)}, \widetilde{S}_{t_n}^{(2)}, \ldots, \widetilde{S}_{t_n}^{(k)})$ from $\widetilde{\Sigma}_{\xi_{n+1}}$ using Langevin dynamics (4.3) with step size $\epsilon$ and compute $\overline{\mu}_{S_{t_n}} := \sum_{i=1}^k \delta_{S_{t_n}^{(i)}}/k$ and $\overline{\mu}_{\widetilde{S}_{t_n}} := \sum_{i=1}^k \delta_{\widetilde{S}_{t_n}^{(i)}}/k$.
  Agent samples action: $A_{t_n} \sim \Pi_{\psi_n}(\cdot \mid X_{t_n})$.
  Agent observes reward $r_{n+1}$ and next state $X_{t_{n+1}}$ from the environment.
  Agent computes TD target: $y_{n+1} = r_{n+1} + e^{-\beta \Delta t} V_{\theta_n}(X_{t_{n+1}})$.
  Agent computes TD error: $\delta_{\theta_n} = y_{n+1} - V_{\theta_n}(X_{t_n})$.
  Agent computes critic loss: $L_V(\theta_n) = \delta_{\theta_n}^2$.
  Agent updates critic with SGD: $\theta_{n+1} = \theta_n - \rho_V \nabla_\theta L_V(\theta_n)$.
  Agent computes actor loss: $L_\Pi(\psi_n) = -\delta_{\theta_n} \log \Pi_{\psi_n}(A_{t_n} \mid X_{t_n})$.
  Agent updates actor with SGD: $\psi_{n+1} = \psi_n - \rho_\Pi \nabla_\psi L_\Pi(\psi_n)$.
**end for**
**return** $(\Pi_{\psi_N}, \Sigma_{\varphi_N}, \widetilde{\Sigma}_{\xi_N})$

### 6.1 A linear-quadratic benchmark

We test Algorithm 2 on the following linear-quadratic MFCG. We wish to minimize

$$\mathbb{E}\Bigg[\int_0^\infty e^{-\beta t}\Bigg(\frac{1}{2}\alpha_t^2 + c_1\big(X_t^{\alpha,\mu} - c_2 m\big)^2 + c_3\big(X_t^{\alpha,\mu} - c_4\big)^2 + \tilde{c}_1\big(X_t^{\alpha,\mu} - \tilde{c}_2 m^{\alpha,\mu}\big)^2 + \tilde{c}_5\big(m^{\alpha,\mu}\big)^2\Bigg)\mathrm{d}t\Bigg] \tag{6.4}$$

subject to the dynamics

$$\mathrm{d}X_t^{\alpha,\mu} = \alpha_t\,\mathrm{d}t + \sigma\,\mathrm{d}W_t, \quad t \in [0,\infty), \tag{6.5}$$

where

$$m = \int x\,\mathrm{d}\mu(x), \quad m^{\alpha,\mu} = \int x\,\mathrm{d}\mu^{\alpha,\mu}(x),$$

and the fixed point condition

$$m = \lim_{t\to\infty} \mathbb{E}\big(X_t^{\hat{\alpha},\mu}\big) = m^{\hat{\alpha},\mu},$$

where $\hat{\alpha}$ is the optimal action.

We present the analytic solution to the MFCG problem using notation consistent with the derivation in [2]. The value function is defined as

$$v(x) := \inf_{\alpha\in\mathbb{A}} \mathbb{E}\Bigg[\int_0^\infty e^{-\beta t}\Bigg(\frac{1}{2}\alpha_t^2 + c_1\big(X_t^{\alpha,\mu} - c_2 m\big)^2 + c_3\big(X_t^{\alpha,\mu} - c_4\big)^2 + \tilde{c}_1\big(X_t^{\alpha,\mu} - \tilde{c}_2 m^{\alpha,\mu}\big)^2 + \tilde{c}_5\big(m^{\alpha,\mu}\big)^2\Bigg)\mathrm{d}t \mid X_0 = x\Bigg]. \tag{6.6}$$

The explicit formula $v(x) = \Gamma_2 x^2 + \Gamma_1 x + \Gamma_0$ can be derived as the solution to the Hamilton-Jacobi-Bellman equation where

$$\Gamma_2 = \frac{-\beta + \sqrt{\beta^2 + 8(c_1 + c_3 + \tilde{c}_1)}}{4},$$

$$\Gamma_1 = -\frac{2\Gamma_2 c_3 c_4}{c_1(1-c_2) + \tilde{c}_1(1-\tilde{c}_2)^2 + c_3 + \tilde{c}_5},$$

$$\Gamma_0 = \frac{c_1 c_2^2 m^2 + \big(\tilde{c}_1\tilde{c}_2^2 + \tilde{c}_5\big)(m^{\alpha,\mu})^2 + \sigma^2\Gamma_2 - \Gamma_1^2/2 + c_3 c_4^2}{\beta}.$$

Then the optimal control for the MFCG is

$$\hat{\alpha}(x) = -(2\Gamma_2 x + \Gamma_1). \tag{6.7}$$

Substituting (6.7) into (6.5) yields the Ornstein-Uhlenbeck process

$$\mathrm{d}X_t = -(2\Gamma_2 X_t + \Gamma_1)\,\mathrm{d}t + \sigma\,\mathrm{d}W_t,$$

whose limiting distribution is

$$\hat{\mu} = \mu^{\hat{\alpha},\hat{\mu}} = \mathcal{N}\left(-\frac{\Gamma_1}{2\Gamma_2}, \frac{\sigma^2}{4\Gamma_2}\right). \tag{6.8}$$

We note that an equation for $\hat{m}$ and $m^{\hat{\alpha},\hat{\mu}}$ that only depends on the running cost coefficients is

$$m := \hat{m} = m^{\hat{\alpha},\hat{\mu}} = \frac{c_3 c_4}{c_1(1-c_2) + \tilde{c}_1(1-\tilde{c}_2)^2 + c_3 + \tilde{c}_5}. \tag{6.9}$$

## 6.2 Hyperparameters and numerical specifics

For the LQ benchmark problem, we consider the following choice of parameters: $c_1 = 0.5$, $c_2 = 1.5, c_3 = 0.5, c_4 = 0.25, \tilde{c}_1 = 0.3, \tilde{c}_2 = 1.25, \tilde{c}_5 = 0.25$, discount factor $\beta = 1$, and volatility $\sigma = 0.5$. The time discretization is again $\Delta t = 0.01$. Our intention was to modify as few of the numerical hyperparameters from Section 5 as possible, including the neural network architectures for the actor and critic. The global and local score networks both inherit the architecture from the score network described in Section 5.5 and Table 5.2. The learning rates for the networks are taken directly from Table 5.1 with the global and score network learning rates assuming the values used to obtain the MFG and MFC results, respectively, from Section 5.5. This is to say,

$$(\rho_\Pi, \rho_V, \rho_\Sigma, \rho_{\widetilde{\Sigma}}) = (5 \times 10^{-6}, 10^{-5}, 10^{-6}, 5 \times 10^{-4}),$$

which satisfy $\rho_\Sigma < \rho_\Pi < \rho_V < \rho_{\widetilde{\Sigma}}$, the learning rate inequality proposed in (6.3). The global and local distribution samples are computed at each time step using Langevin dynamics with $\epsilon = 5 \times 10^{-2}$ for 200 iterations using $k = 1000$ samples.

The results of the IH-MFCG-AC algorithm (Algorithm 2) are presented in Figs. 6.1, 6.2, 6.3. As expected, the learning of the global and local distributions reflects that of the optimal MFG distribution and the optimal MFC distribution, respectively. We observe that the global score is learned faster and with more accuracy than the local score, which is prone

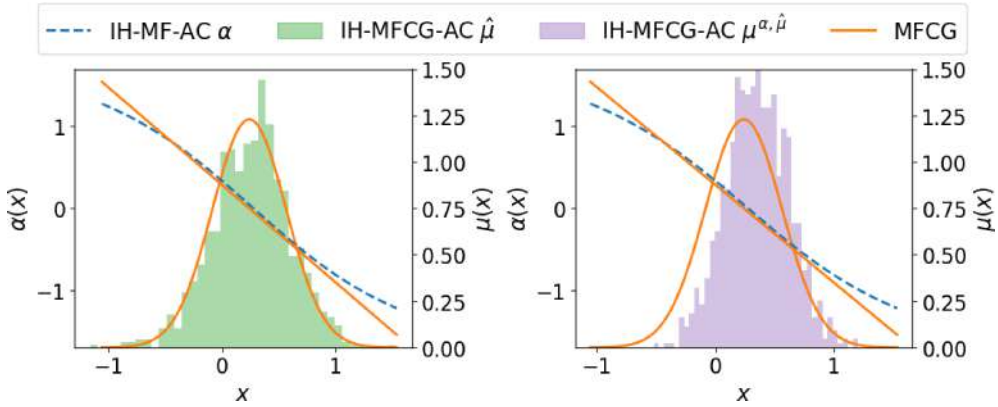



Figure 6.1: The histograms are the learned distributions generated using samples from the global score $\Sigma_{\varphi_n}$ (green) representing the global distribution $\hat{\mu}$ and the local score $\widetilde{\Sigma}_{\tilde{\zeta}_n}$ (purple) representing the local score $\mu^{\alpha,\hat{\mu}}$ after $N = 2 \times 10^6$ iterations. The dashed line (blue) is the learned feedback control averaged over five runs with different initial samples. The benchmark solution to the MFCG is provided in orange. The $x$-axis shows the state variable $x$, the left $y$-axis refers to the value of the control $\alpha(x)$, and the right axis represents the probability density of $\mu(x)$.

DOI https://doi.org/10.4208/jml.230919 | Generated on 2025-04-09 07:43:42

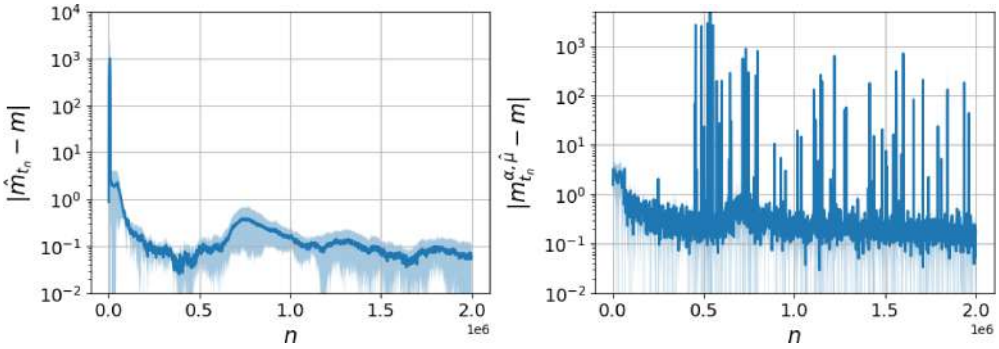



Figure 6.2: The blue curve is a rolling average of the absolute error of the mean of samples produced from the global score function $\Sigma_{\varphi_n}$ (left) – denoted $\hat{m}_{t_n}$ – and the local score function $\widetilde{\Sigma}_{\xi_n}$ – denoted $m^{\alpha,\hat{\mu}}_{t_n}$ – compared to the optimal mean $m$ from (6.9). These values were averaged over five runs each with different random initial samples with the standard deviation given by the light blue shaded region. Large jumps are due to random outliers which result from the stochasticity of our algorithm.

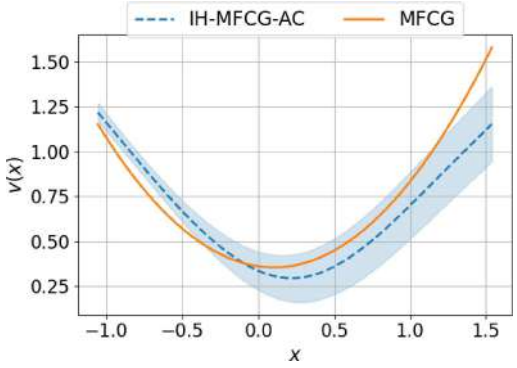



Figure 6.3: The orange curve is the optimal value function for the MFCG problem. The blue dashed line is the learned value function given by the negative of critic $V_{\theta_N}$ averaged over five runs with different initial samples after $N = 2 \times 10^6$ iterations. Since the original optimization problem aims to minimize cost while our algorithm seeks to maximize reward, we take the negative of the critic function to make the problems equivalent. The light blue shaded region depicts one standard deviation from the learned value.

to outliers and instability. The optimal control is learned well within the support of the optimal distribution, but could possibly be expanded with a more advanced exploration strategy.

# 7 Conclusion

We have introduced a novel AC algorithm for solving infinite horizon mean field games and mean field control problems in continuous spaces. This algorithm, called IH-MF-AC, uses neural networks to parameterize a policy and value function, from which an optimal control is derived, as well as a score function, which represents the optimal mean field distribution on a continuous space. The MFG or MFC solution is arrived at depending on

the choice of learning rates for the actor, critic, and score networks. We test our algorithm against a linear-quadratic benchmark problem and are able to recover the analytic solutions with a high degree of accuracy. Finally, we propose and test a modification of the algorithm, called IH-MFCG-AC, to solve the recently developed mixed mean field control game problems. For future work, several directions are worth investigating: enhancing the algorithm for scenarios where the signal-to-noise ratio is low (i.e. $\sigma$ is large) or where the horizon is finite, as well as conducting a rigorous numerical analysis of the proposed algorithm.

# Appendix A Solution for multivariate linear-quadratic asymptotic MFG and MFC problems

For this problem, we assume the state process takes values in $\mathbb{R}^d$. Mimicking the notation established in Section 5.1, let $C_1, C_3, C_5 \in \mathbb{R}^{d\times d}$ be symmetric positive-definite matrices, $c_4 \in \mathbb{R}^d$, and $C_2 \in \mathbb{R}^{d\times d}$. Further, let $W_t$ be an $m$-dimensional Brownian motion and $\sigma \in \mathbb{R}^{d\times m}$. For this class of mean field problems, the running cost is given by

$$f(x,\mu,\alpha) = \frac{1}{2}\alpha^\top\alpha + (x - C_2 m)^\top C_1 (x - C_2 m) + (x - c_4)^\top C_3 (x - c_4) + m^\top C_5 m, \quad \text{(A.1)}$$

and the state dynamics by

$$\mathrm{d}X_t = \alpha_t\,\mathrm{d}t + \sigma\,\mathrm{d}W_t. \quad \text{(A.2)}$$

Recall that $m = \int_{\mathbb{R}^d} x\,\mu(\mathrm{d}x)$.

## A.1 Mean field game

Traditional methods for deriving the LQ problem solution begin with recovering the value function

$$v(x) := \inf_{\alpha\in\mathbb{A}} \mathbb{E}\bigg[\int_0^\infty e^{-\beta t}\bigg(\frac{1}{2}\alpha_t^\top\alpha_t + (X_t - C_2 m)^\top C_1 (X_t - C_2 m) + (X_t - c_4)^\top C_3 (X_t - c_4) + m^\top C_5 m\bigg)\,\mathrm{d}t \,\bigg|\, X_0 = x\bigg] \quad \text{(A.3)}$$

as the solution of a Hamilton-Jacobi-Bellman (HJB) equation. In the MFG case, we denote the optimal value function as $\hat{v}$ and assume it takes the form of the ansatz

$$\hat{v}(x) = x^\top \hat{\Gamma}_2 x + \hat{\Gamma}_1^\top x + \hat{\Gamma}_0. \quad \text{(A.4)}$$

The HJB equation for the value function in the asymptotic infinite horizon setting is

$$\begin{aligned} 0 &= \beta\hat{v}(x) - H(x,\mu) \\ &= \beta\hat{v}(x) - \inf_\alpha \Big\{\alpha^\top\nabla\hat{v}(x) + \frac{1}{2}\operatorname{tr}\left(\sigma\sigma^\top\nabla^2\hat{v}(x)\right) + \frac{1}{2}\alpha^\top\alpha \\ &\qquad + (x - C_2 m)^\top C_1 (x - C_2 m) + (x - c_4)^\top C_3 (x - c_4) + m^\top C_5 m\Big\}, \end{aligned} \quad \text{(A.5)}$$

DOI https://doi.org/10.4208/jml.230919 | Generated on 2025-04-09 07:43:42

where $H$ is the Hamiltonian for the given MFG problem and $\nabla^2$ is the Hessian operator. As the infimum term in (A.5) is quadratic in $\alpha$, the value at which the infimum is attained is

$$\hat{\alpha}(x) = -\nabla\hat{v}(x) \quad \big(= -(\hat{\Gamma}_2 + \hat{\Gamma}_2^\top)x - \hat{\Gamma}_1\big).$$

Substituting this control into the HJB equation yields

$$\begin{aligned} 0 = \beta\hat{v}(x) + \frac{1}{2}\nabla\hat{v}(x)^\top\nabla\hat{v}(x) - \frac{1}{2}\operatorname{tr}\left(\sigma\sigma^\top\nabla^2\hat{v}(x)\right) \\ - (x - C_2m)^\top C_1(x - C_2m) - (x - c_4)^\top C_3(x - c_4) - m^\top C_5 m. \end{aligned}$$

Finally, we plug in our ansatz for the value function and simplify to obtain

$$\begin{aligned} 0 = x^\top\big(2\hat{\Gamma}_2^2 + \beta\hat{\Gamma}_2 - C_1 - C_3\big)x \\ + \big((\beta I + 2\hat{\Gamma}_2)\hat{\Gamma}_1 + 2C_1C_2m + 2C_3c_4\big)^\top x \\ + \beta\hat{\Gamma}_0 + \frac{1}{2}\hat{\Gamma}_1^\top\hat{\Gamma}_1 - \operatorname{tr}\left(\sigma\sigma^\top\hat{\Gamma}_2\right) - m^\top\big(C_2^\top C_1C_2 + C_5\big)m - c_4^\top C_3c_4. \end{aligned}$$

From this we can solve for the coefficients in (A.4) to get

$$\begin{aligned} \hat{\Gamma}_2 &= \frac{1}{4}\Big(-\beta I + \big[\beta^2 I + 8(C_1 + C_3)\big]^{\frac{1}{2}}\Big), \\ \hat{\Gamma}_1 &= (\beta I + 2\hat{\Gamma}_2)^{-1}\left(-2C_1C_2m - 2C_3c_4\right), \\ \hat{\Gamma}_0 &= \frac{1}{\beta}\Big(-\frac{1}{2}\hat{\Gamma}_1^\top\hat{\Gamma}_1 + \operatorname{tr}\left(\sigma\sigma^\top\hat{\Gamma}_2\right) + m^\top\big(C_2^\top C_1C_2 + C_5\big)m + c_4^\top C_3c_4\Big). \end{aligned}$$

Note that the square root of $\beta^2 I + 8(C_1 + C_3)$ in the first equation exists and is unique since this matrix is, in fact, symmetric positive-definite. As a result, $\hat{\Gamma}_2$ is also symmetric positive-definite which means we may rewrite the optimal control as

$$\hat{\alpha}(x) = -2\hat{\Gamma}_2x - \hat{\Gamma}_1. \tag{A.6}$$

To recover the equilibrium mean field distribution, we will substitute (A.6) into the state SDE (A.2) which yields the Ornstein-Uhlenbeck process

$$\mathrm{d}\hat{X}_t = -(2\hat{\Gamma}_2\hat{X}_t + \hat{\Gamma}_1)\mathrm{d}t + \sigma\,\mathrm{d}W_t.$$

Note that since $\hat{\Gamma}_2$ is symmetric positive-definite, $\hat{X}_t$ has a limiting distribution [41]

$$\hat{\mu} = \lim_{t\to\infty}\mathcal{L}(\hat{X}_t).$$

Further, assuming $\hat{X}_0$ is normally distributed, the limiting distribution has an explicit form given by

$$\hat{\mu} = \mathcal{N}\Big(-\frac{1}{2}\hat{\Gamma}_2^{-1}\hat{\Gamma}_1, \hat{\Sigma}\Big). \tag{A.7}$$

Here, the covariance matrix $\hat{\Sigma}$ is defined by

$$\mathrm{vec}(\hat{\Sigma}) = \frac{1}{2}(\hat{\Gamma}_2 \oplus \hat{\Gamma}_2)^{-1}\mathrm{vec}(\sigma\sigma^\top),$$

where $\mathrm{vec}(A)$ is the vector obtained by stacking the columns of $A$ into a vector from left to right and $\oplus$ is the Kronecker sum [41].

Since the mean field interaction for the LQ problem is only through the mean $\hat{m} = \int x\,\hat{\mu}(\mathrm{d}x)$, we note that a simplified form of $\hat{m}$ is

$$\hat{m} = -\frac{1}{2}\hat{\Gamma}_2^{-1}\hat{\Gamma}_1 = [C_1 + C_3 - C_1C_2]^{-1}C_3c_4. \tag{A.8}$$

## A.2 Mean field control

As done previously, we denote the MFC value function by $v^*$ and claim that is has the form

$$v^*(x) = x^\top\Gamma_2^*x + \Gamma_1^{*\top}x + \Gamma_0^*. \tag{A.9}$$

The MFC HJB equation differs from the MFG HJB equation (A.5) with the addition of an extra term involving the derivative of the Hamiltonian with respect to the measure $\mu$

$$0 = \beta v^*(x) - H(x,\mu) - \int_{\mathbb{R}^n}\frac{\delta H}{\delta\mu}(h,\mu)(x)\,\mu(\mathrm{d}h). \tag{A.10}$$

The derivative of $H$ with respect to $\mu$ in the sense of $L$-derivatives [11] is

$$\begin{aligned}
\frac{\delta H}{\delta\mu}(h,\mu)(x) &= \frac{\delta}{\delta\mu}\left[(h - C_2m)^\top C_1(h - C_2m) + m^\top C_5m\right](x)\\
&= \frac{\delta}{\delta\mu}\left[\left(h - C_2\int_{\mathbb{R}^n} y\,\mu(\mathrm{d}y)\right)^\top C_1\left(h - C_2\int_{\mathbb{R}^n} y\,\mu(\mathrm{d}y)\right)\right.\\
&\qquad\left. + \left(\int_{\mathbb{R}^n} y\,\mu(\mathrm{d}y)\right)^\top C_5\int_{\mathbb{R}^n} y\,\mu(\mathrm{d}y)\right](x)\\
&= -2(h - C_2m)^\top C_1C_2x + 2m^\top C_5x,
\end{aligned}$$

and the integral in (A.10) is therefore

$$\int_{\mathbb{R}^n}\frac{\delta H}{\delta\mu}(h,\mu)(x)\,\mu(\mathrm{d}h) = -2m^\top(I - C_2)^\top C_1C_2x + 2m^\top C_5x.$$

Recalling that the optimal control is given by

$$\alpha^*(x) = -\nabla v^*(x) \quad \big(= -\big(\Gamma_2^* + \Gamma_2^{*\top}\big)x - \Gamma_1^*\big),$$

we plug this into (A.10) and simplify to obtain

$$\begin{aligned}
0 = {} & x^\top\big(2\Gamma_2^{*2} + \beta\Gamma_2^* - C_1 - C_3\big)x\\
& + \big((\beta I + 2\Gamma_2^*)\Gamma_1^* + 2\big(C_1C_2 + C_2^\top C_1(I - C_2) - C_5\big)m + 2C_3c_4\big)^\top x\\
& + \beta\Gamma_0^* + \frac{1}{2}\Gamma_1^{*\top}\Gamma_1^* - \mathrm{tr}\left(\sigma\sigma^\top\Gamma_2^*\right) - m^\top\big(C_2^\top C_1C_2 + C_5\big)m - c_4^\top C_3c_4.
\end{aligned}$$

We solve for the coefficients of $v^*(x)$ as before to get

$$\Gamma_2^* = \frac{1}{4}\Big( -\beta I + \big[\beta^2 I + 8(C_1 + C_3)\big]^{\frac{1}{2}}\Big),$$
$$\Gamma_1^* = -2(\beta I + 2\Gamma_2^*)^{-1}\big[\big(C_1C_2 + C_2^\top C_1(I - C_2) - C_5\big)m + C_3c_4\big],$$
$$\Gamma_0^* = \frac{1}{\beta}\Big( -\frac{1}{2}\Gamma_1^{*\top}\Gamma_1^* + \mathrm{tr}\left(\sigma\sigma^\top\Gamma_2^*\right) + m^\top\left(C_2^\top C_1C_2 + C_5\right)m + c_4^\top C_3c_4\Big).$$

As before, we observe that $\Gamma_2^*$ must be symmetric positive-definite, so the optimal control can be rewritten as

$$\alpha^*(x) = -2\Gamma_2^* x - \Gamma_1^*. \tag{A.11}$$

To derive the optimal mean field distribution, we again substitute (A.11) into (A.2) to get the Ornstein-Uhlenbeck process

$$\mathrm{d}X_t^* = -\big(2\Gamma_2^* X_t^* + \Gamma_1^*\big)\,\mathrm{d}t + \sigma\,\mathrm{d}W_t,$$

whose limiting distribution $\mu^* = \lim_{t\to\infty}\mathcal{L}(X_t^*)$ is

$$\mu^* = \mathcal{N}\Big( -\frac{1}{2}\Gamma_2^{*-1}\Gamma_1^*, \Sigma^*\Big), \tag{A.12}$$

assuming that $X_0^*$ is normally distributed. The covariance matrix $\Sigma^*$ is again defined by

$$\mathrm{vec}(\Sigma^*) = \frac{1}{2}\big(\Gamma_2^* \oplus \Gamma_2^*\big)^{-1}\mathrm{vec}(\sigma\sigma^\top).$$

Since the mean field interaction is only through the mean $m^* = \int x\,\mu^*(\mathrm{d}x)$, we note that an equation for $m^*$ which only depends explicitly on the running cost coefficients is

$$m^* = \big[C_1 + C_3 - C_1C_2 - C_2^\top C_1(I - C_2) + C_5\big]^{-1}C_3c_4. \tag{A.13}$$

## Acknowledgment

Ruimeng Hu is grateful to Jingwei Hu for the useful discussions.

Jean-Pierre Fouque was supported by the NSF grant DMS-1953035. Ruimeng Hu was partially supported by the NSF grant DMS-1953035, by the Regents' Junior Faculty Fellowship at UCSB, a grant from the Simons Foundation (MP-TSM-00002783), and an ONR grant under #N00014-24-1-2432. Use was made of computational facilities purchased with funds from the National Science Foundation (CNS-1725797) and administered by the Center for Scientific Computing (CSC). The CSC is supported by the California NanoSystems Institute and by the Materials Research Science and Engineering Center (MRSEC; NSF DMR 1720256) at UC Santa Barbara.

# References


[1] A. Angiuli, N. Detering, J.-P. Fouque, M. Laurière, and J. Lin, Reinforcement learning for intra-and-inter-bank borrowing and lending mean field control game, in: *Proceedings of the Third ACM International Conference on AI in Finance*, ACM, 369–376, 2022.

[2] A. Angiuli, N. Detering, J.-P. Fouque, M. Laurière, and J. Lin, Reinforcement learning algorithm for mixed mean field control games, *J. Mach. Learn.*, 2(2):108–137, 2023.

[3] A. Angiuli, J.-P. Fouque, and M. Laurière, Unified reinforcement Q-learning for mean field game and control problems, *Math. Control Signals Systems*, 34(2):217–271, 2022.

[4] A. Angiuli, J.-P. Fouque, and M. Laurière, Reinforcement learning for mean field games, with applications to economics, in: *Machine Learning and Data Sciences for Financial Markets: A Guide to Contemporary Practices*, Cambridge University Press, 393–425, 2023.

[5] A. Angiuli, J.-P. Fouque, M. Laurière, and M. Zhang, Convergence of multi-scale reinforcement q-learning algorithms for mean field game and control problems, *arXiv:2312.06659*, 2023.

[6] A. Angiuli, C. Graves, H. Li, J.-F. Chassagneux, F. Delarue, and R. Carmona, CEMRACS 2017: Numerical probabilistic approach to MFG, *ESAIM: Proc. Surv.*, 65:84–113, 2019.

[7] V. S. Borkar, Stochastic approximation with two time scales, *Syst. Control. Lett.*, 29(5):291–294, 1997.

[8] V. S. Borkar, *Stochastic Approximation: A Dynamical Systems Viewpoint*, Cambridge University Press, 2008.

[9] V. S. Borkar and V. R. Konda, The actor-critic algorithm as multi-time-scale stochastic approximation, *Sadhana*, 22:525–543, 1997.

[10] R. Carmona, Applications of mean field games in financial engineering and economic theory, in: *Mean Field Games*, Vol. 78 of *Proceedings of Symposia in Applied Mathematics*, AMS, 165–218, 2020.

[11] R. Carmona and F. Delarue, *Probabilistic Theory of Mean Field Games with Applications I-II*, Springer, 2018.

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

[13] R. Carmona and M. Laurière, Deep learning for mean field games and mean field control with applications to finance, in: *Machine Learning and Data Sciences for Financial Markets: A Guide to Contemporary Practices*, Cambridge University Press, 369–392, 2023.

[14] R. Carmona, M. Laurière, and Z. Tan, Model-free mean-field reinforcement learning: Mean-field MDP and mean-field Q-learning, *Ann. Appl. Probab.*, 33(6B):5334–5381, 2023.

[15] X. Chen, J. Duan, Y. Liang, and L. Zhao, Global convergence of two-timescale actor-critic for solving linear quadratic regulator, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, AAAI Press, Vol. 37, 7087–7095, 2023.

[16] K. Cui and H. Koeppl, Approximately solving mean field games via entropy-regularized deep reinforcement learning, in: *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, PMLR, Vol. 130, 1909–1917, 2021.

[17] T. Degris, M. White, and R. S. Sutton, Off-policy actor-critic, in: *Proceedings of the 29th International Conference on Machine Learning*, Omnipress, 179–186, 2012.

[18] N. Frikha, M. Germain, M. Laurière, H. Pham, and X. Song, Actor-critic learning for mean-field control in continuous time, *arXiv:2303.06993*, 2023.

[19] X. Guo, A. Hu, R. Xu, and J. Zhang, Learning mean-field games, in: *Adv. Neural Inf. Process. Syst.*, Vol. 32, Curran Associates, Inc., 2019.

[20] X. Guo, R. Xu, and T. Zariphopoulou, Entropy regularization for mean field games with learning, *Math. Oper. Res.*, 47(4):2547–3399, 2022.

[21] M. Huang, R. P. Malhamé, and P. E. Caines, Large population stochastic dynamic games: Closed-loop McKean-Vlasov systems and the Nash certainty equivalence principle, *Commun. Inf. Syst.*, 6(3):221–252, 2006.

[22] A. Hyvärinen, Estimation of non-normalized statistical models by score matching, *J. Mach. Learn. Res.*, 6(24):695–709, 2005.

[23] Y. Jia and X. Y. Zhou, Policy evaluation and temporal-difference learning in continuous time and space: A martingale approach, *J. Mach. Learn. Res.*, 23(154):1–55, 2022.

[24] Y. Jia and X. Y. Zhou, Policy gradient and actor-critic learning in continuous time and space: Theory and

algorithms, *J. Mach. Learn. Res.*, 23(275):1–50, 2022.
[25] Y. Jia and X. Y. Zhou, Q-learning in continuous time, *J. Mach. Learn. Res.*, 24(161):1–61, 2023.
[26] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, in: *3rd International Conference on Learning Representations, Conference Track Proceedings*, Ithaca, 2015.
[27] V. R. Konda and J. N. Tsitsiklis, On actor-critic algorithms, *SIAM J. Control Optim.*, 42(4):1143–1166, 2003.
[28] J.-M. Lasry and P.-L. Lions, Mean field games, *Jpn. J. Math.*, 2(1):229–260, 2007.
[29] M. Laurière, S. Perrin, S. Girgin, P. Muller, A. Jain, T. Cabannes, G. Piliouras, J. Perolat, R. Elie, O. Pietquin, and M. Geist, Scalable deep reinforcement learning algorithms for mean field games, in: *Proceedings of the 39th International Conference on Machine Learning*, Vol. 162, PMLR, 12078–12095, 2022.
[30] R. P. Malhamé and C. Graves, *Mean Field Games: A Paradigm for Individual-Mass Interactions*, in: *Proceedings of Symposia in Applied Mathematics*, AMS, Vol. 78, 2020.
[31] H. P. McKean, A class of Markov processes associated with nonlinear parabolic equations, *Proc. Natl. Acad. Sci. USA*, 56:1907–1911, 1966.
[32] H. P. McKean, Propagation of chaos for a class of nonlinear parabolic equations, in: *Lecture Series in Differential Equations*, Catholic Univ., Vol. 7, 41–57, 1967.
[33] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, Asynchronous methods for deep reinforcement learning, in: *Proceedings of The 33rd International Conference on Machine Learning*, PMLR, Vol. 48, 1928–1937, 2016.
[34] S. Perrin, M. Laurière, J. Pérolat, M. Geist, R. Élie, and O. Pietquin, Mean field games flock! The reinforcement learning way, in: *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, 356–362, International Joint Conferences on Artificial Intelligence Organization, 2021.
[35] D. J. Rezende and S. Mohamed, Variational inference with normalizing flows, in: *Proceedings of the 32nd International Conference on International Conference on Machine Learning*, JMLR.org, Vol. 37, 1530–1538, 2015.
[36] Y. Song and S. Ermon, Generative modeling by estimating gradients of the data distribution, in: *Adv. Neural Inf. Process. Syst.*, Vol. 32, Curran Associates, Inc., 2019.
[37] R. S. Sutton, Learning to predict by the methods of temporal differences, *Mach. Learn.*, 3(1):9–44, 1988.
[38] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, The MIT Press, 2018.
[39] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, Policy gradient methods for reinforcement learning with function approximation, in: *Adv. Neural Inf. Process. Syst.*, Vol. 12, MIT Press, 1999.
[40] H. van Hasselt, *Reinforcement learning in continuous state and action spaces*, in: *Reinforcement Learning. Adaptation, Learning, and Optimization*, Vol. 12, Springer, 207–251, 2012.
[41] P. Vatiwutipong and N. Phewchean, Alternative way to derive the distribution of the multivariate Ornstein-Uhlenbeck process, *Adv. Differential Equations*, 2019(1):276, 2019.
[42] H. Wang, T. Zariphopoulou, and X. Y. Zhou, Reinforcement learning in continuous time and space: A stochastic control approach, *J. Mach. Learn. Res.*, 21(198):1–34, 2020.
[43] C. Watkins, *Learning From Delayed Rewards*, PhD Thesis, King's College, 1989.
[44] X. Wei and X. Yu, Continuous-time q-learning for mean-field control problems, *arXiv:2306.16208*, 2023.