ESAIM: PROCEEDINGS AND SURVEYS, **October 2024, Vol. 77, p. 145-175**
Emmanuel Franck, Hélène Hivert, Guillaume Latu, Hélène Leman, Bertrand Maury, Michel Mehrenberger, Laurent Navoret

DEEP LEARNING FOR MEAN FIELD OPTIMAL TRANSPORT *

SEBASTIAN BAUDELET[1], BRIEUC FRÉNAIS[2], MATHIEU LAURIÈRE[3], AMAL MACHTALAY[4] AND YUCHEN ZHU[5]

Abstract. Mean field control (MFC) problems have been introduced to study social optima in very large populations of strategic agents. The main idea is to consider an infinite population and to simplify the analysis by using a mean field approximation. These problems can also be viewed as optimal control problems for McKean-Vlasov dynamics. They have found applications in a wide range of fields, from economics and finance to social sciences and engineering. Usually, the goal for the agents is to minimize a total cost which consists in the integral of a running cost plus a terminal cost. In this work, we consider MFC problems in which there is no terminal cost but, instead, the terminal distribution is prescribed as in optimal transport problem. By analogy with MFC, we call such problems mean field optimal transport problems (or MFOT for short) since they can be viewed as a generalization of classical optimal transport problems when mean field interactions occur in the dynamics or the running cost function. We propose three numerical methods based on neural networks. The first one is based on directly learning an optimal control. The second one amounts to solve a forward-backward PDE system characterizing the solution. The third one relies on a primal-dual approach. We illustrate these methods with numerical experiments conducted on two families of examples.

1. INTRODUCTION

Mean field games (MFGs) have been introduced by Lasry and Lions [44–46] and Caines, Huang and Malhamé [42, 43] to approximate Nash equilibria in games with a very large number of players. At a high level, the main idea is to use a mean field approximation to represent the state of the population, and then to focus on the interactions between a single representative player and the distribution of the states of the other players. Mean field control (MFC) [15] relies on a similar approximation but aims at representing situations in which a large number of agents cooperate to minimize a common social cost. The problem can be interpreted as an optimal control problem for a McKean-Vlasov (MKV) stochastic differential equation (SDE) or an optimal control for a Kolmogorov-Fokker-Planck (KFP) partial differential equation (PDE). In the past decade, the analysis of both MFGs and MFC problems has been extensively developed, see e.g. [15] for an introduction to this topic, and [22] for a probabilistic viewpoint.

* *The authors would like to thank the CIRM for welcoming the CEMRACS 2022, the organizers of the CEMRACS 2022 for the opportunity to work on this project as well as their respective institutions. They are also grateful to the NYU-ECNU Institute of Mathematical Sciences at NYU Shanghai, the CIMPA fellowships program, and the ENS Rennes for their support. This work was supported in part through the NYUSH IT High Performance Computing resources, services, and staff expertise, as well as the ASCC Toubkal cluster resources.*

[1] Université Côte d'Azur, 28 Avenue de Valrose, 06103 Nice, France, sebastian.baudelet@univ-cotedazur.fr

[2] IRMA UMR 7501, Université de Strasbourg, 7 Rue René Descartes, 67000 Strasbourg, France, brieuc.frenais@math.unistra.fr

[3] NYU Shanghai Frontiers Science Center of Artificial Intelligence and Deep Learning; NYU-ECNU Institute of Mathematical Sciences at NYU Shanghai, 3663 Zhongshan Road North, Shanghai, 200062, China, mathieu.lauriere@nyu.edu

[4] Mohammed VI Polytechnic University. Lot 660, Hay Moulay Rachid Ben Guerir, 43150, Morocco, amal.machtalay@um6p.ma

[5] Georgia Institute of Technology, 686 Cherry Street NW, Atlanta, 30332, USA, yzhu738@gatech.edu

Article published online by EDP Sciences and available at https://www.esaim-proc.org or https://doi.org/10.1051/proc/202477145

In the most common setup, the players try to minimize a total cost which is composed of a running cost integrated over time and a terminal cost. These costs generally account for the efforts made to control the dynamics as well as the preferences for some states over others. Another class of models has been introduced, in which there is no terminal cost and instead the terminal distribution of the population is imposed as a constraint. Nash equilibria have been studied under the name of planning problem for mean field game. This class of problems has been analyzed mostly using PDE-based techniques [1, 17, 36, 53, 55, 56]. In the special case of linear dynamics and a quadratic running cost in the control, the problem is related to the Schrödinger bridge problem, and equilibrium conditions can be phrased in terms of ordinary differential equations (ODEs) [26–30]. Mean-field Schrödinger bridge problems have also received a growing interest from the stochastic viewpoint, see e.g. [10, 40].

This research direction is tightly connected to optimal transport (OT). Benamou and Brenier proposed in [11] a fluid mechanics framework for the L^2 Monge-Kantorovich mass transport problem. Many works built on this approach to relate optimal transport and optimal control problems for continuity equations. Of particular interest for MFGs is the work [20], which clarified the link between geodesics for a class of distances between probability measures and a PDE system similar to the one arising in MFGs. For more background on OT, we refer the interested reader to the monographs [8, 54, 60, 64, 65]. However, the solutions of MFGs correspond to Nash equilibria, and hence, in general, MFGs do not admit a variational structure. Furthermore, in many applications, it is not immediately clear to us why selfish players caring only about their individual costs would manage to agree and reach a target terminal distribution. Imposing a fixed terminal distribution seems more natural in the MFC setting, where the agents behave in a cooperative way to minimize the social cost. In the present work, we focus on such MFC with planning problems, in which a mean field of agents try to collectively minimize a social cost while ensuring that a fixed distribution is attained at the terminal time.

Since the work of Benamou and Brenier [11], several numerical methods have been investigated for similar problems, including MFGs with planning. Achdou et al. have proposed in [1] a method based on finite differences and Newton method to solve the PDE system of MFG with planning. Benamou and Carlier have used in [12, 14] an Augmented Lagrangian method approach with the alternating direction method of multipliers to solve OT and MFG (without planning). Similar methods have been used in [5, 9] to solve MFGs and MFC problems (still without planning). Benamou et al. proposed in [13] a method to solve MFG with planning through entropy regularization and Sinkhorn algorithm.

Recently, several deep learning methods have been proposed to solve high-dimensional optimal control problems and PDEs, such as the DeepBSDE method [37–39], the Deep Galerkin Method [62] and physics-informed neural networks [57]. Some of these methods have been extended to MFGs and MFC problems. In particular, [7,23] proposed deep learning methods to solve the PDE systems arising in mean field problems, [24, 33, 34] introduced deep learning methods for differential MFC problems. Ruthotto et al. introduced a deep learning method for variational MFG with degenerate diffusion in [59]. Lin et al. introduced a deep learning method in [48] that utilizes the primal-dual relationship of variational MFG. Cao et al. noticed a connection between MFGs, generative adversarial networks and OT in [19]. We refer the interested reader to e.g. [25, 35, 41] for recent surveys on this topic. The work most related to ours is the work of Liu et al. in [49], where they considered the planning problems in a class of MFGs based on a generalized version of the Schrödinger bridge problem and proposed a neural network-based numerical method to solved it.

The main goal of this paper is to propose numerical methods based on deep learning to solve MFC problems with planning constraint, that we will call mean field optimal transport problems. To the best of our knowledge, the theory remains to be investigated in detail, and this is beyond the scope of the present work. Here, we proceed formally when needed, and we focus on the numerical aspects using machine learning tools. The rest of the paper is organized as follows. In Section 2, we introduce the problem and discuss several examples. In Section 3, we describe three numerical methods, each based on a different approach for the problem. In Section 4, we present numerical results on several benchmark problems.

2. DEFINITION OF THE PROBLEM

Before presenting the mean field optimal transport problem, let us first recall the definition of a typical mean field control problem. Let T be a time horizon. Let $\mathcal{Q} = \mathbb{R}^d$ and $\mathcal{Q}_T = [0,T] \times \mathcal{Q}$ denote the space domain and the time-space domain. Denote by $\mathcal{P}_2(\mathcal{Q})$ the set of square-integrable probability measures on $\mathcal{Q}$. Let $f : \mathcal{Q} \times \mathcal{P}_2(\mathcal{Q}) \times \mathbb{R}^k \to \mathbb{R}$ be a running cost function, $g : \mathcal{Q} \times \mathcal{P}_2(\mathcal{Q}) \to \mathbb{R}$ be a terminal cost function, $b : \mathcal{Q} \times \mathcal{P}_2(\mathcal{Q}) \times \mathbb{R}^k \to \mathbb{R}^d$ be a drift function and $\sigma \in \mathbb{R}$ be a non-negative constant diffusion coefficient. In a classical MFC problem with given initial distribution ρ_0 in $\mathcal{P}_2(\mathcal{Q})$, the goal is to find a feedback control $v^* : \mathcal{Q}_T \to \mathbb{R}^k$ minimizing:

$$J^{MFC} : v \mapsto \mathbb{E}\left[\int_0^T f(X_t^v, \mu^v(t), v(t, X_t^v))\mathrm{d}t + g(X_T^v, \mu^v(T))\right] \tag{1}$$

where $\mu^v(t)$ is the distribution of X_t^v, under the constraint that the process $X^v = (X_t^v)_{t\geq 0}$ solves the SDE

$$\begin{cases} X_0^v \sim \rho_0 \\ \mathrm{d}X_t^v = b(X_t^v, \mu^v(t), v(t, X_t^v))\mathrm{d}t + \sigma \mathrm{d}W_t, \qquad t \geq 0, \end{cases} \tag{2}$$

where W is a standard d-dimensional Brownian motion. It would also be interesting to consider open-loop controls, but since we are motivated by numerical applications, we restrict our attention to feedback controls. The cost (1) can be interpreted either as the expected cost for a single representative player, or as the average cost for the whole population, which we refer to as the social cost.

In this work, we are interested in a modified version of the above problem, where instead of having a terminal cost, a terminal distribution is imposed. This type of problem encompasses optimal transport as a special case, but it may incorporate mean field interactions in the drift and the running cost. For this reason, we will refer to this class of problems as mean field optimal transport (MFOT for short).[1] Given two distributions ρ_0 and $\rho_T \in \mathcal{P}_2(\mathcal{Q})$, the goal is to find a feedback control $v^* : \mathcal{Q}_T \to \mathbb{R}^k$ minimizing

$$J^{MFOT} : v \mapsto \mathbb{E}\left[\int_0^T f(X_t^v, \mu^v(t), v(t, X_t^v))\mathrm{d}t\right], \tag{3}$$

where $\mu^v(t)$ is the distribution of X_t^v, under the constraint that the process $X^v = (X_t^v)_{t\geq 0}$ solves the SDE

$$\begin{cases} X_0^v \sim \rho_0, \qquad X_T^v \sim \rho_T \\ dX_t^v = b(X_t^v, \mu^v(t), v(t, X_t^v))\mathrm{d}t + \sigma \mathrm{d}W_t, \qquad t \geq 0. \end{cases} \tag{4}$$

We stress that the terminal constraint implicitly restricts the class of admissible controls since we are interested in minimizing only over controls v that make X_T^v have distribution ρ_T.

We now present a few useful examples, some of which will be revisited in the numerical experiments (see Section 4).

Example 1 (Optimal transport)**.** *When $b(x,\mu,a) = a$, $f(x,\mu,a) = \frac{1}{2}a^\top a$ and $\sigma = 0$, the MFOT problem reduces to a* standard OT *problem. See e.g. [11].*

Example 2 (Linear-quadratic)**.** *Take $b(x,\mu,a) = Ax + \bar{A}\bar{\mu} + Ba$, $f(x,\mu,a) = x^\top Qx + \bar{\mu}^\top \bar{Q}\bar{\mu} + a^\top Ra$, and $g(x,\mu) = x^\top Q_T x + \bar{\mu}^\top \bar{Q}_T \bar{\mu}$, where $\bar{\mu} = \int \xi \mu(\mathrm{d}\xi)$, where $A, \bar{A}, B, Q, \bar{Q}, R, Q_T$ and $\bar{Q}_T$ are matrices of suitable sizes. In this setting, the MFC problem has an explicit solution, up to solving a forward-backward system of ODEs.*

[1] By analogy with MFG of planning type, we could also call such problems "MFC of planning type". Yet another possible terminology would be "mean field control problems with a terminal constraint". But referring to "optimal transport" seems clearer so we will stick to the MFOT terminology. We note that we present the problem with a stochastic dynamics while standard optimal transport problems are usually presented without noise in the dynamics, see [60, 64]. However, the topic of optimal transport with stochastic dynamics has recently attracted interest [52, 63], which justifies our terminology.

Furthermore, if the initial distribution is Gaussian, then the optimal flow of distribution remains Gaussian. See e.g. [15, Chapter 6]. To the best of our knowledge, in the MFOT setting, a similar result is available in the literature only when $Q = \bar{Q} = 0$, which corresponds to the Schrödinger bridge problem. See [30, Section 7.1].

Example 3 (Crowd motion with congestion)**.** *Take $b(x, \mu, a) = a$, $f(x, \mu, a) = (c+\rho\star\mu(x))^\gamma|a|^2+\ell(x, \mu(x))$, where $c \geq 0$ is a constant, ρ is a regularizing kernel and $\star$ denotes the convolution. For $\gamma = 0$, the model is linear-quadratic in the control. If $\gamma > 0$, the cost of moving increases with the density surrounding the agent, which represents the fact that the "energy" spent to move is higher in regions with higher density. This models a* congestion effect*. The last term in f can be used to represent* crowd aversion *if ℓ is increasing with respect to $\mu(x)$, and it can be used to represent spatial preferences by taking for instance $\ell(x, \mu(x)) = |x_* - x|^2$, where x_* is a preferred position. The terminal cost g can also be used to represent crowd aversion or spatial preferences. See e.g. [3, 4] for more details on the analysis of the MFC PDE system for this class of models and [5] for numerical aspects. When $\ell = 0$ and $\sigma = 0$, the corresponding MFOT problem has been studied e.g. in [20]. Similar models have also been studied in the context of MFGs, see e.g. [2, 6].*

3. Numerical methods

In this section, we introduce three different numerical methods to solve MFOT. Section 3.1 introduces a direct approach to solve a MFC problem that approximates the MFOT problem. Section 3.2 discusses the Deep Galerkin Method (DGM) to solve the underlying PDE system that characterizes the optimal solution to MFOT, which is composed of a coupled Hamilton-Jacobi-Bellman equation and a Kolomogrov-Fokker-Planck equation. Section 3.3 introduces the DeepADMM algorithm that solves a variational reformulation of the MFOT problem based on an augmented Lagrangian approach.

3.1. Direct approach for the optimal control formulation

We first introduce the direct approach, which does not require any derivation of optimality conditions. In order to make the problem numerically tractable, we make approximations on several levels. Motivated by the deep learning method for MFC problems proposed in [23] (see also the first algorithm in [25]), we first approximate the MFOT problem (3) by an MFC problem in which a terminal penalty is incurred based on the distance between the terminal distribution and the target distribution. We can then apply the algorithm of [23], which trains a neural network to learn the optimal control of the MFC problem. This method itself relies on three approximations.

3.1.1. *Problem Approximation*

Instead of directly tackling the MFOT problem (3), we first consider the following MFC problem as an approximation of the original problem: Find a feedback control $v^* : \mathcal{Q}_T \to \mathbb{R}^k$ minimizing (1) under the constraint (2) when the terminal cost is:

$$g(x, \mu) = G(\mathcal{W}_2(\mu, \rho_T)), \qquad \mu \in \mathcal{P}_2(\mathcal{Q}). \tag{5}$$

where $G : \mathbb{R}_+ \to \mathbb{R}_+$ is an increasing function and $\mathcal{W}_2$ denotes the Wasserstein distance on $\mathcal{P}_2(\mathcal{Q})$. A typical example that we will use in the experiments is a linear function. The purpose of introducing $G(\mathcal{W}_2(\mu, \rho_T))$ is to add a penalty that enforces the planning constraint for the terminal distribution. Here we focus on the Wasserstein distance because of its connection with optimal transport, see e.g. [11, 60], although other similarity measures could be used. In our numerical experiments, we will take an increasing linear function for G.

Then, we use the following approximations:

- Since it is not possible to optimize overall feedback controls, we restrict the space of controls to the space of neural networks with a given architecture. We will denote by v_θ a representative neural network of this class with parameter θ. The problem becomes a finite-dimensional optimization problem, in which the goal is to find a value for the parameter θ that minimizes the loss $J(v_\theta)$, i.e., the total cost of the MFC problem when using control v_θ.

- Since it is not possible to represent the mean field state $\mu^{v_\theta}(t)$ or to compute its evolution exactly, we approximate it by the empirical distribution $\bar{\mu}^{N,v_\theta}(t) = \frac{1}{N}\sum_{i=1}^{N} \delta_{X_t^{i,v_\theta}}$, where each X_t^{i,v_θ} is a solution of,

$$\begin{cases} X_0^{i,v_\theta} \sim \rho_0 \quad \text{i.i.d.} \\ dX_t^{i,v_\theta} = b(X_t^{i,v_\theta}, \bar{\mu}^{N,v_\theta}(t), v_\theta(t, X_t^{i,v_\theta}))\mathrm{d}t + \sigma \mathrm{d}W_t^i, \qquad t \geq 0, \end{cases} \tag{6}$$

  where $(W^i)_{i=1,\dots,N}$ is a family of N independent d-dimensional Brownian motions, which represent idiosyncratic noises affecting each particle independently. All the SDEs are based on the same control function v_θ.
- Last, in order to be able to compute these dynamics using Monte Carlo simulations, we discretize the time variable t. Letting N_T be a number of regular time steps of length $\Delta t = T/N_T$, we replace the interval $[0, T]$ by the time steps $\{t_0 = 0, t_1 = \Delta t, \dots, t_{N_T} = N_T \Delta t\}$. The time steps are $t_n = n\Delta t$, $n = 0, \dots, N_T$. We then approximate the SDE system (6) using an Euler-Maruyama scheme. The family of trajectories $((X_t^{i,v_\theta})_{t\in[0,T]})_{i=1,\dots,N}$ is approximated by the family of sequences $((X_{t_n}^{i,v_\theta,N_T})_{n=0,\dots,N_T})_{i=1,\dots,N}$ satisfying:

$$\begin{cases} X_0^{i,v_\theta,N_T} \sim \rho_0 \quad \text{i.i.d.} \\ X_{t_{n+1}}^{i,v_\theta,N_T} = X_{t_n}^{i,v_\theta,N_T} + b(X_{t_n}^{i,v_\theta,N_T}, \bar{\mu}_{t_n}^{N,v_\theta,N_T}, v_\theta(t_n, X_{t_n}^{i,v_\theta,N_T}))\Delta t + \sigma \Delta W_n^i, \end{cases} \tag{7}$$

  where $\bar{\mu}_{t_n}^{N,v_\theta,N_T} = \frac{1}{N}\sum_{i=1}^{N} \delta_{X_{t_n}^{i,v_\theta,N_T}}$ is the empirical distribution associated with the samples $X_{t_n}^{i,v_\theta,N_T}$. Here, $(\Delta W_n^i)_{i=1,\dots,N,n=0,\dots,N_T-1}$ are independent Gaussian random variables with variance Δt.

To summarize, the new problem is to find θ^* minimizing:

$$J^{N,N_T}(\theta) = \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N}\sum_{n=0}^{N_T-1} f(X_{t_n}^{i,\theta,N_T}, \bar{\mu}_{t_n}^{N,\theta,N_T}, v_\theta(t_n, X_{t_n}^{i,\theta,N_T}))\Delta t + g(X_T^{N,\theta,N_T}, \bar{\mu}_T^{N,\theta,N_T})\right]$$

subject to the dynamics (7). The full analysis of this problem and its rigorous connection with the original MFOT problem (3) is beyond the scope of this paper and is left for future work. We expect the control v_{θ^*}, with the parameter value that is optimal for the above problem, to be approximately optimal for (3), under suitable assumptions on b and f. In particular, b and f should probably depend smoothly on the distribution so that they can be evaluated in a meaningful way at the empirical distribution $\bar{\mu}_{t_n}^{N,\theta,N_T}$.

3.1.2. *Description of the algorithm*

Optimization method. To find an approximate minimizer, we use stochastic gradient descent (SGD) or one of its variants. At iteration k, we have a parameter θ_k that we wish to update. We sample the initial positions $(X_0^{i,v_{\theta_k},N_T})_{i=1,\dots,N}$ and the Brownian motion increments $(\Delta W_n^i)_{i=1,\dots,N,n=0,\dots,N_T-1}$. We then compute the empirical cost for this realization of the N-particle population, and use its gradient with respect to θ_k to update the parameter. In other words, we apply SGD to the following loss function:

$$\mathcal{L}(\theta) = J^{N,N_T}(\theta) = \mathbb{E}_S[\mathcal{L}(\theta; S)],$$

with:

$$\mathcal{L}(\theta; S) = \frac{1}{N}\sum_{i=1}^{N}\sum_{n=0}^{N_T-1} f(X_{t_n}^{i,v_\theta,N_T}, \bar{\mu}_{t_n}^{N,v_\theta,N_T}, v_\theta(t_n, X_{t_n}^{i,v_\theta,N_T}))\Delta t + g(X_T^{N,v_\theta,N_T}, \bar{\mu}_T^{N,v_\theta,N_T})$$

where $S = \left((X_0^{i,v_\theta,N_T})_{i=1,\dots,N}, (\Delta W_n^i)_{i=1,\dots,N,n=0,\dots,N_T-1}\right)$ denotes one random sample.

Computation of the Wasserstein distance. As shown in (5), the new problem we considered involves a Wasserstein distance between two continuous distributions, namely, the mean field distribution at terminal time μ_T^v and the

target distribution ρ_T. This is in general hard to compute. However, in our implementation, the mean field distribution is approximated by an empirical distribution obtained by Monte Carlo simulations, as is explained above. We then sample the same number of points from the target distribution and compute the Wasserstein distance between the two empirical distributions. This is done in the following way. Let X and Y be two sets of N points each sampled from distributions μ, ν, M_p the distance matrix, $(M_p)_{ij} = |X_i - Y_j|^p$, and the following set:

$$U_N = \left\{ A \in \mathbb{R}^{N\times N} | \sum_{j=1}^{N} A_{ij} = \sum_{i=1}^{N} A_{ij} = \frac{1}{N} \right\}. \tag{8}$$

Then

$$\left(\mathcal{W}_p\left(\mu,\nu\right)\right)^p = \lim_{N\to\infty} \min_{T\in U_N} \left\langle T, M_p \right\rangle.$$

In order to efficiently compute the Wasserstein distance, we follow the algorithm proposed by Cuturi in [31]. We consider an extra entropy regularization of the following form. Let $\alpha > 0$, we want to find T^*_α, which is the solution to the following program:

$$\min_{T\in U_N} \left\langle T, M_p \right\rangle - \alpha \left\langle T\log(T), 1 \right\rangle.$$

Optimality conditions and Sinkhorn-Knopp [61] theorem give us the existence and uniqueness of the solution, as well as a unique decomposition of T^*_α using two vectors u and v such that:

$$T^*_\alpha = \mathrm{Diag}(u) \exp\left(-\frac{M_p}{\alpha}\right) \mathrm{Diag}(v).$$

We can then compute u and v with Sinkhorn-Knopp algorithm. Further explanations on this algorithm can be found in [31]. This method allows for fast computations and is easy to export to greater dimensions, at the cost of adding a layer of approximation due to the extra parameter α. It can be noticed that as α tends to zero, the regularized solution tends to the solution of discrete optimal transport. In practice, reducing α to zero increases the number of iterations required for Sinkhorn algorithm to converge. However, in our numerical experiments we usually obtain good results with a small but non-zero α.

Remark 1. *Notice that using our approach, we have one empirical distribution and one continuous distribution. Indeed, we have the empirical distribution obtained by Monte Carlo simulation and the target distribution ρ_T, which is generally given by a closed-form formula for its density. We could thus try to use the designated methods, such as Semi-discrete Optimal transport [51]. While being more accurate, these methods do not scale well in higher dimensions compared to Sinkhorn's alternative. Another interesting option, which might scale well in terms of dimension is [32, 50]. However, in our numerical experiments, we did not succeed to obtain equally good results using this alternative approach. Further investigation is left for future work.*

Terminal penalty. In our implementation, we take G as a linear function $G(r) = C_W r$, where C_W is a positive constant that weighs the importance of the terminal penalty in comparison with the running cost. This leads to a trade-off between minimizing the running cost and satisfying the terminal constraint. We noticed that when C_W is too small, the algorithm minimizes the running cost without much consideration for the terminal condition and hence the terminal distribution is far from the target distribution. Therefore, the penalization has to be a significant component of the total loss if we want the terminal planning constraint to be approximately satisfied with good accuracy.

3.2. **Deep Galerkin Method for the PDE system**

We now turn our attention to a method based on solving a forward-backward PDE system that characterizes the solution. We first discuss the PDE system and then use a deep learning method to solve this system.

3.2.1. *PDE system for MFOT*

As recalled above, in a standard MFC, the whole population uses a given feedback control v. Assuming that the distribution $\mu_t^v = \mathcal{L}(X_t^v)$ of a representative agent with dynamics (2) admits a smooth enough density m_t, the latter satisfies the Kolmogorov-Fokker-Planck (KFP) PDE:

$$\begin{cases} \frac{\partial m}{\partial t}(t,x) - \nu \Delta m(t,x) + \operatorname{div}(m(t,x) b(x, m(t,\cdot), v(t,x))) = 0 & t \in (0,T], x \in \mathcal{Q} \\ m(0,x) = m_0(x), & x \in \mathcal{Q}, \end{cases}$$

where m_0 is the density of the initial distribution ρ_0 and $\nu = \frac{\sigma^2}{2}$. The MFC problem (1) can then be viewed as an optimal control problem driven by the above KFP PDE. Under suitable conditions, the optimal control can be characterized through an adjoint PDE, which can be derived for instance via calculus of variations. See *e.g.* [15, Chapter 4] for more details.

Let $H : \mathcal{Q} \times L^2(\mathcal{Q}) \times \mathbb{R}^d \to \mathbb{R}$ be the Hamiltonian of the control problem faced by an infinitesimal agent in the first point above, which is defined by:

$$H : (x, m, p) \mapsto H(x, m, p) = \max_{v \in \mathbb{R}^k} \{-L(x, m, v, p)\}, \tag{9}$$

where m denotes the density of μ, $L : \mathcal{Q} \times L^2(\mathcal{Q}) \times \mathbb{R}^k \times \mathbb{R}^d \to \mathbb{R}$ is the Lagrangian, defined by:

$$L : (x, m, v, p) \mapsto L(x, m, v, p) = f(x, m, v) + \langle b(x, m, v), p \rangle. \tag{10}$$

A necessary condition for the existence of an optimal control v^* is that:

$$v^*(t,x) = \operatorname*{argmax}_{v \in \mathbb{R}^k} \{ - L(x, m(t,\cdot), v, \nabla u(t,x))\},$$

where (u, m) solve the following system of partial differential equations:

$$\begin{cases} 0 = -\dfrac{\partial u}{\partial t}(t,x) - \nu \Delta u(t,x) + H(x, m(t,\cdot), \nabla u(t,x)) \\ \qquad + \displaystyle\int_{\mathcal{Q}} \frac{\partial H}{\partial m}(\zeta, m(t,\cdot), \nabla u(t,\zeta))(x) m(t,\zeta) \mathrm{d}\zeta, & \text{in } (0,T] \times \mathcal{Q}, \quad (11a) \\ 0 = \dfrac{\partial m}{\partial t}(t,x) - \nu \Delta m(t,x) - \operatorname{div}(m(t,x) \partial_p H(x, m(t,\cdot), \nabla u(t,x))), & \text{in } [0,T) \times \mathcal{Q}, \quad (11b) \\ u(T,x) = g(x, m(T,\cdot)) + \displaystyle\int_{\mathcal{Q}} \frac{\partial g}{\partial m}(\zeta, m(T,\cdot))(x) m(T,\zeta) \mathrm{d}\zeta, & \text{in } \mathcal{Q}, \quad (11c) \\ m(0,x) = m_0(x), & \text{in } \mathcal{Q}. \quad (11d) \end{cases}$$

The partial derivatives with respect to m appear in the backward PDE due to the fact that the population distribution changes when the control changes. These partial derivatives with respect to m should be understood in the following sense: if $\varphi : L^2(\mathbb{R}^d) \to \mathbb{R}$ is differentiable,

$$\frac{\mathrm{d}}{\mathrm{d}\varepsilon} \varphi(m + \varepsilon \tilde{m})(x)_{|\varepsilon=0} = \int_{\mathbb{R}^d} \frac{\partial \varphi}{\partial m}(m)(\zeta) \tilde{m}(\zeta) \mathrm{d}\zeta.$$

We refer to e.g. [15, Chapter 4] for more details and for the derivation using calculus of variations, which clarifies why the partial derivatives with respect to m appear. If the cost functions and the drift function depend on the density only locally (i.e., only on the density at the current position of the agent), $\frac{\partial}{\partial m}$ becomes a derivative in the usual sense.

In this PDE system, m plays the role of the MFC problem's state. The forward equation is a Kolmogorov-Fokker-Planck (KFP) equation which describes the evolution of the mean field distribution. The other unknown function, u, plays the role of an adjoint state. Although the backward PDE has the form of a Hamilton-Jacobi-Bellman (HJB) equation, u cannot, in general, be interpreted as the value function associated to problem (1) because the value function depends on the population distribution, see e.g. [16, 47]. We refer the interested reader to e.g. [15, Chapters 3 and 4] for the comparison with the MFG PDE system, in which the terms involving a derivative with respect to m are absent, and u can be interpreted as the value function of an infinitesimal player.

Now, for the MFOT problem, we can proceed formally in a similar way. We derive an analogous PDE system, except that the terminal condition for u disappears, and a terminal condition for m is added to the system. More precisely, we (formally) obtain the following PDE system:

$$\begin{cases} 0 = -\dfrac{\partial u}{\partial t}(t,x) - \nu\Delta u(t,x) + H(x, m(t,\cdot), \nabla u(t,x)) \\ \qquad + \displaystyle\int_{\mathcal{Q}} \frac{\partial H}{\partial m}(\zeta, m(t,\cdot), \nabla u(t,\zeta))(x) m(t,\zeta)\mathrm{d}\zeta, & \text{in } (0,T]\times\mathcal{Q}, \quad (12a) \\ 0 = \dfrac{\partial m}{\partial t}(t,x) - \nu\Delta m(t,x) - \operatorname{div}(m(t,x)\partial_p H(x, m(t,\cdot), \nabla u(t,x))), & \text{in } [0,T)\times\mathcal{Q}, \quad (12b) \\ m(0,x) = m_0(x), \qquad m(T,x) = m_T(x) & \text{in } \mathcal{Q}. \quad (12c) \end{cases}$$

where m_0 and m_T are respectively the densities of ρ_0 and ρ_T. To the best of our knowledge, this PDE system has not been derived nor analyzed in a general setting. Notice that even the existence of a solution is a non trivial question due to the fact that there is both an initial and a terminal constraint on the density. However, this system has been analyzed in special cases corresponding to optimal transport [11, 20] or to MFGs with planning [1, 36, 55, 56]. In the numerical examples of Section 4, we will mostly focus on cases that have been previously studied, such as standard optimal transport or MFOT with congestion effects captured by the running cost.

3.2.2. *Description of the algorithm*

To solve the PDE system (12), we follow the idea of the Deep Galerkin Method (DGM) introduced by Sigignano and Spiliopoulos [62] and adapted to the MFG and PDE systems in [7, 24, 25]. The main motivation underlying this approach is to learn the PDE solutions using parameterized functions. This avoids computing the functions on a mesh, which is not feasible in high dimensions. In the DGM, we replace the function(s) solving the PDE(s) with a neural network(s), which are trained to minimize the PDE residual(s) as well as the boundary condition(s).

To be specific, in our setting, we replace the functions m and u with neural networks, denoted by m_θ and u_ω and parameterized by θ and ω respectively. When the state x is in high dimension, i.e., d is large, we expect m_θ and u_ω to provide good approximations of m and u using much fewer parameters than the number of points in a grid. Furthermore, for the numerical implementation, we restrict our attention to a compact domain $\tilde{\mathcal{Q}}$. We denote $\tilde{\mathcal{Q}}_T = [0,T]\times\tilde{\mathcal{Q}}$. We expect the density to have a negligible mass outside a compact set so that by solving the PDE system on a large enough compact set, we obtain a good approximation of the solution, at least in the region where the density is significantly positive. We then define the loss function:

$$\mathcal{L}(\theta,\omega) = \mathcal{L}^{(\mathrm{KFP})}(m_\theta, u_\omega) + \mathcal{L}^{(\mathrm{HJB})}(m_\theta, u_\omega),$$

where, for any $(m,u) \in \mathcal{C}^{1,2}(\tilde{\mathcal{Q}}_T)\times\mathcal{C}^{1,2}(\tilde{\mathcal{Q}}_T)$, the two losses are:

$$\begin{aligned} \mathcal{L}^{(\mathrm{KFP})}(m,u) = C^{(\mathrm{KFP})} \left\| \frac{\partial m}{\partial t} - \nu\Delta m - \operatorname{div}(m\partial_p H(\cdot, m, \nabla u)) \right\|^2_{L^2(\tilde{\mathcal{Q}}_T)} \\ + C_0^{(\mathrm{KFP})} \left\| m(0,\cdot) - m_0 \right\|^2_{L^2(\tilde{\mathcal{Q}})} + C_T^{(\mathrm{KFP})} \left\| m(T,\cdot) - m_T \right\|^2_{L^2(\tilde{\mathcal{Q}})}, \end{aligned} \quad (13)$$

and (omitting the dependence no x and t, and writing explicitly only the dependence on the extra variable ζ in the integral):

$$\mathcal{L}^{(\mathrm{HJB})}(m,u) = C^{(\mathrm{HJB})}\| - \frac{\partial u}{\partial t} - \nu\Delta u + H(\cdot, m, \nabla u) + \int_{\tilde{\mathcal{Q}}} \frac{\partial H}{\partial m}(\zeta, m, \nabla u(\zeta))(\cdot) m(\zeta) \mathrm{d}\zeta \|^2_{L^2(\tilde{\mathcal{Q}}_T)}.$$

Here, $C^{(\mathrm{KFP})}, C_0^{(\mathrm{KFP})}, C_T^{(\mathrm{KFP})}, C^{(\mathrm{HJB})}$ are positive weights that give more or less importance to each component. If the space domain is bounded, we must include more penalty terms. Note that any smooth enough solution (m,u) to the PDE system (12) makes $\mathcal{L}^{(\mathrm{KFP})}$ and $\mathcal{L}^{(\mathrm{HJB})}$ vanish. The goal is to find two neural networks which approximately minimize these losses.

Since it is not possible to compute exactly the above residuals, we approximate the L^2 norms using Monte Carlo samples. For example, we rewrite:

$$\begin{aligned}
&\left\| \frac{\partial m}{\partial t} - \nu\Delta m - \mathrm{div}(m \partial_p H(\cdot, m, \nabla u)) \right\|^2_{L^2(\tilde{\mathcal{Q}}_T)} \\
&= C(\tilde{\mathcal{Q}}_T)\, \mathbb{E}_{\tau,\xi} \left[\left| \frac{\partial m}{\partial t}(\tau,\xi) - \nu\Delta m(\tau,\xi) - \mathrm{div}(m(\tau,\xi)\partial_p H(\xi, m(\tau), \nabla u(\tau,\xi))) \right|^2 \right],
\end{aligned}$$

where (τ,ξ) follows a uniform distribution over $\tilde{\mathcal{Q}}_T$, and $C(\tilde{\mathcal{Q}}_T)$ is a normalizing constant that depends on the domain. Likewise, for the other norms, it would also be possible to use different norms and different distributions to sample (τ,ξ). But for the sake of simplicity, we will stick to this setting for the present work. We obtain the following probabilistic formulation of the loss function $\mathcal{L}$:

$$\mathcal{L}(\theta,\omega) = \mathbb{E}_S\left[\mathcal{L}(\theta,\omega;S)\right], \qquad \mathcal{L}(\theta,\omega;S) = \mathcal{L}^{(\mathrm{KFP})}(m_\theta, u_\omega; S) + \mathcal{L}^{(\mathrm{HJB})}(m_\theta, u_\omega; S),$$

where $S = (\tau, \xi, \xi_0, \xi_T) \in [0,T] \times \tilde{\mathcal{Q}} \times \tilde{\mathcal{Q}} \times \tilde{\mathcal{Q}}$ denotes one sample, and for any $(m,u) \in \mathcal{C}^{1,2}(\mathcal{Q}_T) \times \mathcal{C}^{1,2}(\mathcal{Q}_T)$, the two losses at S are as:

$$\begin{aligned}
\mathcal{L}^{(\mathrm{KFP})}(m,u;S) = C^{(\mathrm{KFP})} &\left| \frac{\partial m}{\partial t}(\tau,\xi) - \nu\Delta m(\tau,\xi) - \mathrm{div}(m(\tau,\xi)\partial_p H(\xi, m(\tau), \nabla u(\tau,\xi))) \right|^2 \\
&+ C_0^{(\mathrm{KFP})} |m(0,\xi_0) - m_0(\xi_0)|^2 + C_T^{(\mathrm{KFP})} |m(T,\xi_T) - m_T(\xi_T)|^2,
\end{aligned}$$

and

$$\begin{aligned}
\mathcal{L}^{(\mathrm{HJB})}(m,u;S) = C^{(\mathrm{HJB})}| &- \frac{\partial u}{\partial t}(\tau,\xi) - \nu\Delta u(\tau,\xi) + H(\xi, m(\tau), \nabla u(\tau,\xi)) \\
&+ \int_{\mathcal{Q}} \frac{\partial H}{\partial m}(\zeta, m(\tau), \nabla u(\tau,\zeta))(\xi) m(\tau,\zeta) \mathrm{d}\zeta|^2.
\end{aligned}$$

Finally, to optimize over (θ,ω), we use SGD (or one of its variants) on the loss $\mathcal{L}$. In practice, we use a mini-batch of samples at each iteration, which amounts to approximate the expectation by an empirical average over several samples.

3.3. **Augmented Lagrangian Method with Deep Learning**

In this subsection, we present an approach based on a primal-dual formulation of the MFOT problem. We then introduce a deep learning adaptation of the alternating direction method of multipliers. We focus on the case when the interactions are local, and the drift is the control.

3.3.1. *Primal and dual problems*

Under suitable assumptions, the MFOT problem admits a variational formulation, which can be tackled using a direct optimization approach. As in the previous subsection, we assume that ρ_0 and ρ_T have respectively density m_0 and m_T.

We focus on a model with local interactions, meaning that an agent at state x interacts with the density of the population at x. To alleviate the presentation, we will use the same notations for the costs and the drift functions, but now their second input is a real number m instead of an element $\mu \in \mathcal{P}_2(\mathcal{Q})$. So we have $f : \mathcal{Q} \times \mathbb{R} \times \mathbb{R}^k \to \mathbb{R}$, $g : \mathcal{Q} \times \mathbb{R} \to \mathbb{R}$, and $b : \mathcal{Q} \times \mathbb{R} \times \mathbb{R}^k \to \mathbb{R}^d$. We also modify accordingly the definition of the Hamiltonian H in (9) and the Lagrangian L in (10) in subsection 3.2. We further assume that $f(x, m, v)$ is convex in v for every (x, m), and $mf(x, m, v)$ is convex in m for every (x, v). For simplicity, we consider that $b(x, m, v) = v$, i.e., the drift is the control. Therefore, in this model, we take $k = d$ (and we will later use k to denote the iteration index).

Primal problem. The MFOT problem (3) introduced in Section 2 is formally equivalent to the following PDE-constrained optimization problem:

$$\begin{aligned} &\inf_{v:\mathcal{Q}_T\to\mathbb{R}^d} \int_{\mathcal{Q}_T} f(x, m(t,x), v(t,x))m(x,t)\mathrm{d}x\,\mathrm{d}t \\ \text{subject to}\quad &\frac{\partial m}{\partial t}(t,x) - \nu\Delta m(t,x) + \operatorname{div}(m(t,x)v(t,x)) = 0 \qquad t \in (0,T], x \in \mathcal{Q} \\ &m(0,x) = m_0(x), \qquad m(T,x) = m_T(x) \end{aligned} \tag{14}$$

The PDE constraint is the KFP equation corresponding to the stochastic dynamics in (4). Note that the formulation in terms of (m, v), while intuitive, is not convex in general. For this reason, we consider an equivalent formulation in terms of $(m, z) = (m, mv)$. We define:

$$\tilde{f}(x, m, z) = \begin{cases} mf\left(x, m, \frac{z}{m}\right) & \text{if } m > 0 \\ 0 & \text{if } (m, z) = (0, 0) \\ +\infty & \text{otherwise} \end{cases} \tag{15}$$

Note that $(m, z) \mapsto \tilde{f}(x, m, z)$ is LSC on $\mathbb{R}\times\mathbb{R}^d$. Under suitable conditions, it can be proved that $(m, z) \mapsto \tilde{f}(x, m, z)$ is convex on $\mathbb{R} \times \mathbb{R}^d$. We also define the space $\mathbf{K}$,

$$\mathbf{K} = \left\{(m, z) \,|\, \frac{\partial m}{\partial t}(t,x) - \nu\Delta m(t,x) + \operatorname{div} z(t,x) = 0, m(0,x) = m_0(x), m(T,x) = m_T(x), m \geq 0\right\} \tag{16}$$

With all these definitions, we are ready to present the primal problem:

$$\inf_{(m,z)\in\mathbf{K}} \mathcal{B}(m, z) = \inf_{(m,z)\in\mathbf{K}} \int_{\mathcal{Q}_T} \tilde{f}(x, m(t,x), z(t,x))\mathrm{d}x\mathrm{d}t \tag{17}$$

Assuming that problem (17) has a unique optimal solution (m^*, z^*) and that problem (14) has a unique optimal control v^*, then the following connection holds: $v^*(t,x) = z^*(t,x)/m^*(t,x)$ if $m^*(t,x) > 0$, $v^*(t,x) = 0$ if $m^*(t,x) = 0$.

Dual problem. We now introduce a dual optimization problem. We define the following functionals:

$$\mathcal{A}(u) = \inf_{m\geq 0} \int_{\mathcal{Q}_T} m(t,x)(\frac{\partial u}{\partial t}(t,x) + \nu\Delta u(t,x) - H(x, m(t,x), \nabla u(t,x)))\mathrm{d}x\,\mathrm{d}t$$
$$+ \int_{\mathcal{Q}} (m_0(x)u(0,x) - m_T(x)u(T,x))\,\mathrm{d}x \tag{18}$$

$$\mathcal{F}(u) = \int_{\mathcal{Q}} m_T(x)u(T,x) - m_0(x)u(0,x)\mathrm{d}x \tag{19}$$

$$\mathcal{G}(\mathfrak{a},\mathfrak{b}) = -\inf_{m\geq 0} \int_{\mathcal{Q}_T} m(t,x)(\mathfrak{a}(t,x) - H(x, m(t,x), \mathfrak{b}(t,x)))\mathrm{d}x\,\mathrm{d}t. \tag{20}$$

Note that if we define the linear differential operator $\Lambda u = (\frac{\partial u}{\partial t} + \nu\Delta u, \nabla u)$, then $\mathcal{A}(u) = -(\mathcal{G}(\Lambda u) + \mathcal{F}(u))$. Similar variation formulation for first-order mean field game, mean field type control, and first-order planning mean field game has been considered e.g. in the works [4, 12, 36] [2]. Here, we generalize the variation formulation to the problem of mean field optimal transport. The functional $\mathcal{A}(u)$ could be considered as the Lagrangian dual function of the primal functional $\mathcal{B}(m,z)$, which can be further related to the sum of two functionals $\mathcal{F}(u)$ and $\mathcal{G}(\Lambda u)$. Consider the following problem:

$$\inf_u \mathcal{F}(u) + \mathcal{G}(\Lambda u). \tag{21}$$

Based on Fenchel-Rockafellar duality theorem (see Section 31, Theorem 31.1 in [58]), we expect problems (17) and (21) to be in duality, meaning:

$$\inf_{(m,z)\in\mathbf{K}} \mathcal{B}(m,z) = \sup_u \mathcal{A}(u) = -\inf_u \mathcal{F}(u) + \mathcal{G}(\Lambda u) \tag{22}$$

Note that this primal-dual relationship also plays an important role in demonstrating the uniqueness and existence of solutions to MFG and MFC PDE systems, see e.g. [4,21,46]. Here, we expect a similar result to hold for MFOT under suitable conditions. The rigorous definition of the two problems and the analysis of this duality relationship is left for future work. For now, we proceed formally.

We can at least formally establish a connection between the primal problem, the dual problem, and the optimal control in the following way. Let u^* be the optimal solution to the dual (21) and let (m^*, z^*) be the optimal solution to the primal problem (17). Then the optimal control for the original problem (14) is given by:

$$v^*(t,x) = \partial_p H(x, m^*(t,x), \nabla u^*(t,x)).$$

We notice that (u^*, m^*) forms a solution to the MFOT PDE system (12). This fact suggests that we can work on the dual problem (21) directly to solve the MFOT problem. Under suitable assumptions, it can be shown that the dual problem (21) is a strongly convex, unconstrained optimization problem over function space, which motivates the use of classic algorithms in convex optimization. However, the presence of the infinite dimensional linear operator Λ makes the problem hard to solve efficiently in general. Fortunately, the structure of the objective as a sum of two convex functionals makes the problem amenable to algorithms based on splitting schemes, such as the Alternating Direction Method of Multipliers (ADMM) [18].

3.3.2. *Description of the algorithm*

Introducing a new variable q that will play the role of Λu, we can rewrite problem (21) as the following constrained optimization program:

$$\inf_{u,q:\, q=\Lambda u} \mathcal{F}(u) + \mathcal{G}(q). \tag{23}$$

[2] See Lemma 2.2 in [4], Lemma 2.1 in [36]

Algorithm 1: Vanilla ADMM for MFOT

Data: Initial Guess $(u^{(0)}, q^{(0)}, \lambda^{(0)})$; number of iterations N; hyperparameter $r > 0$
Result: Function $(u^{(N)}, q^{(N)}, \lambda^{(N)})$ that are close to the saddle point of $\mathcal{L}_r$ defined in (24)

begin
  for $k = 1, \cdots, N$, **do**
    $u^{(k)} = \underset{u:\mathcal{Q}_T \to \mathbb{R}}{\arg\min}\ \mathcal{F}(u) - \langle \lambda^{(k-1)}, \Lambda u \rangle + \frac{r}{2}\|\Lambda u - q^{(k-1)}\|^2$
    $q^{(k)} = \underset{q:\mathcal{Q}_T \to \mathbb{R}^{d+1}}{\arg\min}\ \mathcal{G}(q) + \langle \lambda^{(k-1)}, q \rangle + \frac{r}{2}\|\Lambda u^{(k)} - q\|^2$
    $\lambda^{(k)} = \lambda^{(k-1)} - r(\Lambda u^{(k)} - q^{(k)})$

The goal is now to find a saddle point of the associated Lagrangian. In fact, for numerical purposes, we will consider an augmented Lagrangian, defined as follows.

Let $r > 0$ be a constant and introduce $\lambda : \mathcal{Q}_T \to \mathbb{R}^{d+1}$, the Lagrangian multiplier associated with the constraint $q = \Lambda u$. Let $\langle \cdot, \cdot \rangle$ denote the inner product on $L^2(\mathcal{Q}_T)$. We introduce the augmented Lagrangian:

$$\mathcal{L}_r(u, q, \lambda) = \mathcal{F}(u) + \mathcal{G}(q) - \langle \lambda, \Lambda u - q \rangle + \frac{r}{2}\|\Lambda u - q\|^2 \tag{24}$$

Now, the original MFOT problem is reduced to finding a saddle point of $\mathcal{L}_r$. Here, we state the original ADMM method in Algorithm 1 that finds the saddle point via an alternating optimization procedure.

This general procedure can be implemented, for example, when the functions (u, q, λ) are approximated by their values on a finite-difference grid. Such a procedure has been used for MFG and MFC problems, using finite elements [9, 12] or finite differences [5]. Furthermore, Benamou et.al. proved that under the existence of solution and the full column rank condition of the finite-difference differential operator Λ, the ADMM method converges to the desired solution for mean field games, see Theorem 3.1 in [12]. However, as already mentioned, approximating functions by their values on a mesh is not feasible in high dimensions. We thus propose a different implementation of the ADMM based on neural network approximations.

In Algorithm 1, the objectives in the steps are given by functionals to be minimized over functional spaces, which is not tractable in general. We restrict our attention to spaces of parameterized functions that can be expressed as neural networks, denoted by $(u_\theta, q_\omega, \lambda_\psi)$ with parameter θ, ω, ψ respectively. We then follow the strategy introduced with the DGM [62] and already used in Section 3.2 to create computable loss functions that are stochastic approximations of the functionals.

Recall that the truncated space domain $\tilde{\mathcal{Q}}$ and the associated time-space domain $\tilde{\mathcal{Q}}_T$. Let $X \sim \mathcal{U}(\tilde{\mathcal{Q}}_T), Y \sim \mathcal{U}(\tilde{\mathcal{Q}})$ be two random variables with uniform distribution in the time-space domain and the space domain respectively. Let ρ_X, ρ_Y be the value of the uniform density on $\tilde{\mathcal{Q}}_T$ and $\tilde{\mathcal{Q}}$ respectively. Here, we overload the notation $\langle \cdot, \cdot \rangle$ and $\|\cdot\|$ to represent the Euclidean inner product and norm on both $L_2(\tilde{\mathcal{Q}}_T)$ and $\mathbb{R}^{d+1}$:

$$\mathcal{L}^{(u)}(\theta; \omega, \psi) = \mathcal{L}_1(u_\theta, q_\omega, \lambda_\psi), \quad \mathcal{L}^{(q)}(\omega; \theta, \psi) = \mathcal{L}_2(u_\theta, q_\omega, \lambda_\psi), \quad \mathcal{L}^{(\lambda)}(\psi; \theta, \omega, \psi_{old}) = \mathcal{L}_3(u_\theta, q_\omega, \lambda_{\psi_{old}}, \lambda_\psi),$$

where

$$\mathcal{L}_1(u, q, \lambda) = \frac{1}{\rho_Y}\mathbb{E}_Y\left[u(T, Y)m_T(Y) - u(0, Y)m_0(Y)\right] + \frac{1}{\rho_X}\mathbb{E}_X\left[\frac{r}{2}\|\Lambda u(X) - q(X)\|^2 - \langle \Lambda u(X), \lambda(X) \rangle\right] \tag{25}$$

$$\mathcal{L}_2(u, q, \lambda) = \frac{1}{\rho_X}\mathbb{E}_X\left[\mathcal{G}(q(X)) + \langle \lambda(X), q(X) \rangle + \frac{r}{2}\|\Lambda u(X) - q(X)\|^2\right]$$

$$\mathcal{L}_3(u, q, \lambda_{old}, \lambda) = \frac{1}{\rho_X}\mathbb{E}_X[\|\lambda_{old}(X) - r\left(\Lambda u(X) - q(X)\right) - \lambda(X)\|^2]. \tag{26}$$

Here, the subscript old is used to refer to the previous iteration: the loss for λ involves the previous estimate λ_{old}. When using a neural network, it amounts to using the previous neural network parameters ψ_{old}. The loss function aims at mimicking the effect of the direct update in the third step of standard ADMM (Algorithm 1) when λ is approximated by a neural network.

The algorithm DeepADMM is presented in Algorithm 2.

Algorithm 2: DeepADMM for MFOT

Data: Initial parameter $\theta^{(0)}, \omega^{(0)}, \psi^{(0)}$; number of ADMM iterations K; SGD parameters
Result: Final parameter $\theta^{(K)}, \omega^{(K)}, \psi^{(K)}$

begin
  for $k = 1, \cdots, K$ **do**
    Compute $\theta^{(k)}$ using SGD to (approximately) minimize the loss $\mathcal{L}^{(u)}(\cdot; \omega^{(k-1)}, \psi^{(k-1)})$
    Compute $\omega^{(k)}$ using SGD to (approximately) minimize the loss $\mathcal{L}^{(q)}(\cdot; \theta^{(k)}, \psi^{(k-1)})$
    Compute $\psi^{(k)}$ using SGD to (approximately) minimize the loss $\mathcal{L}^{(\lambda)}(\cdot; \theta^{(k)}, \omega^{(k)}, \psi^{(k-1)})$

We have several remarks regarding DeepADMM and the augmented Lagrangian formulation in order for readers to better understand this approach. First, compared with Algorithm 1, the updates in Algorithm 2 for functions u and q are quite straightforward to understand. Instead of searching optimizer over function space, we reduce the problem to a finite dimension through stochastic approximation of the objective and search in the parameter space instead. The computed stochastic gradient can be considered as an unbiased estimator of the population gradient with respect to the functional, and the variance of this stochastic gradient decreases as the batch size increases.

In Appendix B, we discuss the computation of $\mathcal{G}$ for several typical models.

4. Numerical experiments

In this section, we present numerical experiments obtained with the three methods discussed in the previous section. For brevity, we refer to the three methods respectively introduced in sections 3.1, 3.2 and 3.3 as Method 1, Method 2 and Method 3 (and M1, M2, and M3 for short in the plots).

We first consider two test cases for which we have explicit solutions (up to solving ODE systems) and can thus be used to benchmark our algorithms in any dimension. We then consider two test cases that can be viewed as modifications of standard OT with crowd aversion or congestion effects.

4.1. Case 1: Linear Quadratic Problem

The first class of models that we consider has a linear-quadratic structure, which falls in the setting discussed in Example 2.

4.1.1. *Description of the problem*

In this model, we take:

$$b(x, \mu, a) = Ax + Ba, \qquad f(x, \mu, a) = a^\top Ra, \qquad \rho_0 = \mathcal{N}(\bar{x}_0; \Sigma_0), \qquad \rho_T = \mathcal{N}(\bar{x}_T; \Sigma_T),$$

where $A, B, R, \Sigma_0, \Sigma_T$ are (constant) matrices of suitable sizes. The vectors $\bar{x}_0$ and $\bar{x}_T$ correspond to the initial and terminal means. We will consider two settings. In order to have a benchmark solution, we will take $\sigma = B$. This enables us to use the solution provided by [30, Section 7.1], which boils down to solving a system of ODEs. For the sake of completeness, we provide the details in Appendix A.

4.1.2. *Evaluation Metrics*

In this model, since we have access to the optimal solution, we can evaluate the learnt solutions given by the three methods we proposed with respect to the ground-truth solution. We denote by v^* the optimal control and $\hat{v}$ a learnt control. As explained below in detail, we use the following metrics: the total cost (namely $J^{MFOT}(\hat{v})$, with J^{MFOT} introduced in (3)), the relative error between the achieved cost, and the optimal cost (namely $J^{MFOT}(\hat{v})$ and $J^{MFOT}(v^*)$), the deviation from the terminal distribution (i.e., the Wasserstein distance between the achieved terminal distribution and the target terminal distribution, ρ_T), and the weighted L^2 error between the learnt control $\hat{v}$ and the optimal control v^*, weighted by the population distribution.

Computation of the control. The control is parameterized in different ways across different methods. For Method 1, $\hat{v}(t,x) = v_\theta(t,x)$. For Method 2 and Method 3, $\hat{v}(t,x) = -\frac{1}{2}BR^{-1}\nabla\hat{u}_\theta(t,x)$, where $\hat{u}_\theta$ is the neural network that approximates the dual variable, solution to the HJB equation.

Total cost. Recall the definition of the objective J^{MFOT} defined in (1). Let v be a control. In the present Linear-Quadratic case, we have that,

$$J^{MFOT}(v) = \int_0^T \int_{\mathcal{Q}} m^v(t,x) f(x, m^v, v) \mathrm{d}x\, \mathrm{d}t = \int_0^T \int_{\mathcal{Q}} m^v(t,x) v(t,x)^\top R v(t,x) \mathrm{d}x\, \mathrm{d}t,$$

where m^v is the density of mean field distribution driven by v, which satisfies the KFP PDE (12). In order to evaluate $J^{MFOT}(v)$, we use Monte Carlo simulations. We discretize the time variable t. Let N_T be a number of time steps of length $\Delta t = T/N_T$. We consider an equi-distanced time discretization with time-steps $\{t_0 = 0, t_1 = \Delta t, \dots, t_{N_T} = N_T \Delta t\}$. Again, we simulate solutions to the underlying SDE using an Euler-Maruyama scheme similar to the one used in (7). We simulate a family of N sequences $((X^{i,v}_{t_n})_{n=0,\dots,N_T})_{i=1,\cdots,N}$ using the following update

$$\begin{cases} X^{i,v}_0 \sim \rho_0 \quad \text{i.i.d.} \\ X^{i,v}_{t_{n+1}} = X^{i,v}_{t_n} + (AX^{i,v}_{t_n} + Bv(t_n, X^{i,v}_{t_n}))\Delta t + \sigma\sqrt{\Delta t}\Delta W^i_n, \end{cases} \tag{27}$$

where $\{\Delta W^i_n\}$ are i.i.d standard Gaussian random variables in $\mathbb{R}^d$. With these sampled sequences, we compute the objective as

$$J^{MFOT}(v) = \frac{1}{N}\sum_{i=1}^{N}\sum_{n=0}^{N_T-1} v(t_n, X^{i,v}_{t_n})^\top R v(t_n, X^{i,v}_{t_n})\Delta t.$$

Relative Error. The relative error between $J^{MFOT}(\hat{v})$ and $J^{MFOT}(v^*)$ is defined as

$$\frac{|J^{MFOT}(\hat{v}) - J^{MFOT}(v^*)|}{|J^{MFOT}(v^*)|}.$$

The major reason to consider relative error instead of absolute error is because the scale of the running cost varies greatly across different problems. Also, we want to stress that even though v^* is the analytical optimal solution, it may happen that $J^{MFOT}(\hat{v}) < J^{MFOT}(v^*)$ if $\hat{v}$ does not satisfy exactly the constraint (in contrast with v^*).

Expected L^2 error for control. The expected L^2 error between the learnt control $\hat{v}$ and the ground-truth control v^* is defined as,

$$d_{L^2}(\hat{v}, v^*) = \int_0^T \int_{\mathcal{Q}} m^*(t,x) \|\hat{v}(t,x) - v^*(t,x)\|^2 \mathrm{d}x\, \mathrm{d}t$$

where m^* is the density of the optimal mean-field associated with v^*. For the LQ problem, the optimal mean field m^* is Gaussian for any $t \in [0,T]$, with mean μ_t and variance Σ_t given by analytical formulas in Appendix A. We can thus

Test	d	A	B	σ	R	$\bar{x}_0$	Σ_0	$\bar{x}_T$	Σ_T
LQ Test 1	1	1	1	1	$\frac{1}{2}$	0.0	1	2.0	0.5
LQ Test 2	2	I_d	I_d	I_d	$\frac{1}{2}I_d$	$[0.0, 0.0]$	I_d	$[2.0, 2.0]$	$\frac{1}{2}I_d$

TABLE 1. Parameters for the two linear-quadratic test cases. For LQ Test 2, $\bar{x}_0$ and $\bar{x}_T$ are 2-dimensional vectors with all coordinates equal to 0 or 2 respectively, and I_d denotes the 2×2 identity matrix.

evaluate the L^2 error again with Monte Carlo samples for each time step. As above, we discretize the time variable t with $N_T + 1$ points and for $t_n = n\Delta t, n = 0, \dots, N_T$, we generate i.i.d. samples $(X^{i,v}_{t_n})_{i=1,\dots,N} \sim \mathcal{N}(\mu_{t_n}, \Sigma_{t_n})$. We then estimate the L^2 error as:

$$d_{L^2}(\hat{v}, v^*) \approx \frac{1}{N} \sum_{i=1}^{N} \sum_{n=0}^{N_T-1} \|\hat{v}(t_n, X^{i,v}_{t_n}) - v^*(t_n, X^{i,v}_{t_n})\|^2 \Delta t.$$

Deviation of distribution. The deviation of the mean field $\hat{\rho}_T$ from the terminal target distribution ρ_T is quantified by two different metrics: Wasserstein-2 distance $\mathcal{W}_2(\hat{\rho}_T, \rho_T)$ and L^2 distance $d_{L^2}(\hat{m}_T, m_T)$. Here, $\hat{\rho}_T$ is the measure of the mean field distribution driven by the learnt control $\hat{v}$ at time T. $\hat{m}_T$ and m_T are the density of $\hat{\rho}_T$ and ρ_T respectively.

- **Wasserstein-2 distance.** We adopt a similar method to compute the Wasserstein-2 distance as the one discussed in Section 3.1. We simulate N particles following the dynamics (27), and obtain a collection of N samples $(X^{i,\hat{v}}_T)_{i=1,\dots,N}$, which forms an empirical distribution approximating $\hat{\rho}_T$. We also generate N samples directly from the target distribution ρ_T, denoted by $(Y^i_T)_{i=1,\dots,N}$. Then, we define the distance matrix $\mathcal{M}$ by $\mathcal{M}_{ij} = |X^{i,\hat{v}}_T - Y^j_T|^2$, and we recall that the set U_N is defined by (8). We approximate the Wasserstein-2 distance between the two empirical distributions formed by $X^{i,\hat{v}}_T$ and Y^i_T through the following linear program:

$$\mathcal{W}_2(\hat{\rho}_T, \rho_T) \approx \left(\min_{\mathcal{A} \in U_N} \langle \mathcal{A}, M \rangle\right)^{1/2}.$$

- L^2 **distance.** We will also use as a metric the L^2 distance between $\hat{m}_T$ and m_T on the truncated domain $\tilde{\mathcal{Q}}$. To evaluate the integral, in the absence of analytical formula for m_T, we again use a Monte Carlo approach. We uniformly sample N points in $\tilde{\mathcal{Q}}$ denoted by $(X_j)_{j=1,\dots,N}$. Let $C(\tilde{\mathcal{Q}})$ denotes the inverse of the value of the uniform density. Then we can approximate the L^2 distance by:

$$\frac{C(\tilde{\mathcal{Q}})}{N} \sum_{i=1}^{N} \|\hat{m}_T(X_i) - m_T(X_i)\|^2.$$

4.1.3. *Numerical results*

In the numerical tests, we take the values given in Table 1 for the parameters of the model, with the time horizon $T = 1.0$.

For LQ test 1, in dimension 1, Figure 1 displays the evolution of the density. For methods 1, 2, and 3, we obtain the control learnt using the neural networks and simulate N trajectories by the Monte Carlo method following the dynamics (27), with v replaced by the learnt control. We then estimate the mean field distribution using kernel density estimation (KDE). We see that the distributions obtained with the three methods match well the ground-truth one obtained with ODEs. The distributions move towards the right and concentrate around the final mean. Figure 2 shows the evolution of the control. We see that the three methods provide good approximations of the true optimal control,

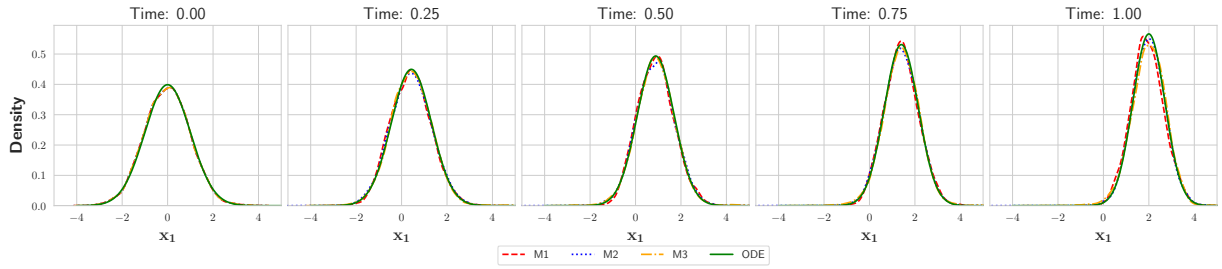


FIGURE 1. Evolution of the density in the LQ Test case 1. Each plot corresponds to one time step and displays the densities as functions of the space variable, x. The densities are: The density obtained by applying the control learnt by each of the three deep learning methods as well as the ground-truth density given by the ODE method.

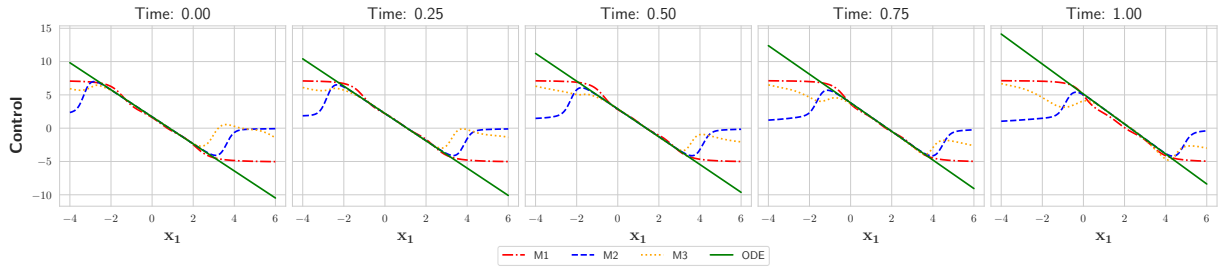


FIGURE 2. Evolution of the control in the LQ Test case 1. Each plot corresponds to one time step and displays the controls as functions of the space variable, x. The controls are: The control learnt by each of the three deep learning methods as well as the ground-truth control given by the ODE method.

at least in the region where the density is high. In regions where the density is very low, the control is not well approximated, but this is not an issue as far as the optimal behavior of the population is concerned. The first part of Table 2 shows the results obtained for the metrics introduced above. We see that some of the methods achieve a smaller total cost than the true optimal control. This would not be possible for controls satisfying perfectly the terminal constraint, but it is possible here due to the fact that the methods satisfy only approximately the planning constraint. This also explains why the relative errors on the total cost are of the order of a few percents. We expect that it would be possible to reduce the relative errors by further fine-tuning the hyperparameters. Moreover, the optimal control is well approximated, as shown by the L^2 distance to the true optimal control. Furthermore, we see that Methods 2 and 3 have a higher Wasserstein-2 distance between the terminal distribution and the target distribution, but the L^2 distance is much lower.

As for LQ test 2, in dimension 2, Figure 3 displays the evolution of the density for each of the methods. The densities move from the bottom left corner to the top right corner. Furthermore, the terminal distribution is more concentrated because the terminal variance is smaller than the initial variance. Figure 4 shows the evolution of the first dimension of the control (the second dimension is similar, so we omit it for brevity). The ground-truth control is linear in space for each time step. We see that the three methods manage to learn approximately linear controls, at least in the region where the density is significantly positive. Table 3 shows the results obtained for the metrics introduced above. We see that here again, each of the three methods achieves a smaller total cost than the true optimal control due to the fact that the terminal constraint is not perfectly satisfied. The optimal control is well approximated, and the terminal distribution is matched with good accuracy.

Test case ($\mu \pm \sigma$)	Method	Total Cost	Relative Error	$d_{L^2}(\hat{v}, v^*)(\times 10^{-2})$	$\mathcal{W}_2(\hat{\rho}_T, \rho_T)(\times 10^{-2})$	$d_{L^2}(\hat{m}_T, m_T)(\times 10^{-3})$
	ODE (v^*)	2.112 ± 0.001	-	-	-	-
Linear Quadratic	M1	2.223 ± 0.001	5.25%	2.701 ± 0.872	0.6622 ± 0.388	1.772 ± 1.101
LQ Test 1	M2	2.088 ± 0.003	1.13%	0.038 ± 0.024	7.119 ± 3.643	0.0003 ± 0.0002
	M3	1.998 ± 0.056	5.39%	4.461 ± 1.261	9.763 ± 7.542	3.826 ± 5.044

TABLE 2. Comparison of three different methods v.s. the analytical solution on the LQ test case 1. The evaluation metrics are described in Section 4.1.2.

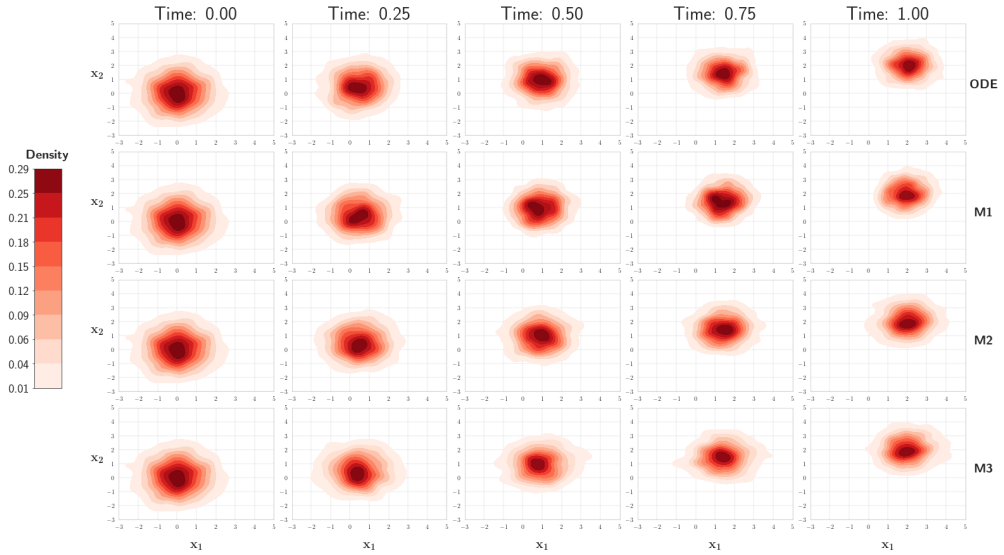


FIGURE 3. Evolution of the density in the Linear Quadratic Test case 2. Each column corresponds to one time step, and each row corresponds to one of the methods. Each plot displays the density as a function of the space variable, i.e., $(m(t,x)_1)_{x\in[-4,6]^2}$. The first row corresponds to the solution obtained by the ground-truth ODE method. The second, third and fourth rows correspond respectively to methods 1, 2 and 3.

Test case ($\mu \pm \sigma$)	Method	Total Cost	Relative Error	$d_{L^2}(\hat{v}, v^*)(\times 10^{-2})$	$\mathcal{W}_2(\hat{\rho}_T, \rho_T)(\times 10^{-2})$	$d_{L^2}(\hat{m}_T, m_T)(\times 10^{-3})$
	ODE (v^*)	4.242 ± 0.039	-	-	-	-
Linear Quadratic	M1	4.338 ± 0.049	2.26%	7.995 ± 1.760	3.192 ± 0.708	2.205 ± 0.567
LQ Test 2	M2	3.822 ± 0.034	9.90%	5.799 ± 0.329	56.86 ± 17.35	0.013 ± 0.002
	M3	4.079 ± 0.083	3.84%	9.348 ± 1.739	52.40 ± 11.24	7.897 ± 2.648

TABLE 3. Comparison of three different methods v.s. the analytical solution on the LQ test case 2. The evaluation metrics are described in Section 4.1.2.

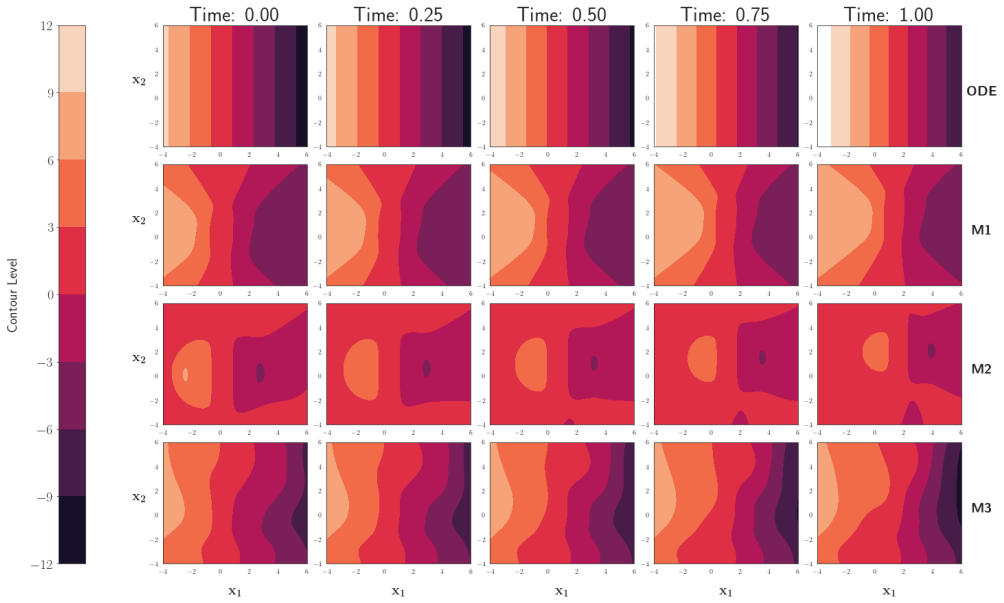


FIGURE 4. Evolution of the control in the Linear Quadratic Test case 2. Each column corresponds to one time step, and each row corresponds to one of the methods. Each plot displays the first dimension of the control as a function of the space variable, i.e., $(v(t,x)_1)_{x\in[-4,6]^2}$. The first row corresponds to the solution obtained by the ground-truth ODE method. The second, third and fourth rows correspond respectively to methods 1, 2 and 3.

4.2. **Case 2: Transport with congestion effects**

The second class of models that we consider is inspired by crowd motion and falls in the setting discussed in Example 3.

4.2.1. *Description of the problem*

In this model, intuitively, the cost is higher when moving through a crowded region, i.e., where the density is high. Specifically, we take:

$$b(x,\mu,a)=a, \qquad f(x,\mu,a)=R|\ell(x,\mu)|^\gamma|a|^2, \qquad \rho_0=\mathcal{N}(\bar{x}_0;\Sigma_0), \qquad \rho_T=\mathcal{N}(\bar{x}_T;\Sigma_T).$$

For the function ℓ, we take two different models. We consider the following non-local dependence:

$$\ell(x,\mu)=c+\rho_\epsilon\star\mu(x),$$

where $c>0$ is a constant, ρ_ϵ is a Gaussian kernel and $\star$ denotes the convolution. We use Method 1 to solve the MFOT problem with this function ℓ. Since it is based on Monte Carlo simulations of trajectories, it is straightforward to compute a convolution with the empirical distribution at a given time step.

We also consider a variation with a local dependence.

$$\ell(x,\mu)=c+m(x)$$

where $c>0$ is a constant and m denotes the density of μ. For this type of model, Methods 2 and 3 are better suited since, in these methods, we directly have access to the approximate density in the form of a neural network.

4.2.2. *Numerical results*

We focus on one test case called "Congestion" below in dimension $d = 1$. In this model, $\gamma = 1$. For the sake of comparison, we also consider the corresponding model with the same choice of parameters except that $\gamma = 0$, i.e., there are no congestion effects in the running cost. The values that we take in the numerical tests are given in Table 4 below.

Test case	d	γ	c	σ	R	$\bar{x}_0$	Σ_0	$\bar{x}_T$	Σ_T
Case 1, No congestion	1	0	0.1	0.1	0.5	0	0.04	2	0.04
Case 2, Congestion	1	1	0.1	0.1	0.5	0	0.04	2	0.04
Case 3, Congestion	5	1	1	I_d	0.5	$[0, \dots, 0]$	$0.1\, I_d$	$[2, \dots, 2]$	$0.1\, I_d$

TABLE 4. Parameters for the test case with congestion and the benchmark model without congestion effects. For Case 3, $\bar{x}_0$ and $\bar{x}_T$ are 5-dimensional vectors with all coordinates equal to 0 or 2 respectively, and I_d denotes the 5×5 identity matrix.

In Figure 5, we present the evolution of the density under the control learnt by each of the three methods for congestion cases 1 and 2. Each row corresponds to one method. We see that, in the case where $\gamma = 0$ (no congestion effect), the mass is transported directly towards the terminal distribution without much change in its shape. In contrast, in the case with $\gamma = 1$, the mass spreads in space and one part starts moving towards the target mean $\bar{x}_T = 2$ whereas another part stays behind and catches up at later time steps. This is consistent with the idea that moving in congested regions is more expensive, so some agents would agree to wait until the density decreases before moving forward.

Finally, in Figure 6, we present the evolution of the density under the control learnt by each of the three methods for congestion case 3, which is in dimension 5. Each row corresponds to one method. To visualize density evolution in dimension 5, we plot the marginal distribution of the mean field distribution on the first and second dimensions. We see that, similarly to the congestion case 2, the mass spreads in space and gradually moves towards the target mean. Compared with congestion case 2, the difference in the moving pattern and extent of spreading is due to the difference of parameters in Table 4. With a larger value c, the behavior of the density would be closer to a direct transport to the terminal distribution without changes in the shape of the distribution.

4.3. **Remarks on the choice of hyperparameters**

Each method has several hyperparameters, including the architecture of the neural networks. We provide below some remarks about the choice of hyperparameters in our implementation.

Method 1. In our implementation, we choose $G(r) = C_W r$ where C_W is a hyperparameter that we adjust dynamically. We increase the constant C_W when we expect a higher running cost (for instance, in a higher dimension) in order to give enough importance to the penalty. The coefficient α of regularization for the computation of the Wasserstein distance is also a hyperparameter that we adjust dynamically using the following heuristics. We start with a given value for α and, when the estimated Wasserstein distance is small enough, we reduce the value of α. The idea is that, as long as the terminal distribution does not match well enough the target distribution, we need a high level of regularization in order to estimate efficiently the Wasserstein distance between them. As the two distributions get closer, we can decrease the degree of regularization in order to have a more accurate estimation of the Wasserstein distance. The way we adjust C_W also depends on the dimension of the state variable. There is also a computational time aspect to take into account: as α becomes smaller, the computations take more time (see 3.1.2 for more details). For the neural network, we take a feedforward fully connected neural network with 6 layers of 60 neurons each. The other hyperparameters are the number of particles N and the number of time steps N_T. We take $N = 300$ and $N_T = 20$.

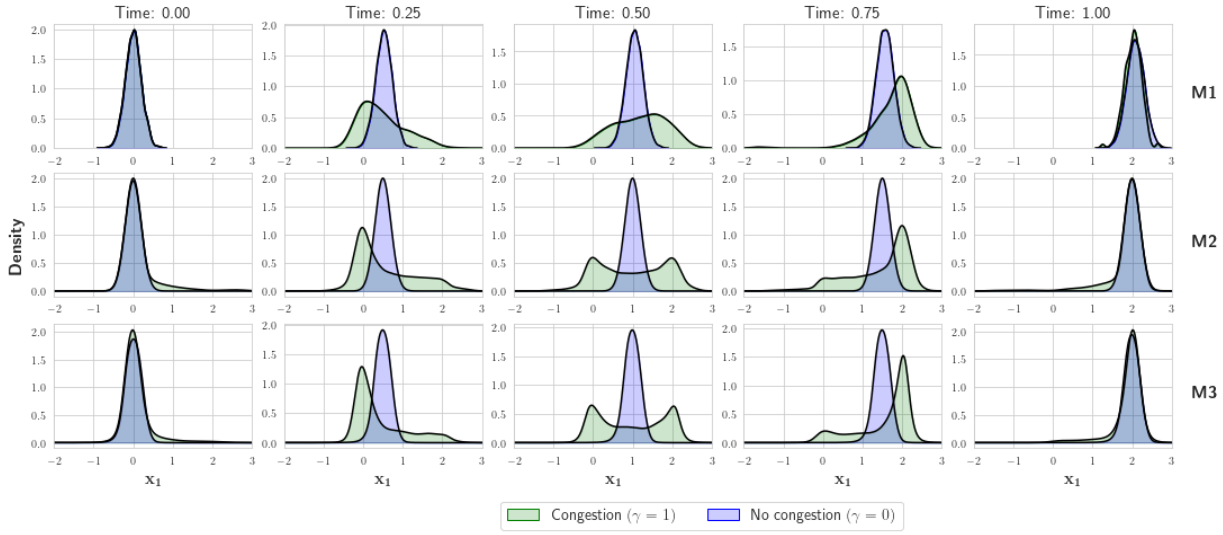


FIGURE 5. Visualization of the mean field density $\hat{m}(t, x)$ in Congestion test Case 1,2

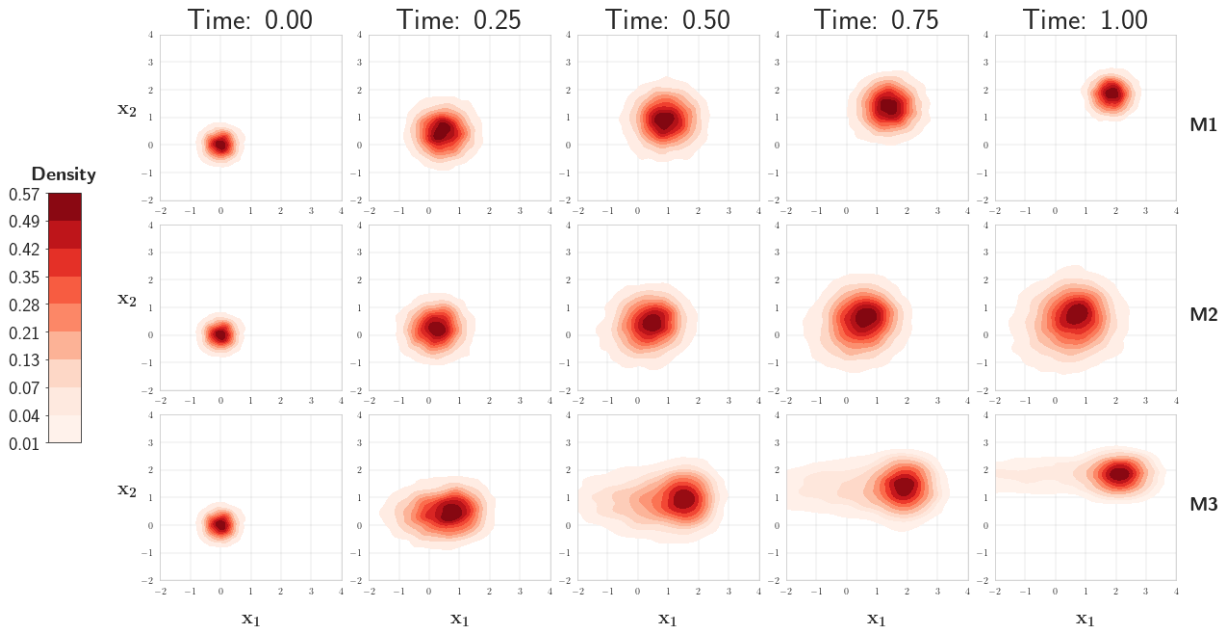


FIGURE 6. Visualization of the mean field density $\hat{m}(t, x)$ in Congestion test Case 3

Method 2. In the second method, no time or space discretization is needed, and the density is directly approximated by a neural network, so we do not need to use a finite number of particles. However, we need to choose the values of the weights $C_0^{(\mathrm{KFP})}$, $C_T^{(\mathrm{KFP})}$, $C^{(\mathrm{KFP})}$, and $C^{(\mathrm{HJB})}$ in the loss function. We used $C_0^{(\mathrm{KFP})} = 20$, $C_T^{(\mathrm{KFP})} = 50$, $C^{(\mathrm{KFP})} = 20$, $C^{(\mathrm{HJB})} = 1$. As for the neural network, we used the architecture proposed in the DGM article [62], with 2 layers and a width equal to 40. During the training, at each iteration of SGD, we use a minibatch of 500 points in time and space, and 500 points in space for the initial and terminal conditions.

Method 3. The main hyperparameter in this method is r, which is used in the definition of the augmented Lagrangian (24). For the experiments, we select $r = 0.1$. Even though, in theory, the convergence of ADMM is independent of the choice of r, in practice, we often find that a large r value could potentially increase numerical instability and lead the algorithm to diverge. Similarly, a small r value could slow down the convergence. As for the neural networks, we use the following architectures. For both u_θ, q_ω, and λ_ψ, in general, we use a fully connected neural network with residual connections, sigmoid activation function, and appropriate output dimension. We use 6 layers and 100 neurons per layer. For LQ test cases, we further consider an extra quadratic correction in addition to the neural networks: the output of u_θ is the sum of neural network output and a quadratic function with trainable weights. To effectively model the mean field density, a sigmoid activation function is applied to the first dimension of the output of the neural network λ_ψ, and then the result is multiplied by a constant C. In this way, the first dimension of λ_ψ takes values in $(0, C)$. In the experiments, we take $C = 1$ for the LQ test cases and $C = 5$ for the congestion test cases. During training, at each iteration of SGD, we use a minibatch of 512 points in time and space, and 512 points in space for the initial and terminal conditions.

5. Conclusion and future directions

In this work, we have proposed three numerical methods based on deep learning for mean field optimal transport problems. The three methods can tackle a larger class of problems than deep learning methods proposed previously, which were mostly focusing on the Schrödinger bridge problem or MFGs with a specific structure. The first method replaces the terminal constraint with a penalty and then directly learns the optimal control using Monte Carlo trajectories. The second method solves a PDE system which is obtained as the optimality conditions for the MFOT problem. The third method relies on an augmented Lagrangian approach for the variational formulation of the problem. The numerical results show that the three methods match the analytical solution on an LQ problem, and that they are able to handle non-trivial mean field interactions modeling congestion effects.

From here, we can envision several research directions. First of all, the theoretical analysis of the MFOT problem remains to be tackled. For example, the existence and uniqueness of the solution to the PDE system have been proved only in relatively specific cases, see e.g. [1, 20, 36, 55]. It would be interesting to extend the analysis to more general forms of dynamics and cost functions. From the numerical point of view, it would be interesting to scale-up the methods proposed in this work to a higher dimension, and to explore other deep learning methods. The numerical analysis and the convergence proof of the proposed methods also remain to be investigated in future work.

References

1. Yves Achdou, Fabio Camilli, and Italo Capuzzo-Dolcetta, *Mean field games: numerical methods for the planning problem*, SIAM Journal on Control and Optimization **50** (2012), no. 1, 77–109.
2. Yves Achdou and Jean-Michel Lasry, *Mean field games for modeling crowd motion*, Contributions to partial differential equations and applications (2019), 17–42.
3. Yves Achdou and Mathieu Laurière, *On the system of partial differential equations arising in mean field type control*, Discrete and Continuous Dynamical Systems **35** (2015), no. 9, 3879–3900.
4. _____, *Mean field type control with congestion*, Applied Mathematics & Optimization **73** (2016), no. 3, 393–418.
5. _____, *Mean field type control with congestion (II): An augmented Lagrangian method*, Applied Mathematics & Optimization **74** (2016), no. 3, 535–578.
6. Yves Achdou and Alessio Porretta, *Mean field games with congestion*, Annales de l'Institut Henri Poincaré C, Analyse non linéaire, vol. 35, Elsevier, 2018, pp. 443–480.
7. Ali Al-Aradi, Adolfo Correia, Danilo Naiff, Gabriel Jardim, and Yuri Saporito, *Solving nonlinear and high-dimensional partial differential equations via deep learning*, arXiv preprint arXiv:1811.08782 (2018).
8. Luigi Ambrosio, Nicola Gigli, and Giuseppe Savaré, *Gradient flows: in metric spaces and in the space of probability measures*, Springer Science & Business Media, 2005.
9. Roman Andreev, *Preconditioning the augmented Lagrangian method for instationary mean field games with diffusion*, SIAM Journal on Scientific Computing **39** (2017), no. 6, A2763–A2783.
10. Julio Backhoff, Giovanni Conforti, Ivan Gentil, and Christian Léonard, *The mean field Schrödinger problem: ergodic behavior, entropy estimates and functional inequalities*, Probability Theory and Related Fields **178** (2020), no. 1, 475–530.

11. Jean-David Benamou and Yann Brenier, *A computational fluid mechanics solution to the Monge-Kantorovich mass transfer problem*, Numerische Mathematik **84** (2000), no. 3, 375–393.
12. Jean-David Benamou and Guillaume Carlier, *Augmented Lagrangian methods for transport optimization, mean field games and degenerate elliptic equations*, Journal of Optimization Theory and Applications **167** (2015), no. 1, 1–26.
13. Jean-David Benamou, Guillaume Carlier, Simone Di Marino, and Luca Nenna, *An entropy minimization approach to second-order variational mean-field games*, Mathematical Models and Methods in Applied Sciences **29** (2019), no. 08, 1553–1583.
14. Jean-David Benamou, Guillaume Carlier, and Maxime Laborde, *An augmented Lagrangian approach to Wasserstein gradient flows and applications*, ESAIM: Proceedings and surveys **54** (2016), 1–17.
15. Alain Bensoussan, Jens Frehse, Phillip Yam, et al., *Mean field games and mean field type control theory*, vol. 101, Springer, 2013.
16. Alain Bensoussan, Jens Frehse, and Sheung Chi Phillip Yam, *The master equation in mean field theory*, Journal de Mathématiques Pures et Appliquées **103** (2015), no. 6, 1441–1474.
17. Charles Bertucci, Jean-Michel Lasry, and Pierre-Louis Lions, *Master equation for the finite state space planning problem*, Archive for Rational Mechanics and Analysis **242** (2021), no. 1, 327–342.
18. Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al., *Distributed optimization and statistical learning via the alternating direction method of multipliers*, Foundations and Trends® in Machine learning **3** (2011), no. 1, 1–122.
19. Haoyang Cao, Xin Guo, and Mathieu Laurière, *Connecting GANs, MFGs, and OT*, arXiv preprint arXiv:2002.04112 (2020).
20. Pierre Cardaliaguet, Guillaume Carlier, and Bruno Nazaret, *Geodesics for a class of distances in the space of probability measures*, Calculus of Variations and Partial Differential Equations **48** (2013), no. 3, 395–420.
21. Pierre Cardaliaguet and P Jameson Graber, *Mean field games systems of first order*, ESAIM: Control, Optimisation and Calculus of Variations **21** (2015), no. 3, 690–722.
22. René Carmona, François Delarue, et al., *Probabilistic theory of mean field games with applications I-II*, Springer, 2018.
23. René Carmona and Mathieu Laurière, *Convergence analysis of machine learning algorithms for the numerical solution of mean field control and games: II: The finite horizon case*, arXiv preprint arXiv:1908.01613 (2019).
24. _____, *Convergence analysis of machine learning algorithms for the numerical solution of mean field control and games I: the ergodic case*, SIAM Journal on Numerical Analysis **59** (2021), no. 3, 1455–1485.
25. René Carmona and Mathieu Laurière, *Deep learning for mean field games and mean field control with applications to finance*, 2021.
26. Yongxin Chen, Tryphon T Georgiou, and Michele Pavon, *Optimal steering of a linear stochastic system to a final probability distribution, part I*, IEEE Transactions on Automatic Control **61** (2015), no. 5, 1158–1169.
27. _____, *Optimal steering of a linear stochastic system to a final probability distribution, part II*, IEEE Transactions on Automatic Control **61** (2015), no. 5, 1170–1180.
28. _____, *On the relation between optimal transport and schrödinger bridges: A stochastic control viewpoint*, Journal of Optimization Theory and Applications **169** (2016), no. 2, 671–691.
29. _____, *Optimal steering of a linear stochastic system to a final probability distribution—part III*, IEEE Transactions on Automatic Control **63** (2018), no. 9, 3112–3118.
30. _____, *Steering the distribution of agents in mean-field games system*, Journal of Optimization Theory and Applications **179** (2018), no. 1, 332–357.
31. Marco Cuturi, *Sinkhorn distances: Lightspeed computation of optimal transport*, Advances in neural information processing systems **26** (2013).
32. Jean Feydy, Thibault Séjourné, François-Xavier Vialard, Shun-ichi Amari, Alain Trouvé, and Gabriel Peyré, *Interpolating between optimal transport and mmd using sinkhorn divergences*, The 22nd International Conference on Artificial Intelligence and Statistics, PMLR, 2019, pp. 2681–2690.
33. Jean-Pierre Fouque and Zhaoyu Zhang, *Deep learning methods for mean field control problems with delay*, Frontiers in Applied Mathematics and Statistics **6** (2020), 11.
34. Maximilien Germain, Joseph Mikael, and Xavier Warin, *Numerical resolution of McKean-Vlasov FBSDEs using neural networks*, Methodology and Computing in Applied Probability (2022), 1–30.
35. Maximilien Germain, Huyên Pham, and Xavier Warin, *Neural networks-based algorithms for stochastic control and PDEs in finance*, arXiv preprint arXiv:2101.08068 (2021).
36. P Jameson Graber, Alpár R Mészáros, Francisco J Silva, and Daniela Tonon, *The planning problem in mean field games as regularized mass transport*, Calculus of Variations and Partial Differential Equations **58** (2019), no. 3, 1–28.
37. Jiequn Han, Ruimeng Hu, and Jihao Long, *Learning high-dimensional McKean-Vlasov forward-backward stochastic differential equations with general distribution dependence*, arXiv preprint arXiv:2204.11924 (2022).
38. Jiequn Han, Arnulf Jentzen, and Weinan E, *Solving high-dimensional partial differential equations using deep learning*, Proceedings of the National Academy of Sciences **115** (2018), no. 34, 8505–8510.
39. Jiequn Han, Arnulf Jentzen, et al., *Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations*, Communications in mathematics and statistics **5** (2017), no. 4, 349–380.
40. Camilo Hernández and Ludovic Tangpi, *Propagation of chaos for mean field schr\" odinger problems*, arXiv preprint arXiv:2304.09340 (2023).
41. Ruimeng Hu and Mathieu Laurière, *Recent developments in machine learning methods for stochastic control and games*, Preprint. SSRN:4096569 (2022).
42. Minyi Huang, Peter E Caines, and Roland P Malhamé, *Large-population cost-coupled lqg problems with nonuniform agents: individual-mass behavior and decentralized ε-Nash equilibria*, IEEE transactions on automatic control **52** (2007), no. 9, 1560–1571.

43. Minyi Huang, Roland P Malhamé, and Peter E Caines, *Large population stochastic dynamic games: closed-loop McKean-Vlasov systems and the Nash certainty equivalence principle*, Communications in Information & Systems **6** (2006), no. 3, 221–252.
44. Jean-Michel Lasry and Pierre-Louis Lions, *Jeux à champ moyen. I–le cas stationnaire*, Comptes Rendus Mathématique **343** (2006), no. 9, 619–625.
45. ———, *Jeux à champ moyen. II–Horizon fini et contrôle optimal*, Comptes Rendus Mathématique **343** (2006), no. 10, 679–684.
46. ———, *Mean field games*, Japanese journal of mathematics **2** (2007), no. 1, 229–260.
47. Mathieu Laurière and Olivier Pironneau, *Dynamic programming for mean-field type control*, Comptes Rendus Mathematique **352** (2014), no. 9, 707–713.
48. Alex Tong Lin, Samy Wu Fung, Wuchen Li, Levon Nurbekyan, and Stanley J Osher, *Apac-net: Alternating the population and agent control via two neural networks to solve high-dimensional stochastic mean field games*, arXiv preprint arXiv:2002.10113 (2020).
49. Guan-Horng Liu, Tianrong Chen, Oswin So, and Evangelos Theodorou, *Deep generalized schrödinger bridge*, Advances in Neural Information Processing Systems (Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, eds.), 2022.
50. Gonzalo Mena and Jonathan Niles-Weed, *Statistical bounds for entropic optimal transport: sample complexity and the central limit theorem*, Advances in neural information processing systems **32** (2019).
51. Quentin Merigot and Boris Thibert, *Optimal transport: discretization and algorithms*, Handbook of Numerical Analysis, vol. 22, Elsevier, 2021, pp. 133–212.
52. Toshio Mikami, *Stochastic optimal transport revisited*, SN Partial Differential Equations and Applications **2** (2021), no. 1, 5.
53. Carlo Orrieri, Alessio Porretta, and Giuseppe Savaré, *A variational approach to the mean field planning problem*, Journal of Functional Analysis **277** (2019), no. 6, 1868–1957.
54. Gabriel Peyré, Marco Cuturi, et al., *Computational optimal transport: With applications to data science*, Foundations and Trends® in Machine Learning **11** (2019), no. 5-6, 355–607.
55. Alessio Porretta, *On the planning problem for a class of mean field games*, Comptes Rendus Mathematique **351** (2013), no. 11-12, 457–462.
56. ———, *On the planning problem for the mean field games system*, Dynamic Games and Applications **4** (2014), no. 2, 231–256.
57. Maziar Raissi, Paris Perdikaris, and George E Karniadakis, *Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations*, Journal of Computational physics **378** (2019), 686–707.
58. R Tyrrell Rockafellar, *Convex analysis*, vol. 18, Princeton university press, 1970.
59. Lars Ruthotto, Stanley J Osher, Wuchen Li, Levon Nurbekyan, and Samy Wu Fung, *A machine learning framework for solving high-dimensional mean field game and mean field control problems*, Proceedings of the National Academy of Sciences **117** (2020), no. 17, 9183–9193.
60. Filippo Santambrogio, *Optimal transport for applied mathematicians*, Birkäuser, NY **55** (2015), no. 58-63, 94.
61. Richard Sinkhorn and Paul Knopp, *Concerning nonnegative matrices and doubly stochastic matrices*, Pacific Journal of Mathematics **21** (1967), no. 2, 343–348.
62. Justin Sirignano and Konstantinos Spiliopoulos, *DGM: A deep learning algorithm for solving partial differential equations*, Journal of computational physics **375** (2018), 1339–1364.
63. Xiaolu Tan and Nizar Touzi, *Optimal transportation under controlled stochastic dynamics*, The Annals of Probability **41** (2013), no. 5, 3201.
64. Cédric Villani, *Optimal transport: old and new*, vol. 338, Springer, 2009.
65. ———, *Topics in optimal transportation*, vol. 58, American Mathematical Soc., 2021.

A. SOLUTION FOR LQ PROBLEM

In this section, we provide an explicit solution for the LQ setting considered in Section 4.1. We summarize the analytical solution derived in [30, Section 7.1] and we add the analytical formula for the variance, which is useful to fully describe the evolution of the population distribution. We consider the following setting, which is actually slightly more general than the one used in Section 4.1. For any probability distribution μ admitting a first moment, we use the notation $\bar{\mu} = \int x\mu(dx)$, to be understood coordinate-wise. In this model, we take:

$$b(x,\mu,a) = Ax + \bar{A}\bar{\mu} + Ba, \qquad \sigma = B, \qquad f(x,\mu,a) = \frac{1}{2}a^\top a, \qquad \rho_0 = \mathcal{N}(\bar{x}_0;\Sigma_0), \qquad \rho_T = \mathcal{N}(\bar{x}_T;\Sigma_T).$$

We denote by v^* the optimal control and by $(\mu_t)_{t\in[0,T]}$ the optimal flow of distributions. Since the dynamics are linear and the initial and terminal distributions are Gaussians, it can be shown that the distribution remains Gaussian at every t. At each t, we denote by $\bar{\mu}_t$ its mean and by Σ_t its covariance matrix. In the case without mean field interactions, the optimal control is linear in space, and the coefficient depends on the solution of a Ricatti equation in time. The mean field interaction being uniform in space and the fact that our problem is linear explains why we can set

$$v^*(t,x) = -B^\top \Pi_t x + B^\top n_t$$

and solve for $(\Pi_t)_{t\in[0,T]}$ and $(n_t)_{t\in[0,T]}$, as time-dependent functions only. Note that $\bar{\mu}$ is now solution of the following ODE depending on n_t:

$$\dot{\bar{\mu}}_t = (A + \bar{A} - BB^\top \Pi_t)\bar{\mu}_t + BB^\top n_t, \quad \bar{\mu}_0 = \bar{x}_0. \tag{28}$$

Furthermore, if we apply Itô's formula to $V_t = \mathbb{E}[X_t X_t^\top] = \Sigma_t + \bar{\mu}_t\bar{\mu}_t^\top$, we also obtain a matrix ODE depending on $(n_t)_{t\in[0,T]}$ for $(V_t)_{t\in[0,T]}$:

$$\begin{cases} \dot{V}_t = V_t(A^\top - \Pi_t^\top BB^\top) + (A - BB^\top\Pi_t)V_t + \bar{\mu}_t n_t^\top BB^\top + BB^\top n_t\bar{\mu}_t^\top + \bar{\mu}_t\bar{\mu}_t^\top \bar{A}^\top + \bar{A}\bar{\mu}_t\bar{\mu}_t^\top + BB^\top \\ V_0 = \Sigma_0 + \bar{\mu}_0\bar{\mu}_0^\top. \end{cases} \tag{29}$$

We will also need the state transition matrices associated to A and $A + \bar{A}$: we define Φ as the solution of

$$\frac{\partial \Phi(t,s)}{\partial t} = A\Phi(t,s), \quad \Phi(s,s) = I, \quad 0 \le s \le t \le T.$$

Since A is constant, we have $\Phi(t,s) = e^{(t-s)A}$. Similarly, we define $\bar{\Phi}(t,s) = e^{(t-s)(A+\bar{A})}$. We also set

$$\begin{cases} M(t,s) = \displaystyle\int_s^t \Phi(t,\tau)BB^\top\Phi(t,\tau)^\top \mathrm{d}\tau, \quad 0 \le s \le t \le T, \\ \bar{M}(t,s) = \displaystyle\int_s^t \bar{\Phi}(t,\tau)BB^\top\bar{\Phi}(t,\tau)^\top d\tau, \quad 0 \le s \le t \le T, \end{cases} \tag{30}$$

and denote for brevity $\Phi_{T,0} = \Phi(T,0)$, $\bar{\Phi}_{T,0} = \bar{\Phi}(T,0)$, $M_{T,0} = M(T,0)$, and $\bar{M}_{T,0} = \bar{M}(T,0)$. Notice that if $BB^\top$ commutes with Φ and $\bar{\Phi}$, and if the matrices $A + A^\top$ and $A + \bar{A} + A^\top + \bar{A}^\top$ are non-singular, then we have the following closed form expressions for $M(t,s)$ and $\bar{M}(t,s)$, with $0 \le s \le t \le T$:

$$M(t,s) = BB^\top \int_s^t e^{A+A^\top}(t-\tau)\mathrm{d}\tau = BB^\top[e^{(A+A^\top)t} - e^{(A+A^\top)s}](A + A^\top)^{-1}$$

$$\bar{M}(t,s) = BB^\top \int_s^t e^{A+A^\top+\bar{A}+\bar{A}^\top}(t-\tau)\mathrm{d}\tau = BB^\top[e^{(A+A^\top+\bar{A}+\bar{A}^\top)t} - e^{(A+A^\top+\bar{A}+\bar{A}^\top)s}](A + A^\top + \bar{A} + \bar{A}^\top)^{-1}.$$

Now, $(\Pi_t)_{t\in[0,T]}$ is the solution to the following Riccati ODE, which is independent from the other variables:

$$\begin{cases} \dot{\Pi}_t = -A^\top \Pi_t - \Pi_t A + \Pi_t B B^\top \Pi_t \\ \Pi_0 = \Sigma_0^{-1/2}\left[\frac{I}{2} + \Sigma_0^{1/2}\Phi_{T,0}^\top M_{T,0}^{-1}\Phi_{T,0}\Sigma_0^{1/2} - \left(\frac{I}{4} + \Sigma_0^{1/2}\Phi_{T,0}^\top M_{T,0}^{-1}\Sigma_T M_{T,0}^{-1}\Phi_{T,0}\Sigma_0^{1/2}\right)^{1/2}\right]\Sigma_0^{-1/2}. \end{cases} \tag{31}$$

Under some conditions, Riccati equations admit explicit solutions in dimension 1, see e.g. page 110 in [22]. More generally, we can solve (31) using a forward time-marching method.

We can then show that

$$n_t = \Pi_t \bar{\Phi}(t,T)\bar{M}(T,t)\bar{M}_{T,0}^{-1}\bar{\Phi}_{T,0}z_0 + \Pi_t \bar{M}(T,0)\bar{\Phi}(T,t)^\top \bar{M}_{T,0}^{-1} z_T + \bar{\Phi}(T,t)^\top \bar{M}_{T,0}^{-1}(z_T - \bar{\Phi}_{T,0}z_0), \tag{32}$$

where $(z_t)_{t\in[0,T]}$ is defined by

$$z_t = \bar{\Phi}(T,t)^\top \bar{M}_{T,0}^{-1}(\bar{x}_T - \bar{\Phi}_{T,0}\bar{x}_0). \tag{33}$$

Now, up to the computation of Π, we have an explicit formula for n and we can obtain the mean $\bar{\mu}_t$ from (28) and then the covariance matrix Σ_t from (29). The optimal mean field distribution at time t is the Gaussian distribution with mean $\bar{\mu}_t$ and variance Σ_t.

B. Computation of $\mathcal{G}$

In this section, we discuss some of the issues arising when computing $\mathcal{G}$ in practice, as well as our solutions. As the reader may notice, the losses defined in (25) and (26) depend on the exact form of functionals $\mathcal{F}$ and $\mathcal{G}$. As $\mathcal{F}$ is already defined in an explicit, easy-to-compute form, we are left with the problem of figuring out a good approach to compute $\mathcal{G}$. Recalling the definition of $\mathcal{G}$, we have the following observation,

$$\begin{aligned} \mathcal{G}(\mathfrak{a},\mathfrak{b}) &= -\inf_{m\geq 0}\int_{\mathcal{Q}_T} m(t,x)(\mathfrak{a}(t,x) - H(x,m(t,x),\mathfrak{b}(t,x)))\mathrm{d}x\,\mathrm{d}t \\ &= -\int_{\mathcal{Q}_T}\inf_{m\geq 0}[m(\mathfrak{a}(t,x) - H(x,m,\mathfrak{b}(t,x)))]\mathrm{d}x\,\mathrm{d}t \qquad (34) \\ &= -\int_{\mathcal{Q}_T}\mathcal{K}(\mathfrak{a}(t,x),\mathfrak{b}(t,x))\mathrm{d}x\,\mathrm{d}t, \qquad (35) \end{aligned}$$

where we denoted $\mathcal{K}(a,b) = \inf_{m\geq 0}[m(\mathfrak{a} - H(x,m,\mathfrak{b}))]$.

Therefore, the form of $\mathcal{G}$ depends on $\mathcal{K}$. In general, we do not know any closed form of $\mathcal{K}$ in terms of $\mathfrak{a}$ and $\mathfrak{b}$. However, for several Hamiltonian functions $H(x,m,p)$ of interest, we can derive such closed-form solution. Here we demonstrate some of the calculations to deliver a general idea.

Example 4 (Mean-field Aversion)**.** *Consider the running cost $f(m,v) = \frac{1}{4}\|v\|^2 + m$ and drift $b(x,m,v) = v$. The corresponding Hamiltonian is $H(x,m,p) = \|p\|^2 - m$. Then*

$$\mathcal{K}(\mathfrak{a},\mathfrak{b}) = \inf_{m\geq 0}\ m^2 + (\mathfrak{a} - \|\mathfrak{b}\|^2)m = \begin{cases} -\frac{1}{4}(\mathfrak{a} - \|\mathfrak{b}\|^2)^2 & \text{if } \mathfrak{a} - \|\mathfrak{b}\|^2 \geq 0 \\ 0 & \text{otherwise.} \end{cases} \tag{36}$$

Example 5 (Mean-field Maximum Entropy)**.** *Consider the running cost $f(m,v) = \frac{1}{4}\|v\|^2 + \log(m)$ and drift $b(x,m,v) = v$, the corresponding Hamiltonian is $H(x,m,p) = \frac{1}{2}\|p\|^2 - \log(m)$, then*

$$\mathcal{K}(\mathfrak{a},\mathfrak{b}) = \inf_{m\geq 0}\ m\log m + (\mathfrak{a} - \frac{1}{2}\|\mathfrak{b}\|^2)m = -\exp(\frac{1}{2}\|\mathfrak{b}\|^2 - \mathfrak{a} - 1). \tag{37}$$

Example 6 (Continuous Optimal Transport)**.** *Consider the running cost* $f(m,v) = \frac{1}{2}\|v\|^2$ *and drift* $b(x,m,v) = v$*, the corresponding Hamiltonian is* $H(x,m,p) = \frac{1}{2}\|p\|^2$*, then*

$$\mathcal{K}(\mathfrak{a},\mathfrak{b}) = \inf_{m\geq 0}\ m(\mathfrak{a} - \frac{1}{2}\|\mathfrak{b}\|^2) = \begin{cases} 0 & \text{if } \mathfrak{a} - \frac{1}{2}\|\mathfrak{b}\|^2 \geq 0 \\ -\infty & \text{otherwise.} \end{cases} \tag{38}$$

Example 7 (Mean-field Congestion)**.** *Consider the running cost* $f(m,v) = \frac{1}{4}m\|v\|^2$ *and drift* $b(x,m,v) = v$*, the corresponding Hamiltonian is* $H(x,m,p) = \frac{\|p\|^2}{m}$*, then*

$$\mathcal{K}(\mathfrak{a},\mathfrak{b}) = \inf_{m\geq 0}\ m(\mathfrak{a} - \frac{\|\mathfrak{b}\|^2}{m}) = \begin{cases} -\|\mathfrak{b}\|^2 & \text{if } \mathfrak{a} \geq 0 \\ -\infty & \text{otherwise.} \end{cases} \tag{39}$$

From these examples, it can be seen that $\mathcal{K}$ has a closed form for many smooth Hamiltonian. However, the existence of closed form expressions of $\mathcal{K}$ alone is not enough for making the training loss tractable. In the example of (38) and (39), $\mathcal{K}$ takes values $-\infty$, which makes $\mathcal{G}$ singular and computationally intractable at some points. Moreover, gradient-based training cannot be carried out successfully in the presence of infinite values as well. The presence of $-\infty$ in $\mathcal{K}$ is due to the degeneracy of H in terms of order in m. For cases with singular $\mathcal{K}$ and $\mathcal{G}$, we need an additional trick to tackle this issue.

Recall that in Algorithm 1, the update for function q is given by,

$$\begin{aligned} q^{(k)} &= \operatorname*{arg\,min}_{q:\mathcal{Q}_T\to\mathbb{R}^{d+1}} \mathcal{G}(q) + \langle \lambda^{(k-1)}, q\rangle + \frac{r}{2}\|\Lambda u^{(k)} - q\|^2 \\ &= \operatorname*{arg\,min}_{q:\mathcal{Q}_T\to\mathbb{R}^{d+1}} \int_{\mathcal{Q}_T} \left(-\mathcal{K}(q(t,x)) + \langle \lambda^{(k-1)}(t,x), q(t,x)\rangle + \frac{r}{2}\|\Lambda u^{(k)}(t,x) - q(t,x)\|^2\right) \mathrm{d}x\ \mathrm{d}t \end{aligned} \tag{40}$$

Since we don't have any additional constraint on the value of q, the function $q^{(k)}$ that minimizes the integral in (40) should minimize the integrand point wisely. Therefore, it holds that:

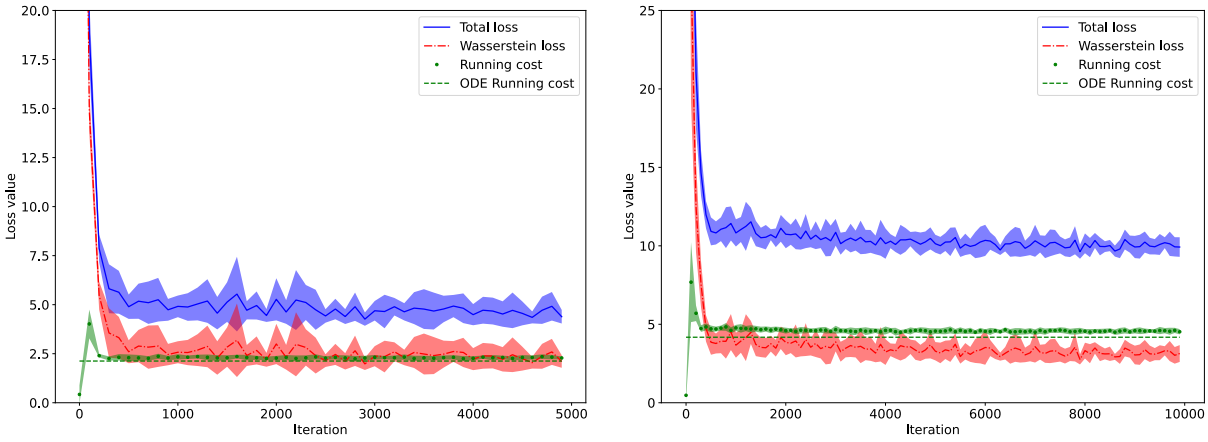
$$\begin{aligned} q^{(k)}(t,x) &= \operatorname*{arg\,min}_{q\in\mathbb{R}^{d+1}} \left(-\mathcal{K}(q) + \langle \lambda^{(k-1)}(t,x), q\rangle + \frac{r}{2}\|\Lambda u^{(k)}(t,x) - q\|^2\right) \\ &= \operatorname*{arg\,min}_{q\in\mathbb{R}^{d+1}} \sup_{m\geq 0}\ m(H(x,m,q_2) - q_1) + \langle \lambda^{(k-1)}(t,x), q\rangle + \frac{r}{2}\|\Lambda u^{(k)}(t,x) - q\|^2, \end{aligned} \tag{41}$$

where we denote $q = (q_1, q_2)$, with $q_1 \in \mathbb{R}$ and $q_2 \in \mathbb{R}^d$. For fixed $\lambda^{(k-1)}$ and $u^{(k)}$, let us define $\mathcal{L}(q,m) = m(H(x,m,q_2) - q_1) + \langle \lambda^{(k-1)}(t,x), q\rangle + \frac{r}{2}\|\Lambda u^{(k)}(t,x) - q\|^2$. Under certain conditions, the following minimax equality holds for the right-hand side of (41),

$$\inf_{q\in\mathbb{R}^{d+1}} \sup_{m\geq 0} \mathcal{L}(q,m) = \sup_{m\geq 0} \inf_{q\in\mathbb{R}^{d+1}} \mathcal{L}(q,m) \tag{42}$$

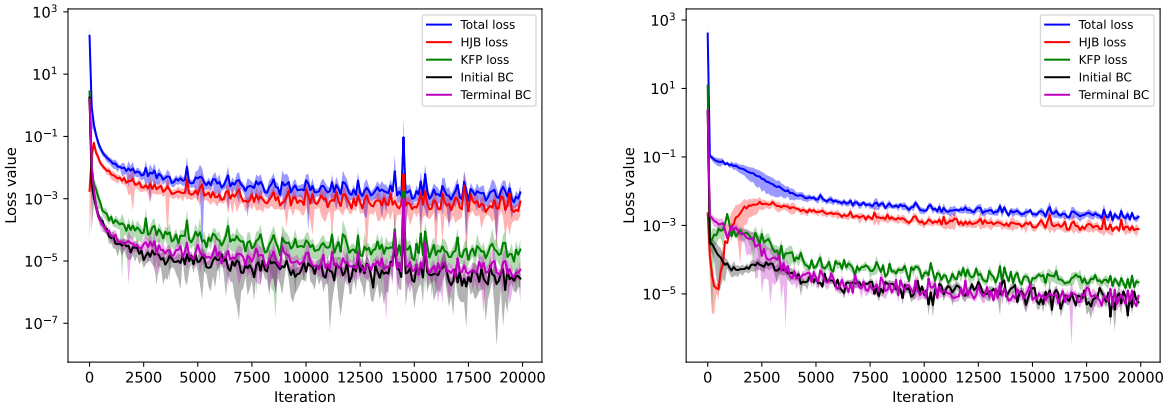
Therefore, we can exploit (42) to address the issues of singular $\mathcal{K}$. We notice that for the cases of congestion and general linear quadratic, $\mathcal{L}(q,m)$ is quadratic in q for fixed m. This means that we can solve explicitly in a closed form for $\inf_q \mathcal{L}(q,m)$ for fixed m, then we solve for the maximization problem over $m \geq 0$. In this way, we can effectively avoid the issues generated by infinity values in $\mathcal{K}$.

Moreover, using this trick, we directly obtain the value for $q^{(k)}$ at any (t,x) based on the value of $u^{(k)}(t,x)$ and $\lambda^{(k-1)}(t,x)$. Therefore, we can skip the neural network training in Algorithm 2 for function q in each DeepADMM iteration. Instead, we compute the values for $q^{(k)}(t,x)$ directly following the above procedure whenever the values are needed to compute $\mathcal{L}^{(u)}$ and $\mathcal{L}^{(\lambda)}$.



(A) LQ Test 1 ($d = 1$) (B) LQ Test 2 ($d = 2$)

FIGURE 7. Evolution of the loss in the linear-quadratic case for method 1.



(A) LQ Test 1 ($d = 1$) (B) LQ Test 2 ($d = 2$)

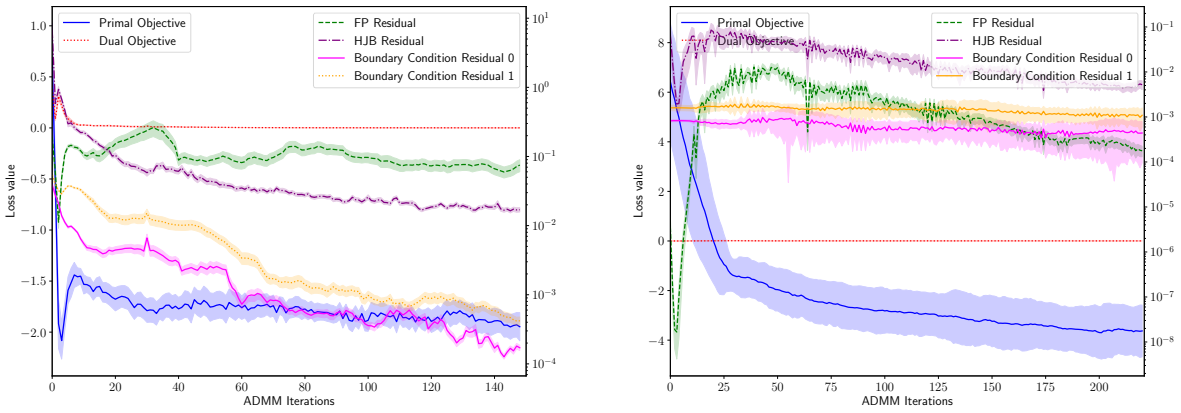
FIGURE 8. Evolution of the loss in the linear-quadratic case for method 2.

C. LOSS PLOTS

In this section, we provide plots for the training losses of the proposed algorithms on each of the examples discussed in the text. For each test case and each of the three methods, we ran the algorithm 10 times and computed the average loss (or average component of the loss, for each component). We also computed the standard deviation over these 10 runs. In each plot, we display the average loss by a line and we represent by a shaded area the region contained between the average minus the standard deviation and the average plus the standard deviation.

We start by presenting the loss plots for the two LQ test cases in Figure 7 for Method 1, Figure 8 for Method 2, and Figure 9 for Method 3.

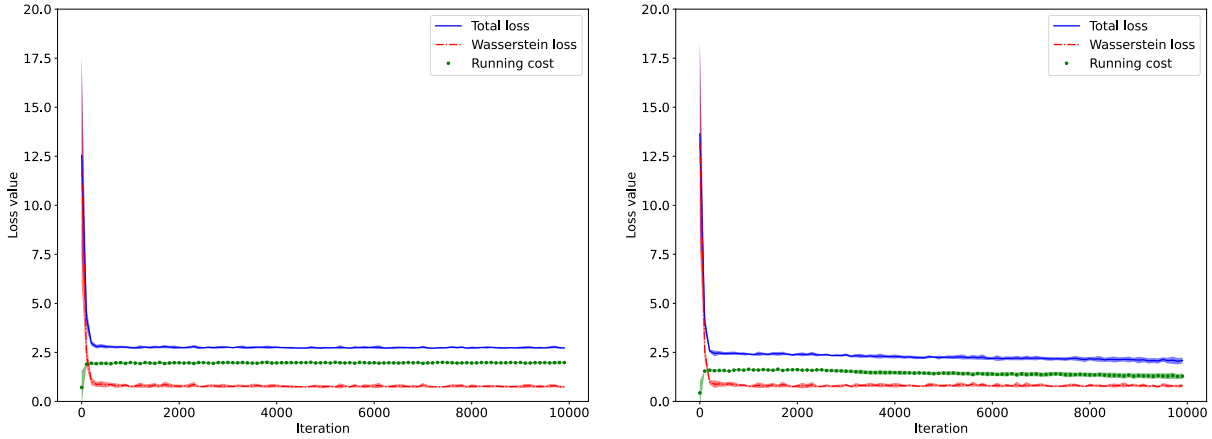
We then present the losses for the congestion test cases. Contrary to the LQ case, we do not have any benchmark solution to compare our numerical results to. We already explained why the methods gave consistent results, namely



(A) LQ Test 1 ($d = 1$)

(B) LQ Test 2 ($d = 2$)

FIGURE 9. Evolution of the loss in the linear-quadratic case for method 3.



(A) Case 1, No congestion ($d = 1$)

(B) Case 2, Congestion ($d = 1$)

FIGURE 10. Evolution of the loss in the congestion cases 1 and 2 for method 1.

that we see the density spreading before reforming the terminal distribution, in contrast with the LQ case where the shape of the population remains the same along the trajectory (see 4.2.2). The following comments can be made about the plots of the losses:

- Figures 10 and 11 show the losses for Method 1. The total loss is the sum of the "running cost" loss and the "terminal penalization" loss multiplied by C_W. For instance in dimension 1 we took $C_W = 10$, meaning that the penalization is ten times the Wasserstein distance between the effective terminal distribution of agents and the desired one. In all three cases, we observe that for the first iterations, the distance is high and the running cost almost zero, which comes from the fact that the first try is not to move. Then the algorithm makes the penalization decrease by paying a trade-off in the form of the running cost. In the case with no congestion (Figure 10a), it seems to reach a plateau which approximately corresponds to the theoretical optimal running

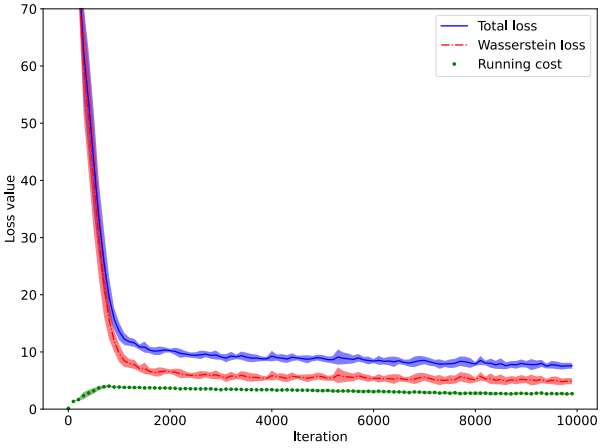
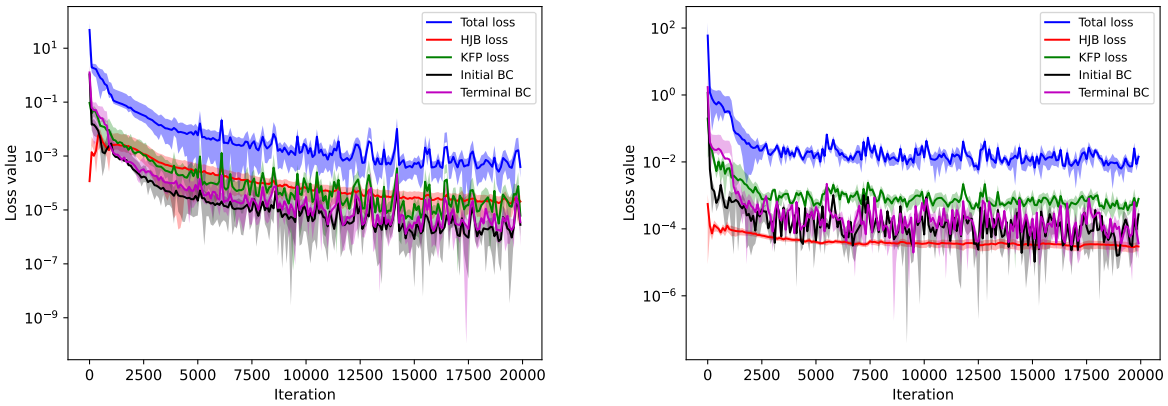


FIGURE 11. Evolution of the loss in the congestion case in dimension 5 for method 1.



(A) Case 1, No congestion ($d = 1$)

(B) Case 2, Congestion ($d = 1$)

FIGURE 12. Evolution of the loss in the congestion cases 1 and 2 for method 2.

cost[3], while in the cases with congestion (in dimension 1 and 5), the running cost keeps on decaying slowly as the algorithm tries to spread as much as possible before moving.

- Figure 12 and 13 show the losses for Method 2. The "HJB loss" is the squared L^2 residual for the HJB equation. The "KFP loss" is the squared L^2 residual for the KFP equation. The "initial BC" loss and the "terminal BC" loss respectively correspond to the squared L^2 error on the initial and terminal distributions. The "total loss" is the sum of the other losses, up to multiplicative weights.
- Figures 14 and 15 show, for Method 3, the squared L^2 residuals for the HJB and KFP equations, as well as the squared L^2 loss for the initial and terminal conditions. Note that these losses are not directly minimized during the algorithm of Method 3, but they are minimized as a by-product of the iterations.

[3]This test case with no congestion is in fact another instance of the LQ test in which we only want to move from one Gaussian distribution to another one with the same variance. Hence the optimal cost is the distance between the two means, which in this case is 2.

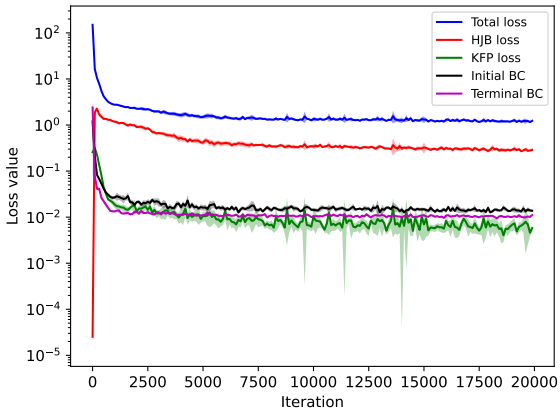
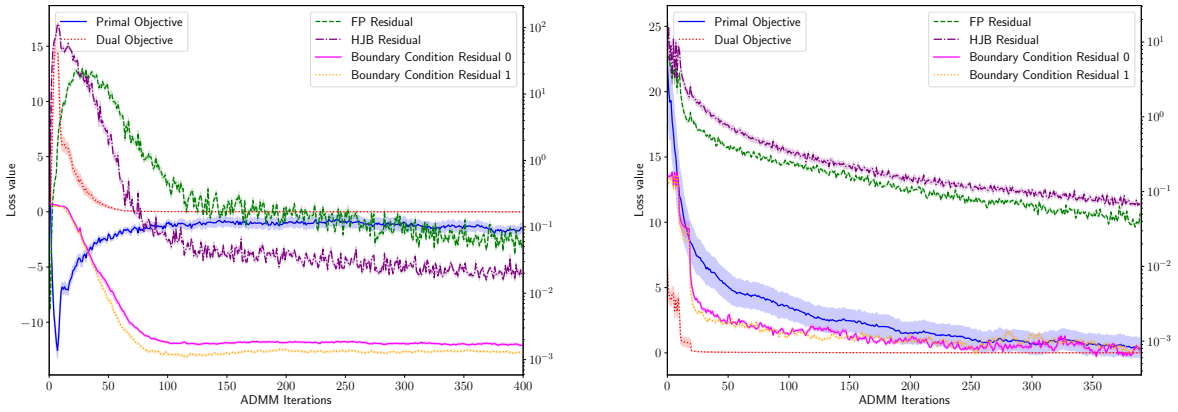


FIGURE 13. Evolution of the loss in the congestion case in dimension 5 for method 2.



(A) Case 1, No Congestion ($d = 1$)

(B) Case 2, Congestion ($d = 1$)

FIGURE 14. Evolution of the loss in the congestion cases 1 and 2 for method 3.

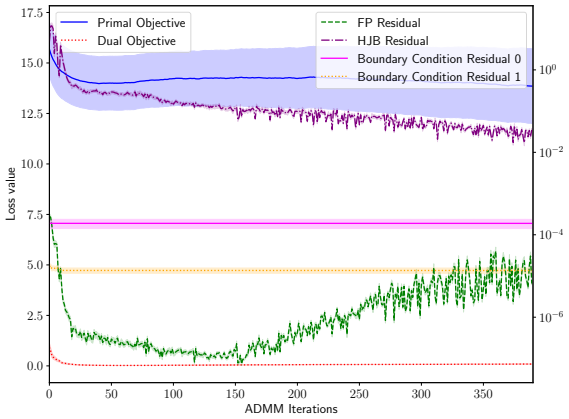


FIGURE 15. Evolution of the loss in the congestion case in dimension 5 for method 3.