

# Kernel Expansions for High-Dimensional Mean-Field Control with Non-local Interactions

Alexander Vidal, Samy Wu Fung, Stanley Osher, Luis Tenorio, Levon Nurbekyan

**Abstract**—Mean-field control (MFC) problems aim to find the optimal policy to control massive populations of interacting agents. These problems are crucial in areas such as economics, physics, and biology. We consider the non-local setting, where the interactions between agents are governed by a suitable kernel. For  $N$  agents, the interaction cost has  $\mathcal{O}(N^2)$  complexity, which can be prohibitively slow to evaluate and differentiate when  $N$  is large. To this end, we propose an efficient primal-dual algorithm that utilizes basis expansions of the kernels. The basis expansions reduce the cost of computing the interactions, while the primal-dual methodology decouples the agents at the expense of solving for a moderate number of dual variables. We also demonstrate that our approach can further be structured in a multi-resolution manner, where we estimate optimal dual variables using a moderate  $N$  and solve decoupled trajectory optimization problems for large  $N$ . We illustrate the effectiveness of our method on an optimal control of 5000 interacting quadrotors.

**Index Terms**—mean-field control, interaction kernels, optimal control, swarm control, deep learning, Hamilton-Jacobi-Bellman, mean-field games.

## I. INTRODUCTION

Mean-Field Control (MFC) [6] is a class of optimal control problems that primarily focus on controlling systems comprising a large number of identical interacting agents. These problems are designed to optimize *collective* rather than individual behaviour of the agents; in other words, the goal is to control the statistical properties of the population, e.g., the distribution of the agents in the state space or the physical space. Therefore, MFC problems can be found in numerous applications, including epidemic modeling [2], [27], [11], finance [9], [12], [7], and water distribution [3], [4]. There are two primary computational challenges when solving MFC problems, the curse-of-dimensionality, and the coupling among agents arising from *non-local interactions*. While many methods have addressed the

AV, SWF, and LT are with the Department of Applied Mathematics and Statistics at Colorado School of Mines. SO is the Department of Mathematics at UCLA. LN is with the Department of Mathematics at Emory University.

This work was partially funded by NSF DMS-2110745, AFOSR MURI FA9550-18-502 and ONR N00014-20-1-2787.

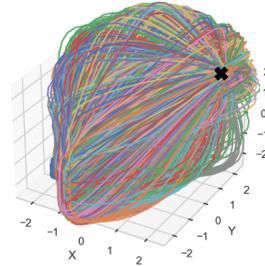


Fig. 1. Optimal swarm trajectories for 5000 quadrotors using our proposed primal-dual approach. The black X denotes the target.

curse-of-dimensionality when solving MFCs [45], [19], [52], [31], a key computational challenge remains when solving high-dimensional MFC problems with *non-local* interactions; namely, when modeling  $N$  agents with non-local interactions [40], evaluation of the interaction cost has complexity  $\mathcal{O}(N^2)$  at any given time, rendering existing MFC approaches computationally expensive.

### A. Our Contribution

We propose a primal-dual framework based on kernel expansions to solve high-dimensional MFC problems with *non-local interactions*. In particular, by introducing coefficients that capture the mean-field interaction [39], our proposed primal-dual approach decouples all the agents at the expense of the updates of a moderate number of dual variables, rendering the optimization of primal variables parallelizable. In our numerical results, we are able to approximate optimal control of 5000 quadrotor agents (see Figure (1)). Accompanying animated graphics and code for our experiments can be found in <https://github.com/mines-opt-ml/kernel-expansions-for-mfc>.

## II. BACKGROUND

We are interested in MFC problems of the form

$$\begin{aligned} & \inf_{\theta, \rho} \mathcal{J}(\theta, \rho) \\ & \text{s.t. } \partial_t \rho_t(\mathbf{x}) + \nabla \cdot (\rho_t(\mathbf{x}) f(t, \mathbf{x}, \theta)) = 0, \quad t \in (0, T), \end{aligned} \quad (1)$$

where  $\{\rho_t\}$ ,  $t \in [0, T]$ , are probability densities on  $\mathbb{R}^d$  with known initial density  $\rho_0$ , representing the distribution of the agents in the state-space at each time  $t$ . The objective function  $\mathcal{J}$  is defined as

$$\mathcal{J}(\boldsymbol{\theta}, \rho) = \int_0^T [\mathbb{E}_{\mathbf{x} \sim \rho_s} L(s, \mathbf{x}, \boldsymbol{\theta}_{\mathbf{x}}(s)) + \mathcal{F}(\rho_s)] ds + \mathcal{G}(\rho_T), \quad (2)$$

where  $\boldsymbol{\theta} : \mathbb{R}^d \times [0, T] \rightarrow \mathcal{U} \subset \mathbb{R}^q$  is the control and  $\mathcal{U}$  is a sufficiently regular domain (see [16] see Sec. I.3, I.8-9); the function  $f : [0, T] \times \mathbb{R}^d \times \mathbb{R}^q \rightarrow \mathbb{R}^d$  models the dynamics of the agents, and  $L : [0, T] \times \mathbb{R}^d \times \mathcal{U} \rightarrow \mathbb{R}$  is the Lagrangian that controls a particular performance criterion, e.g., kinetic energy. Note that the indistinguishability of the agents yields controls that depend (besides the time variable) only on the position of one agent at a time.

The functionals  $\mathcal{F}$  and  $\mathcal{G}$  model the interactions among agents within the time-horizon  $(0, T)$  and at the final time  $T$ , respectively. For the terminal interaction cost we use

$$\mathcal{G}(\rho_T) = \mathbb{E}_{\mathbf{x} \sim \rho_T} G(\mathbf{x})$$

for a suitably chosen  $G$ .

For the non-local interactions within the time horizon  $(0, T)$  we use

$$\mathcal{F}(\rho_t) = \frac{1}{2} \mathbb{E}_{(\mathbf{x}, \mathbf{y}) \sim \rho_t \otimes \rho_t} K(\mathbf{x}, \mathbf{y}), \quad (3)$$

where  $K : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$  is a positive definite kernel [37, Chapter 6]. In our examples, the interaction term penalizes agents from being too close to one another. Thus, we want  $K(\mathbf{x}, \mathbf{y})$  to be large when  $\mathbf{x}$  and  $\mathbf{y}$  are close and small otherwise. Common choices of kernels include a Gaussian kernel [40], [1] or the inverse distance kernel [31], [15]. The double expectation (or integral) in (3) results in a significant computational burden when solving the MFC problem (2) because evaluation of  $\mathcal{F}$  for  $N$  agents results in  $\mathcal{O}(N^2)$  complexity.

### III. PRIMAL-DUAL FRAMEWORK FOR AGENT DECOUPLING

#### A. Saddle Point Formulation

To address the computational challenges associated with evaluation of the interaction cost (3), we extend the kernel expansion framework introduced in [33], [34], [39], [40], [43] to MFC problems. We begin by approximating the kernel in (3) with with a quadratic (in the feature space) function

$$K(\mathbf{x}, \mathbf{y}) \approx K_r(\mathbf{x}, \mathbf{y}) = \boldsymbol{\zeta}(\mathbf{x})^\top \mathbf{K}_r \boldsymbol{\zeta}(\mathbf{y}), \quad (4)$$

where  $\boldsymbol{\zeta} = (\zeta_1, \zeta_2, \dots, \zeta_r)^\top$ ,  $\zeta_i \in \mathcal{C}^2(\mathbb{R}^d)$  are some basis functions or features, and  $\mathbf{K}_r$  is an  $r \times r$  symmetric positive-definite matrix. Note that when  $\mathbf{K}_r$  is diagonal we obtain a truncated Mercer series approximation [36]. While there are many ways to choose  $\boldsymbol{\zeta}$  [43], [1], in this work we use a neural network approximation. More details are provided in Sec. V. The objective function (2) can now be approximated using

$$\begin{aligned} \mathcal{J}_r(\boldsymbol{\theta}, \rho) &= \int_0^T \mathbb{E}_{\mathbf{x} \sim \rho_s} L(s, \mathbf{x}, \boldsymbol{\theta}_{\mathbf{x}}(s)) ds \\ &+ \frac{1}{2} \int_0^T \mathbb{E}_{(\mathbf{x}, \mathbf{y}) \sim \rho_s \otimes \rho_s} K_r(\mathbf{x}, \mathbf{y}) ds \\ &+ \mathcal{G}(\rho_T). \end{aligned} \quad (5)$$

Next, we rewrite the approximate interaction term in (5) using (4) to obtain

$$\begin{aligned} &\frac{1}{2} \mathbb{E}_{(\mathbf{x}, \mathbf{y}) \sim \rho_t \otimes \rho_t} K_r(\mathbf{x}, \mathbf{y}) \\ &= \frac{1}{2} \mathbb{E}_{\mathbf{x} \sim \rho_t} \boldsymbol{\zeta}(\mathbf{x})^\top \mathbf{K}_r \mathbb{E}_{\mathbf{y} \sim \rho_t} \boldsymbol{\zeta}(\mathbf{y}) \\ &= \frac{1}{2} \mathbf{c}_t^\top \mathbf{K}_r \mathbf{c}_t, \end{aligned} \quad (6)$$

where

$$\mathbf{c}_t = \mathbb{E}_{\mathbf{x} \sim \rho_t} \boldsymbol{\zeta}(\mathbf{x}), \quad t \in (0, T).$$

Using convex duality, we have that

$$\begin{aligned} \frac{1}{2} \mathbf{c}_t^\top \mathbf{K}_r \mathbf{c}_t &= \sup_{\mathbf{a}_t} \left[ \mathbf{a}_t^\top \mathbf{c}_t - \frac{1}{2} \mathbf{a}_t^\top \mathbf{K}_r^{-1} \mathbf{a}_t \right] \\ &= \sup_{\mathbf{a}_t} \left[ \mathbb{E}_{\mathbf{x} \sim \rho_t} \mathbf{a}_t^\top \boldsymbol{\zeta}(\mathbf{x}) - \frac{1}{2} \mathbf{a}_t^\top \mathbf{K}_r^{-1} \mathbf{a}_t \right]. \end{aligned} \quad (7)$$

Hence, optimizing (5) is equivalent to the saddle point problem

$$\begin{aligned} &\inf_{\boldsymbol{\theta}, \rho} \sup_{\mathbf{a}} \int_0^T \mathbb{E}_{\mathbf{x} \sim \rho_s} L(s, \mathbf{x}, \boldsymbol{\theta}_{\mathbf{x}}(s)) ds + \mathcal{G}(\rho_T) \\ &- \int_0^T \mathbb{E}_{\mathbf{x} \sim \rho_s} \mathbf{a}_s^\top \boldsymbol{\zeta}(\mathbf{x}) ds - \frac{1}{2} \int_0^T \mathbf{a}_s^\top \mathbf{K}_r^{-1} \mathbf{a}_s ds. \end{aligned}$$

Interchanging the inf and sup and changing the sign of the objective function, we arrive at the variational formulation in [40]:

$$\begin{aligned} &\inf_{\mathbf{a}} \sup_{\boldsymbol{\theta}} \frac{1}{2} \int_0^T \mathbf{a}_s^\top \mathbf{K}_r^{-1} \mathbf{a}_s ds \\ &- \int_0^T \mathbb{E}_{\mathbf{x} \sim \rho_s} L(s, \mathbf{x}, \boldsymbol{\theta}_{\mathbf{x}}(s)) ds - \int_0^T \mathbb{E}_{\mathbf{x} \sim \rho_s} \mathbf{a}_s^\top \boldsymbol{\zeta}(\mathbf{x}) ds \\ &- \mathcal{G}(\rho_T) \end{aligned} \quad (8)$$

s.t.  $\partial_s \rho_s(\mathbf{x}) + \nabla \cdot (\rho_s(\mathbf{x}) f(s, \mathbf{x}, \boldsymbol{\theta}_{\mathbf{x}}(s))) = 0$ ,  $s \in (0, T)$ .

Since Lagrangian coordinates are more suitable for high-dimensional problems [45], [31], [20], [19], we reformulate (8) in Lagrangian coordinates. More specifically, let

$$\partial_s \mathbf{z}_{\mathbf{x}}(s) = f(s, \mathbf{z}_{\mathbf{x}}(s), \boldsymbol{\theta}_{\mathbf{x}}(s)), \quad \mathbf{z}_{\mathbf{x}}(0) = \mathbf{x}, \quad (9)$$

for  $s \in (0, T)$ , where we suppress the dependence of  $\mathbf{z}$  on  $\boldsymbol{\theta}$  until it is necessary.

Then  $\mathbf{x} \sim \rho_0$  yields  $\mathbf{z}_\mathbf{x}(s) \sim \rho_s$ , where  $\rho_s$  is the solution of the constraints equation in (8). Consequently, (8) can be written as

$$\begin{aligned} & \inf_{\mathbf{a}} \sup_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{x} \sim \rho_0} \left[ \frac{1}{2} \int_0^T \mathbf{a}_s^\top \mathbf{K}_r^{-1} \mathbf{a}_s ds \right. \\ & \quad - \int_0^T L(s, \mathbf{z}_\mathbf{x}(s), \boldsymbol{\theta}_\mathbf{x}(s)) ds - \int_0^T \mathbf{a}_s^\top \zeta(\mathbf{z}_\mathbf{x}(s)) ds \\ & \quad \left. - G(\mathbf{z}_\mathbf{x}(T)) \right], \\ & \text{s.t. (9) holds.} \end{aligned} \quad (10)$$

The function  $\mathbf{a}_t$  encodes information about the interactions among the agents. Indeed, the optimization over  $\boldsymbol{\theta}$  in (10) *decouples* into regular optimal control problems

$$\begin{aligned} \phi_{\mathbf{a}}(t, \mathbf{x}) &= \inf_{\boldsymbol{\theta}} \int_t^T L(s, \mathbf{z}_\mathbf{x}(s), \boldsymbol{\theta}_\mathbf{x}(s)) ds \\ & \quad + \int_t^T \mathbf{a}_s^\top \zeta(\mathbf{z}_\mathbf{x}(s)) ds + G(\mathbf{z}_\mathbf{x}(T)), \quad (11) \\ & \text{s.t. (9) holds.} \end{aligned}$$

Thus, once an *optimal*  $\mathbf{a}_t$  is known, agents only need to solve decoupled optimal control problems, and the interaction information is reflected in the dependence of the cost function in (11) on  $\mathbf{a}_t$ . Hence, throughout this paper, we refer to  $\mathbf{a}_t$  as the *global interaction coefficients*.

The function  $\phi_{\mathbf{a}}$  is called the value function [16]; it is the viscosity solution of the Hamilton-Jacobi equation

$$\begin{cases} -\partial_t \phi(t, \mathbf{x}) + H(t, \mathbf{x}, \nabla_{\mathbf{x}} \phi(t, \mathbf{x})) = \mathbf{a}_t^\top \zeta(\mathbf{x}) \\ \phi(T, \mathbf{x}) = G(\mathbf{x}). \end{cases} \quad (12)$$

Eliminating the optimization over  $\boldsymbol{\theta}$  in (10) we arrive at

$$\inf_{\mathbf{a}} \frac{1}{2} \int_0^T \mathbf{a}_s^\top \mathbf{K}_r^{-1} \mathbf{a}_s ds - \mathbb{E}_{\mathbf{x} \sim \rho_0} \phi_{\mathbf{a}}(0, \mathbf{x}), \quad (13)$$

which is then the dual formulation of the optimization of (5) and reveals again the variational nature of the global interaction coefficients.

Some remarks are in order. First, we have transformed the *coupled* optimization problem (2) into a saddle point problem where the agents are *completely decoupled* at the expense of introducing a moderate number of dual variables (global interaction coefficients). That is, there is no longer a double expectation arising from (3) to compute the interaction between every pair of agents. Importantly, for a fixed vector function  $\mathbf{a}$ , we can solve for  $\boldsymbol{\theta}$  in a completely parallel manner. In this case our original control  $\boldsymbol{\theta}$  is the primal variable and the interaction coefficients  $\mathbf{a}_t$  are the dual variables. Second, the global interaction coefficients  $\mathbf{a}_t$  can be *reused*. More

specifically, the primal-dual problem only needs to be solved *once*. One may then solve another instance of (8). In our experiments we show how (8) can be solved for one instance using 100, 400, 800, and 1000 agents and can then reuse the corresponding  $\mathbf{a}_{100}^*$ ,  $\mathbf{a}_{400}^*$ ,  $\mathbf{a}_{800}^*$ , and  $\mathbf{a}_{1000}^*$  for an experiment with a new instance of the MFC problem with 1000 agents (see Sec. V-E).

## B. Discretization and Primal-Dual Algorithm

To approximate the solution to (10), we use the direct transcription approach [14] for simplicity. However, we note that any optimal control/trajectory generation algorithm (e.g., [52], [45], [42]) can be used with our primal-dual framework when solving for  $\boldsymbol{\theta}$  in (10). Fix a uniform time discretization

$$0 = t_1 < t_2 < \dots < t_n = T, \quad (14)$$

with time-step  $h = t_k - t_{k-1}$ . We define a discrete formulation where the time integrals in (10) are approximated using sums over the grid (14) and the expectations are approximated using sample averages over  $N$  agents  $\mathbf{x}_1, \dots, \mathbf{x}_N$  drawn independently from  $\rho_0$ :

$$\begin{aligned} \mathcal{L}(\mathbf{a}, \boldsymbol{\theta}) &= \frac{h}{2} \sum_{k=1}^n \mathbf{a}_{t_k}^\top \mathbf{K}_r^{-1} \mathbf{a}_{t_k} \\ & \quad - \frac{h}{N} \sum_{k=1}^n \sum_{\ell=1}^N L(t_k, \mathbf{z}_{\mathbf{x}_\ell}(t_k), \boldsymbol{\theta}_{\mathbf{x}_\ell}(t_k)) \\ & \quad - \frac{h}{N} \sum_{k=1}^n \sum_{\ell=1}^N \mathbf{a}_{t_k}^\top \zeta(\mathbf{z}_{\mathbf{x}_\ell}(t_k)) - \frac{1}{N} \sum_{\ell=1}^N G(\mathbf{z}_{\mathbf{x}_\ell}(T)), \end{aligned} \quad (15)$$

where  $\mathbf{z}_{\mathbf{x}_\ell}(t_k)$  is the discretized state for the  $\ell^{\text{th}}$  agent at time  $t_k$  obtained using, e.g., Euler's method to solve the ODE (9). From now on we will use  $\boldsymbol{\theta}$  and  $\mathbf{a}$  to denote, respectively, the matrices with entries  $\boldsymbol{\theta}_{\mathbf{x}_\ell}(t_k)$  and  $\mathbf{a}_{t_k}$ , and  $\boldsymbol{\theta}_\mathbf{x} = (\boldsymbol{\theta}_\mathbf{x}(t_1), \dots, \boldsymbol{\theta}_\mathbf{x}(t_n))$ .

Importantly, we note again that  $\mathcal{L}$  is *separable* with respect to the agents; that is, we may write,

$$\mathcal{L}(\mathbf{a}, \boldsymbol{\theta}) = \frac{1}{N} \sum_{\ell=1}^N \mathcal{L}_\ell(\mathbf{a}, \boldsymbol{\theta}_{\mathbf{x}_\ell}), \quad (16)$$

where

$$\begin{aligned} \mathcal{L}_\ell(\mathbf{a}, \boldsymbol{\theta}_\ell) &= \frac{h}{2} \sum_{k=1}^n \mathbf{a}_{t_k}^\top \mathbf{K}_r^{-1} \mathbf{a}_{t_k} \\ & \quad - h \sum_{k=1}^n L(t_k, \mathbf{z}_{\mathbf{x}_\ell}(t_k), \boldsymbol{\theta}_{\mathbf{x}_\ell}(t_k)) \\ & \quad - h \sum_{k=1}^n \mathbf{a}_{t_k}^\top \zeta(\mathbf{z}_{\mathbf{x}_\ell}(t_k)) - G(\mathbf{z}_{\mathbf{x}_\ell}(T)). \end{aligned} \quad (17)$$

Following [39], [33], we approximate the saddle point of (17) with the following primal-dual iterations

$$\boldsymbol{\theta}_{\mathbf{x}_\ell}^{k+1} = \arg \max_{\boldsymbol{\theta}} \mathcal{L}_\ell(\mathbf{a}^k, \boldsymbol{\theta}_{\mathbf{x}_\ell}), \quad \ell = 1, \dots, N \quad (18)$$

$$\mathbf{a}^{k+1} = \arg \min_{\mathbf{a}} \mathcal{L}(\mathbf{a}, \boldsymbol{\theta}_\ell^{k+1}) + \frac{\|\mathbf{a} - \mathbf{a}^k\|^2}{2\gamma}, \quad (19)$$

where  $\gamma > 0$  is a chosen parameter and the optimization variables  $(\mathbf{a}, \boldsymbol{\theta})$  are initialized randomly. We note that the primal update (18) can be done *in parallel* for each agent and can be solved efficiently using, e.g., L-BFGS [32], while the dual update consists of minimizing a quadratic term with the analytic solution

$$\mathbf{a}_{t_i}^{k+1} = (\mathbf{I} + h_a \mathbf{K}_r^{-1})^{-1} \left( \mathbf{a}_{t_i}^k + h_a \frac{1}{N} \sum_{l=1}^N \zeta \left( \mathbf{z}_{\mathbf{x}_i}^{\boldsymbol{\theta}_l^{k+1}}(t_i) \right) \right) \quad (20)$$

for  $i = 1, 2, \dots, n$ , where  $h_a = h\gamma$ . In our experiments,  $\mathbf{K}_r$  is the identity matrix because we use a kernel expansion of the form

$$K_r(\mathbf{x}, \mathbf{y}) = \zeta(\mathbf{x})^\top \zeta(\mathbf{y}), \quad (21)$$

where  $\zeta: \mathbb{R}^d \rightarrow \mathbb{R}^r$  is a neural network (see Sec. V-A)

#### IV. RELATED WORKS

##### A. Multiagent Control

Traditional approaches to multiagent control do not operate in the mean-field framework. The paper [22] introduces a theoretical framework for the discrete time model for autonomous agents presented in [47] that sparked a variety of studies exploring multiagent control. One can now find many surveys and tutorials on multiagent control; for example, [38], [17], [8], [29], [48], [5], [28]. The works in [42], [46], [30], [41] propose methods for multi-agent optimal control that use a neural network to parameterize the value function. A survey on these approaches and their connections to optimal transport and generative modeling can be found in [44]. Several interesting applications of multiagent control include wind turbines [23], vehicle coordination [25], and reusable launch vehicles [18]. Other quadrotor simulation experiments can be found in [21], [35]. We refer to [26] for multiagent swarm planning.

##### B. Mean-Field Control

Many works have explored solving multi-agent swarm problems using MFC. The early work [51] provides a multiagent control approach for interacting agents related to MFC by incorporating statistical estimates of the global system properties. [53] introduces an MFC approach that replaces local coordination strategies and

is supplemented with stability analyses and velocity control experiments consisting of 1024 robot swarms. For a comprehensive introduction/review of MFCs, we refer the reader to the survey [13] and references therein.

Our proposed method is most closely related to [1], which employs a similar kernel expansion approach but is used in the context of potential mean-field games (MFG). In particular, [1] uses a random Fourier features approximation of the interaction term as opposed to a neural network approach. A similar method uses polynomial bases for low-dimensional MFGs [40]. We refer to the survey [50] for MFG approaches that decouple interaction and agent control. Our work is also motivated by the works in [45], [31] and [49]; the former uses deep learning to solve MFC and MFG problems. The latter combines policy gradient estimation with the MFC framework and provides a linear convergence result. However, neither work considers non-local interactions.

#### V. NUMERICAL EXPERIMENTS

In order to demonstrate the effectiveness of our primal-dual framework, we conduct experiments with swarms consisting of 1000 and 5000 agents. The primal-dual formulation proves to be valuable in addressing challenges associated with large-scale swarm control where the dimensionality of the problem arising from the number of agents poses significant computational hurdles when computing the non-local interaction between agents. The generated trajectories showcase coordinated motion of interacting agents using two types of dynamics, double integrator and quadcopter [10]. These experiments emphasize the algorithm's ability to numerically solve optimal control of swarms. For all experiments, we choose the following stopping criteria:  $\|\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}^k)\| \leq \epsilon_{tol}$  (optimality criterion for primal variable),  $\|\mathbf{a}_{t_i}^k - \frac{1}{N} \sum_{l=1}^N \zeta(\mathbf{z}_{\mathbf{x}_i}^{\boldsymbol{\theta}_l^{k+1}}(t_i))\| < \epsilon_{tol}$  for all  $t_i$  (optimality criterion for dual variable), and  $\|\nabla_{\boldsymbol{\theta}} \mathcal{J}_r(\boldsymbol{\theta}^k)\| < \epsilon_{tol}$  in order to ensure the MFC problem (5) is solved. Here, we choose  $\epsilon_{tol} = 0.5$ .

##### A. Kernel Approximation

In our experiments, we use exponential interaction kernels

$$K(\mathbf{x}, \mathbf{y}) = \alpha_1 e^{-\|\mathbf{x}-\mathbf{y}\|^2/2}, \quad \mathbf{x}, \mathbf{y} \in \mathbb{R}^3. \quad (22)$$

We build the approximation in (4) in two steps. First, we find  $\tilde{\zeta} = (\tilde{\zeta}_1, \tilde{\zeta}_2, \dots, \tilde{\zeta}_r)$  such that

$$\tilde{K}(\mathbf{x}, \mathbf{y}) = e^{-\|\mathbf{x}-\mathbf{y}\|^2/2} \approx \tilde{\zeta}(\mathbf{x})^\top \tilde{\zeta}(\mathbf{y}). \quad (23)$$

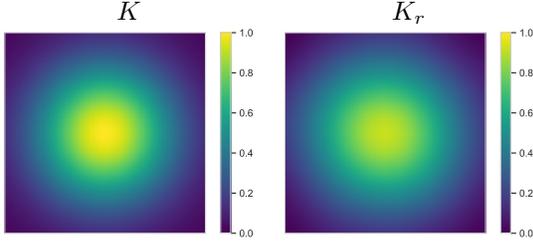


Fig. 2. **Kernel Approximation:** Gaussian kernel (left) and neural network kernel approximation using  $r = 50$  basis functions (right).

Next, we set

$$\zeta(\mathbf{x}) = \sqrt{\alpha_1} \tilde{\zeta}(\mathbf{x}).$$

Hence, we achieve (4) with  $\mathbf{K}_r$  being the identity matrix.

For (23), we train a single-layer feed-forward neural network  $\tilde{\zeta}$  with one 100-dimensional hidden layer. The network outputs  $r = 50$  basis functions  $\tilde{\zeta}_i$  for  $i = 1, \dots, r$  and has 5450 total network parameters. The network is trained on  $1e5$  samples drawn uniformly from the cube  $[-3, 3]^3$  for  $5e4$  iterations using the Adam optimizer [24] with a learning rate of  $1e-3$  and a scheduler that scales the learning rate by  $1e-1$  every  $1e4$  iterations. Training occurs by minimizing the sample mean squared-error between the true kernel values and our approximate kernel values generated by the network. In order to obtain higher accuracy, we also penalize the norm of the difference of the true gradient and the approximate gradient of the kernel. In this experiment, we obtain a validation loss of  $7.70e-03$ , which we found to be accurate enough for our MFC experiments. As an illustration, we show the true Gaussian kernel and its neural network approximation in Figure 2.

## B. Experimental Setup

1) *Double Integrator:* We approximate the optimal controller of a swarm of agents with dynamics given by

$$\dot{\mathbf{z}} = \begin{cases} \dot{x} = v_x \\ \dot{y} = v_y \\ \dot{z} = v_z \\ \dot{v}_x = a_x \\ \dot{v}_y = a_y \\ \dot{v}_z = a_z \end{cases} \quad (24)$$

where the controls are denoted by the variable  $\boldsymbol{\theta} = (v_x, v_y, v_z) \in \mathbb{R}^3$  and  $\dot{v}$  are the accelerations that correspond to each spatial dimension. Agents are penalized if they get too close to each other using the interaction kernel (22) and the penalty parameter  $\alpha_1 = 2e5$  to weigh the interaction cost.

The agents must also avoid colliding into two obstacles. The obstacles are modeled as a function  $Q : \mathbb{R}^3 \rightarrow \mathbb{R}$  defined as in [42]; namely,

$$Q(\mathbf{x}) = Q_A(\mathbf{x})\mathbf{1}_A(\mathbf{x}) + Q_B(\mathbf{x})\mathbf{1}_B(\mathbf{x}),$$

where  $A = [-2, 2] \times [-0.5, 0.5] \times [0, 7]$  and  $B = [2, 4] \times [-1, 1] \times [0, 4]$ ,  $\mathbf{1}_S$  is the indicator function of a set  $S$ , and  $Q_A$  and  $Q_B$  are, respectively, the density functions of the Gaussian distributions  $N(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$  and  $N(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$  with  $\boldsymbol{\mu}_1 = (0, 0, 2)$ ,  $\boldsymbol{\mu}_2 = (2.5, 0, 2)$ ,  $\boldsymbol{\Sigma}_1 = \text{Diag}\{9, 3, 9\}$ , and  $\boldsymbol{\Sigma}_2 = \text{Diag}\{9, 3, 3\}$ . To avoid the obstacles,  $Q$  is included in the Lagrangian as

$$L(t, \mathbf{z}_x(t), \boldsymbol{\theta}) = \|\boldsymbol{\theta}_x(t)\|^2 + \alpha_2 Q(\mathbf{z}_x(t)) \quad (25)$$

with  $\alpha_2 = 1e7$ . Finally, we use the terminal cost

$$G(\mathbf{z}_x(T)) = \frac{\alpha_3}{2} \|\mathbf{z}_x(T) - \mathbf{z}_{\text{target}}\|^2 \quad (26)$$

with  $\alpha_3 = 1e4$ . The agents are initialized randomly as  $\mathbf{z}_{x_l}(0) \sim N(\mathbf{x}_{\text{mean}}, 0.8\mathbf{I})$ ,  $l = 1, 2, \dots, N$ , and  $\mathbf{x}_{\text{mean}} = (0, -0.5, 0, \dots, 0) \in \mathbb{R}^6$ . The target for the agents is chosen to be the point  $(0, 0, 7, 0, \dots, 0) \in \mathbb{R}^6$ .

To solve the primal problem, we use 250 L-BFGS [32] iterations for fast initial improvement followed by 750 iterations of gradient descent to ensure convergence of the gradient norm.

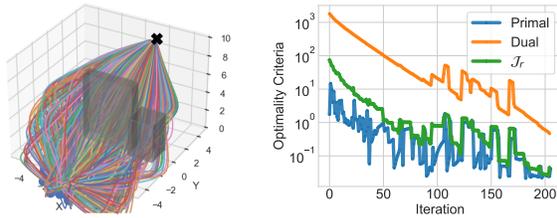
2) *Quadrotor:* We approximate the optimal controller of a swarm of quadrotors whose dynamics are given by

$$\dot{\mathbf{z}} = \begin{cases} \dot{x} = v_x \\ \dot{y} = v_y \\ \dot{z} = v_z \\ \dot{\psi} = v_\psi \\ \dot{\theta} = v_\theta \\ \dot{\phi} = v_\phi \\ \dot{v}_x = \frac{u}{m} (\sin(\phi) \sin(\psi) + \cos(\phi) \cos(\psi) \sin(\theta)) \\ \dot{v}_y = \frac{u}{m} (-\cos(\psi) \sin(\phi) + \cos(\phi) \sin(\psi) \sin(\theta)) \\ \dot{v}_z = \frac{u}{m} \cos(\theta) \cos(\phi) - g \\ \dot{v}_\psi = \tilde{\tau}_\psi \\ \dot{v}_\theta = \tilde{\tau}_\theta \\ \dot{v}_\phi = \tilde{\tau}_\phi \end{cases} \quad (27)$$

where the state

$\mathbf{z} = (x, y, z, \psi, \theta, \phi, v_x, v_y, v_z, v_\psi, v_\theta, v_\phi) \in \mathbb{R}^{12}$  is comprised of the spatial positions  $(x, y, z)$ , the angular orientation of the quadrotor,  $(\psi, \theta, \phi)$ , and  $v$  defines the velocities associated with the spatial positions and angular orientations. The controls are defined by the variable  $\boldsymbol{\theta} = (u, \tilde{\tau}_\psi, \tilde{\tau}_\theta, \tilde{\tau}_\phi) \in \mathbb{R}^4$  where  $u$  is the primary propulsion force oriented downward from the underside of the quadrotor and  $(\tilde{\tau}_\psi, \tilde{\tau}_\theta, \tilde{\tau}_\phi)$  define the torques corresponding to  $(\psi, \theta, \phi)$ . Here, agents must avoid collision with one another. The penalty parameters are the same as those described in the double-integrator

### 1000 Agents: Double Integrator



### 1000 Agents: Quadrotors

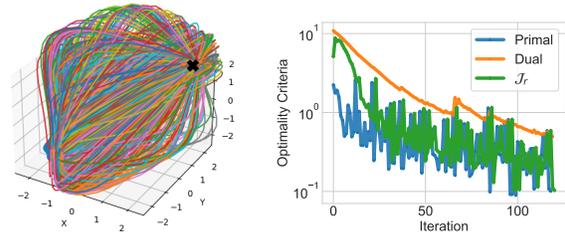


Fig. 3. Swarm Control Experiments. **Left panel:** Trajectories and gradient norms for double integrator dynamics with 1000 agents. **Right panel:** Trajectories and gradient norms for quadrotor dynamics with 1000 agents. On the left of each panel, trajectories are shown where each agent must reach the target (black X) and avoid colliding with one another. On the right of each panel, gradient norm history of primal-dual is shown: The primal and dual optimality criterion are plotted in blue and orange, respectively. The gradient norm of the original MFC problem (5) with the approximate kernel is shown in green.

experiments and are chosen as  $\alpha_1 = 5e4$ ,  $\alpha_2 = 0$ , and  $\alpha_3 = 2e3$ . In each primal-dual iteration, we use 20 L-BFGS [32] iterations for the primal problem (18) per dual update (19).

### C. Results

For both dynamics, we plot trajectories and optimality criteria using 1000 agents in Figure 3. For optimality criteria, we plot the gradient norm history of the objective function (5) (in green) along with the primal and dual optimality criteria (in blue and orange).

Our results show that our proposed primal-dual approach solves the proposed problem (5) in a decoupled manner as seen in the iterates (18). The 5000-quadrotor experiment shown in Figure 1 in the introduction has the same experimental setup. These experiments were run on an Apple MacBook Pro 8-core 3200 MHz M1 laptop running macOS Monterey 12.0.1 with a total of 16GB of RAM.

### D. Time Comparison: Coupled vs. Primal-Dual

To illustrate why the primal-dual approach is preferred, we compare the runtimes between solving (1) directly (we refer to this as the coupled approach) and the proposed primal-dual approach using the quadrotor dynamics with 20 and 50 agents. Figure 5 shows the advantages of using the primal-dual approach, especially as the number of agents increases. The experiments were run on an Apple MacBook Pro 8-core 3200 MHz M1 laptop running macOS Monterey 12.0.1 with a total of 16GB of RAM. As expected, the primal-dual approach is faster in terms of cumulative runtime due to the sequential nature of the coupled approach. It is worth noting that our simple implementation did not fully utilize the parallel nature of our primal-dual framework and further time gains could be achieved with a more

advanced parallel implementation (e.g., using a star network topology infrastructure).

### E. Reusing the Global Mean-Field Interaction

An important benefit of our primal-dual approach is that the global interaction coefficients  $\mathbf{a}$  can be computed, saved, and then reused on future instances of the MFC problem (1). This is because once  $\mathbf{a}$  is obtained, only the primal problem requires optimization regardless of the number of agents. Consequently,  $\mathbf{a}$  can be determined using fewer agents than the number of trajectories that are required as long as the interaction is adequately approximated. This is particularly beneficial for situations where, for example, some quadrotors might lose power because re-computing the interaction is not required. We empirically demonstrate this in Figure 4 as follows. First, we compute the optimal coefficients  $\mathbf{a}_{100}^*$  using 100 agents,  $\mathbf{a}_{400}^*$  using 400 agents,  $\mathbf{a}_{800}^*$  using 800 agents, and  $\mathbf{a}_{1000}^*$  using 1000 agents. Second, we sample 1000 fixed initial conditions  $\mathbf{x} \sim \rho_0$ . Finally, we re-solve the primal problem only with these new initial points using  $\mathbf{a}_{100}$ ,  $\mathbf{a}_{400}$ ,  $\mathbf{a}_{800}$ , and  $\mathbf{a}_{1000}$  until the primal optimality condition is satisfied. Figure 4 shows that, as long as there are enough samples, trajectories generated using  $\mathbf{a}_{100}^*$ ,  $\mathbf{a}_{400}^*$ ,  $\mathbf{a}_{800}^*$ , and  $\mathbf{a}_{1000}^*$  are qualitatively and quantitatively similar. Finally, as previously stated, Figure 4 indicates that one may solve for  $\mathbf{a}^*$  using fewer agents and use this pre-computed  $\mathbf{a}^*$  to solve new instances of the MFC problem, with a potentially larger number of agents, and in a completely parallel manner.

## VI. CONCLUSIONS

We propose a primal-dual framework based on kernel expansions for solving mean-field control (MFC) problems. Our approach decouples agent interactions by introducing global interaction coefficients based on kernel

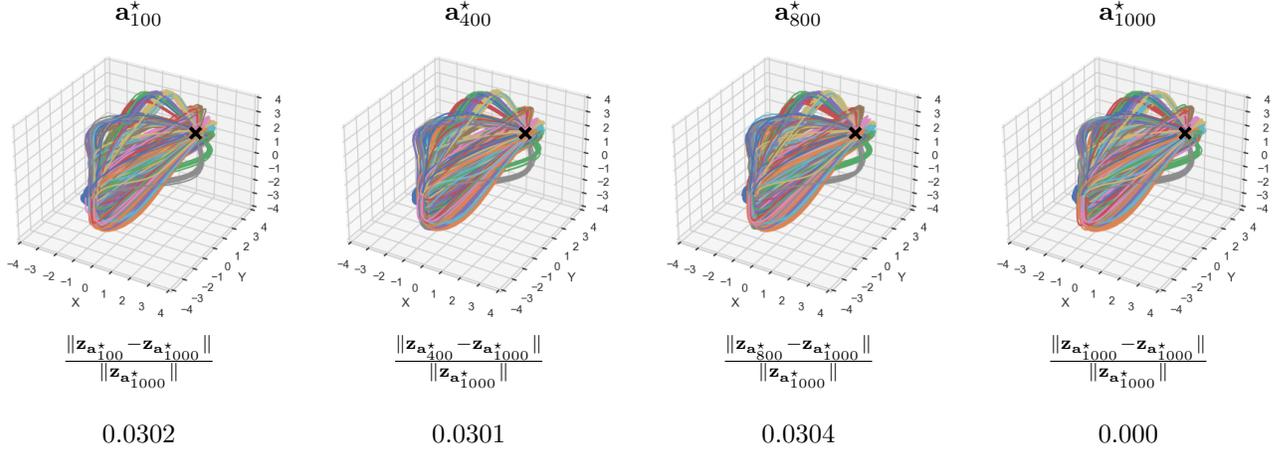


Fig. 4. **Quadrotor Controller:** Illustration of trajectories using different mean-field interaction coefficients. Here,  $a_{100}^*$ ,  $a_{400}^*$ ,  $a_{800}^*$ , and  $a_{1000}^*$  are interaction coefficients by solving the primal-dual for 100, 400, 800, and 1000 agents, respectively. Next, to make a proper comparison among the different coefficients, we sample a fixed set of 1000 initial conditions from  $\rho_0$  and re-solve the primal problem using  $a_{100}^*$ ,  $a_{400}^*$ ,  $a_{800}^*$ ,  $a_{1000}^*$ .

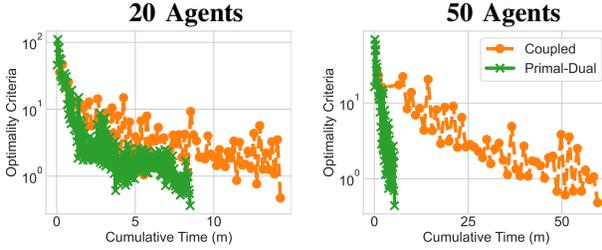


Fig. 5. Gradient norm vs. cumulative runtime (minutes) for the objective function  $J_r$  in (5) for 20 and 50 agents. Each algorithm is run three times and the fastest run is chosen. Optimization stops when the gradient norm reaches  $5e - 1$ . Orange shows the coupled approach and green shows our primal-dual approach.

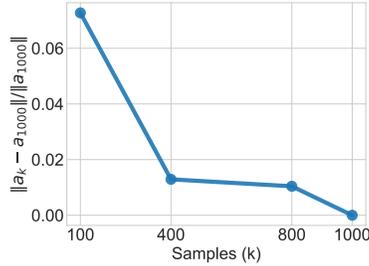


Fig. 6. Number of agents used to train the mean-field interaction coefficients  $\mathbf{a}$  vs. relative norm difference  $\|\mathbf{a}_k - \mathbf{a}_{1000}\|/\|\mathbf{a}_{1000}\|$  for  $k = 100, 400, 800,$  and  $1000$  agents for quadrotor dynamics.

expansions. These coefficients allow us to reformulate the MFC problem as a saddle point problem where the primal problem can be solved in parallel. Moreover, new instances of the MFC problem can be solved at reduced costs by reutilizing the global interaction coefficients.

Our experiments show that the primal-dual framework is effective at solving MFC problems, including an optimal swarm control problem of 5000 quadrotors.

## REFERENCES

- [1] S. Agrawal, W. Lee, S. W. Fung, and L. Nurbekyan. Random features for high-dimensional nonlocal mean-field games. *J. Comput. Phys.*, 459:111136, 2022.
- [2] A. Aurell, R. Carmona, G. Dayanikli, and M. Lauriere. Optimal incentives to mitigate epidemics: a Stackelberg mean field game approach. *SIAM Journal on Control and Optimization*, 60(2):S294–S322, 2022.
- [3] J. Barreiro-Gomez and H. Tembine. Mean-field-type model predictive control: An application to water distribution networks. *IEEE Access*, 7:135332–135339, 2019.
- [4] J. Barreiro-Gomez and H. Tembine. *Mean-field-type Games for Engineers*. CRC Press, 2021.
- [5] J. Bellingham, M. Tillerson, A. Richards, and J. P. How. *Multi-Task Allocation and Path Planning for Cooperating UAVs*, pages 23–41. Springer US, Boston, MA, 2003.
- [6] A. Bensoussan, J. Frehse, P. Yam, et al. *Mean Field Games and Mean Field Type Control Theory*, volume 101. Springer, 2013.
- [7] A. Calvia, S. Federico, G. Ferrari, and F. Gozzi. A mean-field model of optimal investment. *arXiv preprint arXiv:2404.02871*, 2024.
- [8] Y. Cao, W. Yu, W. Ren, and G. Chen. An overview of recent progress in the study of distributed multi-agent coordination. *IEEE Transactions on Industrial Informatics*, 9, 07 2012.
- [9] P. Cardaliaguet and C.-A. Lehalle. Mean field game of controls and an application to trade crowding. *Mathematics and Financial Economics*, 12:335–363, 2018.
- [10] L. R. G. Carrillo, A. E. D. López, R. Lozano, and C. Pégard. *Quad Rotorcraft Control: Vision-Based Hovering and Navigation*. Springer Science & Business Media, 2012.
- [11] A. Charpentier, R. Elie, M. Laurière, and V. C. Tran. Covid-19 pandemic control: balancing detection policy and lockdown intervention under icu sustainability. *Mathematical Modelling of Natural Phenomena*, 15:57, 2020.
- [12] B. Djehiche, H. Tembine, and R. Tempone. A stochastic maximum principle for risk-sensitive mean-field type control. *IEEE Transactions on Automatic Control*, 60(10):2640–2649, 2015.

- [13] K. Elamvazhuthi and S. Berman. Mean-field models in swarm robotics: A survey. *Bioinspiration & Biomimetics*, 15(1):015001, 2019.
- [14] P. J. Enright and B. A. Conway. Discrete approximations to optimal trajectories using direct transcription and nonlinear programming. *Journal of Guidance, Control, and Dynamics*, 15(4):994–1002, 1992.
- [15] G. E. Fasshauer and M. J. McCourt. *Kernel-Based Approximation Methods using Matlab*, volume 19. World Scientific Publishing Company, 2015.
- [16] W. H. Fleming and H. M. Soner. *Controlled Markov Processes and Viscosity Solutions*, volume 25. Springer Science & Business Media, 2006.
- [17] V. Gazi and B. Fidan. Coordination and control of multi-agent dynamic systems: Models and approaches. *Swarm Robotics*, 4433:71–102, 07 2007.
- [18] Q. Gong, W. Kang, N. S. Bedrossian, F. Fahroo, P. Sekhavat, and K. Bollino. Pseudospectral optimal control for military and industrial applications. In *2007 46th IEEE Conference on Decision and Control*, pages 4128–4142, 2007.
- [19] J. Han, A. Jentzen, and W. E. Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences*, 115(34):8505–8510, 2018.
- [20] J. Han and E. Weinan. Deep learning approximation for stochastic control problems. *ArXiv*, abs/1611.07422, 2016.
- [21] W. Hönig, J. A. Preiss, T. K. S. Kumar, G. S. Sukhatme, and N. Ayanian. Trajectory planning for quadrotor swarms. *IEEE Transactions on Robotics*, 34(4):856–869, 2018.
- [22] A. Jadbabaie, J. Lin, and A. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001, 2003.
- [23] K. Johnson, L. Pao, M. Balas, and L. Fingersh. Control of variable-speed wind turbines: standard and adaptive techniques for maximizing energy capture. *IEEE Control Systems Magazine*, 26(3):70–81, 2006.
- [24] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, San Diego, CA, USA, 2015.
- [25] M. R. Kirchner, M. J. DeBord, and J. P. Hespanha. A hamilton-jacobi formulation for optimal coordination of heterogeneous multiple vehicle systems. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11623–11630, 2020.
- [26] S. Knorn, Z. Chen, and R. H. Middleton. Overview: Collective control of multiagent systems. *IEEE Transactions on Control of Network Systems*, 3(4):334–347, 2016.
- [27] W. Lee, S. Liu, W. Li, and S. Osher. Mean field control problems for vaccine distribution. *Research in the Mathematical Sciences*, 9(3):51, 2022.
- [28] F. Lewis, H. Zhang, K. Movric, and A. Das. *Cooperative Control of Multi-Agent Systems: Optimal and Adaptive Design Approaches*. Springer US, 01 2014.
- [29] F. L. Lewis, H. Zhang, K. Hengster-Movric, and A. Das. *Cooperative Control of Distributed Multi-Agent Systems*, pages i–xv. John Wiley and Sons, Ltd, 2007.
- [30] X. Li, D. Verma, and L. Ruthotto. A neural network approach for stochastic optimal control. *arXiv preprint arXiv:2209.13104*, 2022.
- [31] A. T. Lin, S. W. Fung, W. Li, L. Nurbekyan, and S. J. Osher. Alternating the population and control neural networks to solve high-dimensional stochastic mean-field games. *Proceedings of the National Academy of Sciences*, 118(31):e2024713118, 2021.
- [32] D. C. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical programming*, 45(1):503–528, 1989.
- [33] S. Liu, M. Jacobs, W. Li, L. Nurbekyan, and S. J. Osher. Computational methods for first-order nonlocal mean field games with applications. *SIAM Journal on Numerical Analysis*, 59(5):2639–2668, 2021.
- [34] S. Liu and L. Nurbekyan. Splitting methods for a class of non-potential mean field games. *Journal of Dynamics and Games*, 8(4):467–486, 2021.
- [35] T. Meng, Z. Zhang, J. Darbon, and G. Karniadakis. Sympocnet: Solving optimal control problems with applications to high-dimensional multiagent path planning problems. *SIAM Journal on Scientific Computing*, 44(6):B1341–B1368, 2022.
- [36] J. Mercer. Functions of positive and negative type, and their connection the theory of integral equations. *Philosophical transactions of the royal society of London. Series A*, 209(441-458):415–446, 1909.
- [37] M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of Machine Learning*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA, 2 edition, 2018.
- [38] R. M. Murray. Recent Research in Cooperative Control of Multivehicle Systems. *Journal of Dynamic Systems, Measurement, and Control*, 129(5):571–583, 05 2007.
- [39] L. Nurbekyan. One-dimensional, non-local, first-order stationary mean-field games with congestion: A Fourier approach. *Discrete and Continuous Dynamical Systems - S*, 11(5):963–990, 2018.
- [40] L. Nurbekyan et al. Fourier approximation methods for first-order nonlocal mean-field games. *Portugaliae Mathematica*, 75(3):367–396, 2019.
- [41] D. Onken, L. Nurbekyan, X. Li, S. W. Fung, S. Osher, and L. Ruthotto. A neural network approach applied to multi-agent optimal control. In *2021 European Control Conference (ECC)*, pages 1036–1041. IEEE, 2021.
- [42] D. Onken, L. Nurbekyan, X. Li, S. W. Fung, S. Osher, and L. Ruthotto. A neural network approach for high-dimensional optimal control applied to multiagent path finding. *IEEE Transactions on Control Systems Technology*, 31(1):235–251, 2023.
- [43] A. Rahimi and B. Recht. Random features for large-scale kernel machines. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007.
- [44] L. Ruthotto. Differential equations for continuous-time deep learning. *arXiv preprint arXiv:2401.03965*, 2024.
- [45] L. Ruthotto, S. J. Osher, W. Li, L. Nurbekyan, and S. W. Fung. A machine learning framework for solving high-dimensional mean field game and mean field control problems. *Proceedings of the National Academy of Sciences*, 117(17):9183–9193, 2020.
- [46] D. Verma, N. Winovich, L. Ruthotto, and B. v. B. Waanders. Neural network approaches for parameterized optimal control. *arXiv preprint arXiv:2402.10033*, 2024.
- [47] T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen, and O. Shochet. Novel type of phase transition in a system of self-driven particles. *Phys. Rev. Lett.*, 75:1226–1229, Aug 1995.
- [48] J. Wang and X. Hu. Distributed consensus in multi-vehicle cooperative control: Theory and applications (ren, w. and beard, r.w.; 2008) [book shelf]. *IEEE Control Systems Magazine*, 30(3):85–86, 2010.
- [49] W. Wang, J. Han, Z. Yang, and Z. Wang. Global convergence of policy gradient for linear-quadratic mean-field control/game in continuous time. In *International Conference on Machine Learning*, pages 10772–10782. PMLR, 2021.
- [50] Y. Wang, C. Yang, T. Li, X. Mi, L. Li, and Z. Han. A survey on mean-field game for dynamic management and control in space-air-ground network. *IEEE Communications Surveys & Tutorials*, pages 1–1, 2024.
- [51] P. Yang, R. A. Freeman, and K. M. Lynch. Multi-agent coordination by decentralized estimation and control. *IEEE Transactions on Automatic Control*, 53(11):2480–2496, 2008.
- [52] Y. Zhao and J. Han. Offline supervised learning vs online direct policy optimization: A comparative study and a unified training paradigm for neural network-based optimal feedback control. *arXiv preprint arXiv:2211.15930*, 2022.
- [53] T. Zheng, Q. Han, and H. Lin. Transporting robotic swarms via mean-field feedback control. *IEEE Transactions on Automatic Control*, 67(8):4170–4177, 2021.