2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

High-dimensional Optimal Density Control with Wasserstein Metric Matching

Shaojun Ma Mengxue Hou Xiaojing Ye Haomin Zhou

Abstract— **We present a novel computational framework for density control in high-dimensional state spaces. The considered dynamical system consists of a large number of indistinguishable agents whose behaviors can be collectively modeled as a time-evolving probability distribution. The goal is to steer the agents without collision from an initial distribution to reach (or approximate) a given target distribution within a fixed time horizon at minimum cost. To tackle this problem, we propose to model the drift as a nonlinear reduced-order model, such as a deep network, and enforce the matching to the target distribution at terminal time either strictly or approximately using the Wasserstein metric. The resulting saddle-point problem can be solved by an effective numerical algorithm that leverages the excellent representation power of deep networks and fast automatic differentiation for this challenging high-dimensional control problem. A variety of numerical experiments were conducted to demonstrate the performance of our method.**

I. INTRODUCTION

Recent years have witnessed an emerging research interest in optimal control problems with large collective of agents, such as drones, robots, and vehicles [4], [11]. Such problems appear in many real-world applications, including surveillance over a large region, time-sensitive search-and-rescue, infrastructure inspection and maintenance, among many others. Meanwhile, the cost of manufacture, including the builtin electrical components and batteries, has reduced significantly during the past decades. This enables swarming of agents a cost-effective strategy for sophisticated tasks otherwise difficult to accomplish. In practice, the devices used in swarm are often of small size and thus have limited computation and communication power. Hence optimal swarm control, designing efficient algorithms that can steer the agents to achieve specified high-level tasks in a way that accommodates the high dimensionality of the system [37], becomes particularly important.

A. Related Work

Optimal control of large collective of agents has been commonly considered in the context of density control based on mean-field models [12], [44], [27]. The idea of mean-field models is to characterize the collective behaviors of the agents in the swarm using a time-evolving probability density function, which are solutions of some partial differential equations (PDEs). In these models, the probability density does not depend on the number of agents, but only on the state space of individual agents. Nevertheless, direct computation of these PDEs can still be computationally prohibitive due to the high dimension of the state space. Thus, most existing works on density controls approximate the density functions based on spatial discretization [1] or using kernel density estimation [2], [14]. The work [14] designed velocity field to make swarms density converges to the desired/commanded density distribution. In [42], the authors presented a density control by improving Smoothed Particle Hydrodynamic (SPH) [34] method with collision free condition, where one defines the density of an agent as the weighted sum of distances to its neighbors within a certain range. A data-driven approach was proposed in [9] to learn the control by estimating the Koopman operators. In [41], the authors provided a control strategy to steer the state from an initial distribution to a terminal one with specified mean and covariance, which is also called nonlinear covariance control. Also see [7], [8], [21] for related works.

Density control is closely related to optimal transport [38]. Several groups have combined Wasserstein metric and optimal control to develop relevant control algorithms. For instance, [5] develops Wasserstein proximal algorithms to solve a density control problem, where a nonlinear drift term is considered. In [3], the authors discussed the optimality conditions for optimal control problems in Wasserstein spaces. In [40], a Wasserstein based robust control was developed to resolve the issue that uncertain variables are unavailable. In [22], the authors proposed a novel model-predictive control (MPC) method for limiting the safety risk when the distribution of the obstacles is set to be within an ambiguity set, which is measured by the Wasserstein metric. In [23], terminal cost as Wasserstein distance and finding the parameters of normalizing flow [10] by minimizing the terminal cost have been considered. In [13], the authors presented a primal-dual formulation of optimal control problem with existence proof and efficient numerical method, similar method can also be found in [17] where the problem is solved by a Mean Field Game (MFG) approach. In [16], the authors calculated the individual agent trajectory by alternating two gradient flows that involve an attractive potential, a repelling function, and a process of intermittent diffusion, in which way a large group of robots is employed to accomplish the task of shape formation.

Optimal control over discrete state spaces can be cast as Markov chain models as shown in [1]. For optimal

This work was supported in part by NSF grants DMS-1925263, DMS-2152960, DMS-2307465 and DMS-2307466, and ONR N00014-21-1-2891.

S. Ma and H. Zhou are with the School of Mathematics, Georgia Tech, Atlanta, GA 30332, USA {shaojunma,hmzhou}@gatech.edu

M. Hou is with Department of Electrical Engineering, University of Notre Dame, South Bend, IN 46556, USA mhou@nd.edu

X. Ye is with the Department of Mathematics and Statistics, Georgia State University, Atlanta, GA 30303, USA xye@gsu.edu

control with continuous state spaces, the density dynamics are determined by the corresponding continuity equation or Fokker-Planck equation. For instance, [35], [36] provided both theoretical analysis and numerical methods of the optimal control to achieve target equilibrium density. In [6], the dynamics of the control was discretized in time and a multi-marginal optimal transport scheme was used to solve the problem. In [24], the authors developed a method to solve the optimal transport based control using Voronoi Tessellation coverage. A distributed algorithm for density estimation to reduce the computational cost is proposed in [43]. Multi-agent optimal control has been widely applied in the domain of robotics in recent years. For a general introduction, we refer to [15]. In particular, [45] presents the work of using reinforcement learning to control multi-robot system. Furthermore, [29], [39], [33] carefully designed an objective function to minimize the control energy and meanwhile keeping collision-free conditions. Parameterizing controls using neural networks was also considered in [33].

B. Main contributions of this work

In this work, we propose a novel method to compute the optimal control u which steers an initial density as close as possible to a target density at a prescribed terminal time, while minimizing the control cost. The major advantage of the proposed method is that it can be readily applied to the cases with high-dimensional state spaces, where computation of the associated continuity equation or Fokker-Planck equations is prohibitive using existing approaches. In our approach, this issue is effectively tackled by using a large number of synthesized agents whose trajectories can be easily evaluated using decoupled ODEs or SDEs. Therefore, our approach does not require any spatial discretization using finite difference and element methods or any basis representations, and the expected cost function values can be accurately approximated using empirical expectations, which allows our method to scale to much higher dimensions that are traditionally considered computationally infeasible. Moreover, the proposed method can readily adopt parallel computation because of the decoupling, and hence the computation can be done very fast by leveraging the power of standard deep learning computation framework. We also provide convergence analysis, including the rate of convergence, of the numerical algorithm for solving the resulting saddle-point problem. We validate our approach in numerical simulations that include complex scenarios in various tests with dimension up to 100.

C. Notations

Throughout this paper, we use $|\cdot|$ to denote the absolute value of a scalar and Euclidean norm of a vector. Furthermore, $\|\cdot\|$ denotes the matrix norm induced by vector Euclidean norm. We also use $\|\cdot\|_{\cdot}$ to denote function norms where the space is specified in the subscript $\cdot$. The spatial domain is denoted by Ω, which is assumed to be a closed and quasiconvex subset of $\mathbb{R}^d$. Integrals over Ω or $\Omega \times \Omega$ are written without subscript for notation simplicity unless otherwise specified. We use μ to denote standard Lebesgue measure on $\mathbb{R}^d$. We use ∇ to denote the gradient with respect to the state variable x. Gradient and partial derivatives with respect to any other variables will be indicated by their subscripts. We denote $\mathbb{N}$ the set of positive integers and $[n] := \{1, \dots, n\}$ for notation simplicity.

II. PROPOSED METHOD

In this section, we develop a general computational framework for solving density control over high-dimensional state spaces. To maximize applicability of our framework, we use the following prototype mean-field model of density control with any user-defined (running) cost functional c:

$$\min_{u} \ \mathbb{E}\Big[\int_0^T c(x_t, u(x_t))\, dt\Big] + \gamma D(\rho_T, \nu) \tag{1}$$

where $x_t \in \Omega$ is the dynamic following:

$$\begin{cases} dx_t = u(x_t)dt + \sigma(x_t)dW_t, \\ x_0 \sim \rho_0, \end{cases} \tag{2}$$

which represents any indistinguishable agent in the swarm to be controlled. In (1), ρ_0 is a given initial population density, ν is a given target population density, D is a metric that measures the distance between two probability densities, $\gamma > 0$ is a user-chosen weight parameter, and $\rho(t, \cdot)$ is the density of x_t for any t with short hands $\rho(0, \cdot) = \rho_0(\cdot)$ and $\rho(T, \cdot) = \rho_T(\cdot)$. The time evolution of $\rho(t, \cdot)$ is governed by the Fokker-Planck equation:

$$\partial_t \rho = -\nabla \cdot (\rho u) + \frac{1}{2}\langle \nabla^2, \sigma\sigma^\top \rho \rangle, \tag{3}$$

where $\langle \nabla^2, A(x) \rangle := \sum_{i,j} \partial^2_{x_i x_j} a_{ij}(x)$ for any matrix-valued function $A(x) = [a_{ij}(x)] \in \mathbb{R}^{d\times d}$. The goal of (1) is to find the optimal control $u : \Omega \to \mathbb{R}^n$ that steers the population characterized by x_t with initial distribution ρ_0 to approximate the target distribution ν at terminal time T with minimal overall cost. The parameter γ weighs the matching of terminal density to ν against the overall cost. If an exact matching to ν at T is desired, then γ can be cast as the Lagrangian multiplier and solved jointly with u. For simplicity, we will only consider the case with soft-penalty on the matching as given in (1) in the present work. In what follows, we will discuss the choice of cost functional c and density distance D to instantiate (1).

a) Cost functional: The density control problem (1) allows for a general class of cost functional $c : \Omega \times \mathbb{R}^n \to \mathbb{R}$. Typical optimal control problems with minimal energy use $c(x, u) = \frac{1}{2}|u(x)|^2$. If there is a component $u_0(x_t)$ due to environmental force in u, then the cost functional can be set to $c(x, u) = \frac{1}{2}|u(x) - u_0(x)|^2$. In many real-world applications, the swarm with density $\rho(t, \cdot)$ is realized by a number of agents. In such cases, it is necessary that the agents do not collide with each other throughout the control process. To this end, one can impose additional penalty when any two agents become too close. For example, at any time t, such penalty function can be set to (the average should

be over all agents different from the input variable x but we use the following for notation simplicity):

$$\tfrac{1}{N}\sum_{i=1}^{N} V(x, x_t^{(i)}) \quad \left(\approx \int V(x,y)\rho(t,y)\,dy\right), \tag{4}$$

where the approximation in the parenthesis holds as $N \to \infty$, and V is an interactive potential defined as $V(x,y) = c\ln|x-y|$ or $V(x,y) = c|x-y|^{-\alpha}$ for some constants $c, \alpha > 0$, etc. The purpose of V is to incur larger cost when two agents move closer. A cost functional including such interactive potential can effectively prevent collisions.

b) Distance D between densities: We use the 1-Wasserstein distance, also known as the Earth Mover's Distance (EMD), as the distance D to measure the difference between the terminal density ρ_T and target density ν. For any pair of densities (ϱ, ν), their 1-Wasserstein distance is defined by

$$D(\varrho,\nu) = \inf\left\{\int |x-y|\,d\pi(x,y) :\ \pi \in \Pi(\varrho,\nu)\right\}, \tag{5}$$

where $\Pi(\varrho,\nu)$ denotes the set of joint distributions on $\Omega\times\Omega$ whose marginal densities are exactly ϱ and ν. The 1-Wasserstein distance defined in (5) quantifies the minimum total cost to transfer probability density ϱ to ν (and vice versa), and its optimal solution π^* describes how the transfer should be made. In real-world applications, the 1-Wasserstein distance is an appropriate metric for the density control problem (1)—its value indicates how much additional efforts, measured using the standard Euclidean distance, are needed to move the agents to reach the target distribution in case the matching is not exact at terminal time T. Moreover, 1-Wasserstein distance allows two densities to have different supports, which could be problematic for other density distance measure such as Kullback-Leibler (KL) divergence.

Despite the various advantages of Wasserstein distance, its computation can be challenging due to the optimization process required in (5). However, for 1-Wasserstein distance, one can derive its dual form given by [38]:

$$D(\varrho,\nu) = \sup\{\mathbb{E}_\varrho[\phi] - \mathbb{E}_\nu[\phi] :\ \mathrm{Lip}(\phi)\le 1\}, \tag{6}$$

where ϕ is the dual variable and $\mathrm{Lip}(\phi)$ denotes the Lipschitz constant of ϕ. As Ω is quasi-convex, we can relax the constraint $\mathrm{Lip}(\phi)\le 1$ to $\|\nabla\phi(x)\|\le 1$ for all $x\in\Omega$ a.e., where $\nabla\phi(x)$ is the weak derivative of ϕ with respect to x, and $\|\nabla\phi(x)\|$ is the induced 2-norm of $\nabla\phi(x)$ and thus its maximal singular value. This allows us to use nonlinear reduced-order models, such as deep neural networks, to parameterize ϕ and apply various techniques such as spectral normalization to enforce the constraint $\|\nabla\phi(x)\|\le 1$.

c) Complete model and approximations using deep networks: The optimization problem for solving the control problem (1) is given by

$$\inf_u \sup_\phi\ \mathcal{E}(u,\phi) \tag{7}$$

with constraint $\|\nabla\phi(x)\|\le 1$ to hold for $x\in\Omega$ a.e., and the objective functional $\mathcal{E}$ is given by

$$\mathcal{E}(u,\phi) := \mathbb{E}\left[\int_0^T c(x_t, u(x_t))\,dt\right] + \gamma\mathbb{E}_{\rho_T}[\phi] - \gamma\mathbb{E}_\nu[\phi]. \tag{8}$$

As both u and ϕ are in infinite-dimensional function spaces, we need a finite-dimensional representation of them for numerical computation. This can be achieved by approximating them using linear or nonlinear reduced-order models. In particular, we will use deep neural networks to parameterize u and ϕ as u_θ and ϕ_η respectively. Here $\theta,\eta\in\mathbb{R}^m$ (we assume both are m-dimensional for notation simplicity here, but they can be different in practice) denote their trainable network parameters, such as their weights and biases. The approximation power of deep neural networks has been justified by the universal approximation theorem (e.g., [20]).

In addition to using finite-dimensional representations u_θ and ϕ_η, we also need to approximate the integrals and expectations in (8) for numerical computations. In this work, we use the particle dynamics (2) if the agents are given and fixed, or simulate a large number of particles following (2) otherwise, to avoid direct computation of $\rho(t,\cdot)$. More precisely, we approximate the saddle-point problem (7) with deep neural network representations u_θ and ϕ_η and particle dynamics as follows

$$\min_\theta \max_\eta E(\theta,\eta) \tag{9}$$

where the objective function given by

$$E(\theta,\eta) := \frac{1}{N}\sum_{i=1}^{N}\left(c(x_t^{(i)}, u_\theta(x_t^{(i)})) + \gamma\phi_\eta(x_T^{(i)})\right) - \gamma\mathbb{E}_\nu[\phi_\eta].$$

The last term in E can also be replaced with empirical expectation $(\gamma/M)\cdot\sum_{j=1}^{M}\phi_\eta(z^j)$ for M i.i.d. samples $\{z^{(j)} : j\in[M]\}$ drawn from ν. For simplicity, we also assume that $\theta,\eta\in\Theta$ and Θ is a compact set in $\mathbb{R}^m$. Then the optimal solution θ to (9) is the parameter of u_θ approximating the optimal control u in (1).

Remark 1. *If the target distribution ν is given as M fixed discrete points $Z := \{z^{(j)} : j\in[M]\}$, then we can also set D to the Chamfer's distance between the two point clouds $X_T := \{x^{(i)} : i\in[N]\}$ and Z:*

$$\sum_{i=1}^{N}\min_{1\le j\le M}|z_T^{(i)} - z^{(j)}|^2 + \sum_{j=1}^{M}\min_{1\le i\le N}|x_T^{(i)} - z^{(j)}|^2. \tag{10}$$

The Chamfer's distance between X_T and Z given in (10) *is a promising alternative to measure the difference between these two clouds when N and M are moderately small (e.g., dozens). We will show in our experiment that the proposed method can also be applied using this Chamfer distance.*

d) Numerical computations: An important feature of the saddle-point formulation (9) for solving the optimal control problem (1) is that we can leverage parallel computation to effectively handle the large number of simulated trajectories. More precisely, for any fixed θ, we can use the Euler-Maruyama method (or the Euler method if $\sigma = 0$) to generate trajectories

$$x_{t+h}^{(i)} = x_t^{(i)} + hu(x_t^{(i)}) + \sqrt{h}\sigma(x_t^{(i)})\delta_t^{(i)} \tag{11}$$

for discrete time points $t = 0,\dots,\lfloor T/h\rfloor - 1$ where $h>0$ is the time step size and $\delta_t^{(i)}$ are i.i.d. standard normal random

variables for all t and i. Therefore, the objective function $E(\theta, \eta)$ given in (9) can be computed explicitly. Then we can use automatic differentiation to obtain the gradients of E with respect to θ and η, which is crucial to solving the saddle-point problem (9). In particular, we consider the objective function

$$L(\theta) := \max_{\eta \in \Theta} E(\theta, \eta). \tag{12}$$

The optimal solution θ to (9) is thus the minimizer of $L(\theta)$, whose full gradient with respect to θ can be computed by solving the maximization problem (12), as shown in the following lemma (whose proof can be found in [30]).

Lemma 1 (Gradient of L)**.** *Let L be defined in* (12)*. Then for any fixed θ, the gradient $\nabla_\theta L(\theta)$ at θ is given by*

$$\nabla_\theta L(\theta) = \partial_\theta E(\theta, \eta(\theta)), \tag{13}$$

where $\eta(\theta) \in \arg\max_\eta \{E(\theta, \eta) : \|\nabla \phi_\eta\| \leq 1\}$.

Lemma 1 suggests a practical implementation to compute $\nabla_\theta L(\theta)$: calculate the partial derivative of $E(\theta, \eta)$ with respect to θ with η held fixed, then substitute η by the maximizer $\eta(\theta)$ of the problem (12). Since (12) is constrained, the *gradient mapping*, defined by $\mathcal{G}(\theta) := \tau^{-1}[\theta - \Pi(\theta - \tau \nabla_\theta L(\theta))]$, is used as the convergence criterion of θ [18], [28], [25]. Note the gradient mapping reduces to the gradient $\nabla_\theta L(\theta)$ for unconstrained case. In the following theorem (the proof can be found in [30]), we show that for any $\varepsilon > 0$ the standard stochastic gradient descent scheme based on $\nabla_\theta L(\theta)$ (i.e., stochastic gradient is unbiased and has bound variance) is guaranteed to push the gradient mapping $\mathcal{G}(\theta)$ to 0 with ε accuracy within $O(\varepsilon^{-1})$ iterations in the ergodic sense. For simplicity, we assume both θ and η are of the same dimension m and lie in the same closed Euclidean ball in $\mathbb{R}^m$. The results can be easily generalized to the case where they have different dimensions m and m' and are in some convex compact subsets of $\mathbb{R}^m$ and $\mathbb{R}^{m'}$ respectively.

Theorem 1. *Suppose the parameters θ and η are bounded in a ball centered at origin with radius $R > 0$, namely, we have $\Theta := \{\theta : |\theta| \leq R\}$. For any $\varepsilon > 0$, let $\{\theta_j\}$ be a sequence of the network parameters of u_θ generated by the stochastic gradient descent algorithm, where $\nabla_\theta L(\theta)$ is approximated by the empirical expectation using samples as in* (9)*. If the sample complexity is $N = O(\varepsilon^{-1})$ in each iteration, then $min_{1 \leq j \leq J} \mathbb{E}[|\mathcal{G}(\theta_j)|^2] \leq \varepsilon$ after $J = O(\varepsilon^{-1})$ iterations.*

In practice, we found that using a small number of iterations to solve the inner maximization problem with any fixed θ appears to perform the best. The pseudocode to solve the saddle-point problem (9) is given in Algorithm 1. The symbol $\hat{\nabla}$ and $\hat{\partial}$ stand for stochastic gradient and partial derivative using mini-batch samples of X_t. The selection of parameters will be given in Section III.

Algorithm 1 Optimal Density Control (ODC)

Input: Samples $X_0 = \{x_0^{(i)} : i \in [N]\}$ following initial distribution ρ_0, samples following target distribution ν, time horizon $T > 0$, step size h, maximum outer and inner iteration numbers K_1 and K_2.
Initialize: Neural networks u_θ, ϕ_η.
for $k_{out} = 1, \ldots, K_1$ **do**
  Generate trajectories $X_t = \{x_t^{(i)} : i \in [N]\}$ using (11) with $0 \leq t \leq T$.
  Compute running cost $E(\theta, \eta)$ in (9).
  for $k_{inn} = 1, \ldots, K_2$ **do**
    Update $\eta \leftarrow \eta + \tau \hat{\nabla} \{\mathbb{E}_{\rho_{X_T}}[\phi_\eta] - \mathbb{E}_\nu[\phi_\eta]\}$.
  end for
  Update $\theta \leftarrow \theta - \tau \hat{\partial}_\theta E(\theta, \eta)$.
end for
Output: u_θ.

III. EXPERIMENTS

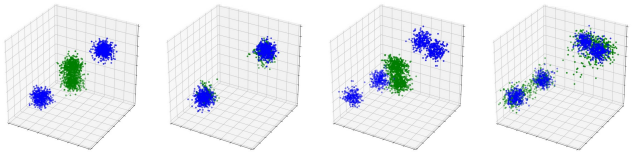
We validate Algorithm 1 (ODC) on both synthetic and realistic data sets. For specific examples, we also test Algorithm 1 using Chamfer distance 10, referred to as ODC-Chamfer. It is easier to implement ODC-Chamfer since the Chamfer distance has a closed-form expression and hence the inner iteration 1 of ODC can be eliminated.

A. Synthetic Data

We first evaluate our model on several synthetic data sets, termed by **Synthetic-1**, **Synthetic-2** and **Synthetic-3**. The state spaces of Synthetic-1 and Synthetic-2 are 3 and 100 dimensional, while Synthetic-3 is 2-dimensional respectively. We set the control $u_\theta = \nabla \psi_\theta$, where ψ_θ is a fully connected neural network with 3 hidden layers and 36 nodes per layer. We also set the dual function ϕ_η as a fully-connected neural network with 6 hidden layers and 256 nodes per layer. We use `tanh` as the activation function for all layers. We implement and test Algorithm 1 using PyTorch and train the networks using the built-in optimizer Adam [26] with learning rate 10^{-4}. This optimizer appears to have improved efficiency compared to SGD empirically. Furthermore, we use spectral normalization to enforce $\|\nabla \phi_\eta\| \leq 1$ [32]. We initialize all network parameters with Xavier initialization [19] and train u_θ according to Algorithm 1 ODC (or ODC-Chamfer if the number of agents is small and the target locations are fixed). When ODC is used, we set the size of simulated agents at each time point as $N = 2,000$, and use $1,500$ points as the training set and the other 500 data points for testing. The step size h and time horizon T are set to 0.1 and 1 respectively. We choose outer iteration number as 10^4 and inner iteration number for approximating Wasserstein distance as 6, for all synthetic experiments.

Synthetic-1: We first apply Algorithm 1 ODC with noise term in a state space of dimension $d = 3$ for two instances (two different pairs of initial-target densities), denoted by Syn-1-a and Syn-1-b, respectively. The initial and target distributions are mixtures of Gaussians. We set the perturbation noise $\sigma = 0.01$. The controlled density and target density are respectively shown as the green and blue point clouds in Figure 1, which shows that the controlled density in green correctly moved from the initial at $t = 0$ to the blue target density at $t = 1$ in the presence of perturbations in both

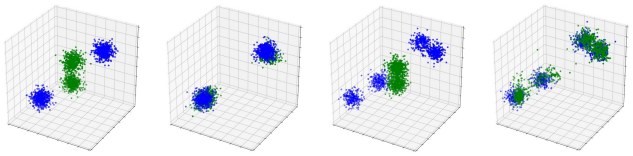
Syn-1-a and Syn-1-b.



(a) Syn-1-a:t=0 (b) Syn-1-a:t=1 (c) Syn-1-b:t=0 (d) Syn-1-b:t=1

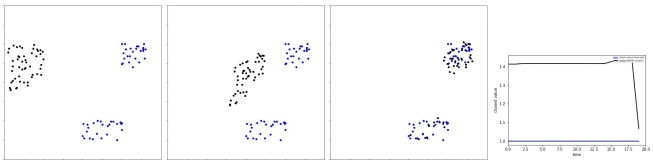
Fig. 1: Results of ODC on Synthetic-1 datasets for two instances (Syn-1-a) and (Syn-1-b) in $[-10, 10]^3$. The green point clouds indicate the controlled density, and the blue cloud points indicate the target density. (a) and (b) show the densities at $t = 0$ and $t = 1$ respectively for Syn-1-a. (c) and (d) show Syn-1-b.

Synthetic-2: We test the control using a large number of agents in a state space of dimension $d = 100$. We simulate 2 pairs of initial-target densities and name them as Syn-2-a and -b. For Syn-2-a and Syn-2-b. We used ResNet and Normalizing Flow to parameterize the control u in Syn-2-a and Syn-2-b respectively in the bottom row. We plot the controlled density in green point clouds and target in blue as above, and show the projections of them onto the first three coordinates (x_1, x_2, x_3) for $t = 0$ and $t = 1$ in Figure 2.



(a) Syn-2-a:t=0 (b) Syn-2-a:t=1 (c) Syn-2-b:t=0 (d) Syn-2-b:t=1

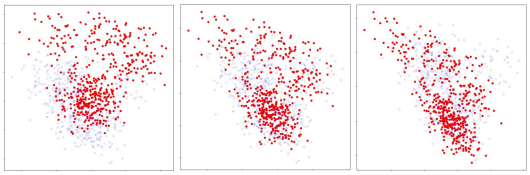
Fig. 2: Results of ODC on Synthetic-2 datasets for two instances (Syn-1-a) and (Syn-1-b) on $[-10, 10]^{100}$ (plots show projections to the first three dimensions). The green point clouds indicate the controlled density, and the blue cloud points indicate the target density. (a) and (b) show the densities at $t = 0$ and $t = 1$ respectively for Syn-1-a. (c) and (d) show Syn-1-b.



(a) Syn-3:t=0 (b) Syn-3:t=10 (c) Syn-3:t=20 (d) Closest Dist

Fig. 3: Results of ODC on Synthetic-3 for collision prevention on $[0, 200]^2$. The black point clouds indicate the controlled density, the blue cloud points indicate the target density at times (a) $t = 0$ (b) $t = 10$ (c) $t = 20$. Plot (d) shows the minimum distance between agents (top curved line) over time period [0,20], which is constantly above the minimum distance tolerance 1.0 (bottom horizontal line).

Synthetic-3: We also test Algorithm 1 (ODC) with an interaction potential with $V(x, y) = |x-y|^{-2}$ in (4) to avoid agent collision on a 2D example. The agents are indicated by the black points, and the target distribution is indicated by the blue points in Figure 3. The left three panels in Figure 3 show them at initial time $t = 0$, middle time 10, and terminal time 20. The rightmost figure shows the minimum pairwise distance (top curved line) of the controlled agents stays around 1.4 until near the terminal time. It is consistently above the required minimum pairwise distance tolerance 1.0 (bottom horizontal line) of the target sample points, showing that the learned control avoided collisions between the agents during their movements.



(a) Real: t=0 (b) Real: t=0.5 (c) Real: t=1

Fig. 4: Results on real data. The controlled density (red) and the target density (light blue) are shown at normalized times (a) $t = 0$, (b) $t = 0.5$ and (c) $t = 1$.

B. Real Data

In this experiment, we aim to control a group of Autonomous Underwater Vehicles (AUVs) to reach a target distribution. This experiment is motivated by the 16-month AUV ocean deployment (Processes driving Exchange At Cape Hatteras, PEACH) near Cape Hatteras, North Carolina, a highly dynamic region characterized by confluent western boundary currents and convergence in the adjacent shelf and slope waters. Due to the AUVs limited forward speed, the motion of AUV is highly influenced by the ocean flow field v. We define the cost functional as $r(x, u) = \frac{1}{2}\mathbb{E}[\int_0^T |u(x_t) - v(x_t)|^2 dt]$. The input flow map in this simulation is given by a 1-km horizontal resolution version of the Navy Coastal Ocean Model [31] made available by J. Book and J. Osborne (Naval Research Laboratory, Stennis Space Center). In the AUV experiment, multiple vehicles were deployed repeatedly in the same domain, to sample the variability in the position of the Hatteras Front. Since the AUVs were deployed at similar locations throughout the 16 months of experiment, we collect the starting and goal positions from all experiments as the initial and target distribution of the system. We aim to move the AUVs to the surround of target location on distribution level. Since the data size is small (10 samples only), we augment the data by generating Gaussian samples around the given locations, thus the final distributions is a mixture of Gaussians. Then we apply Algorithm 1 (ODC) to the data and show the controlled density (red) and target density (blue) at normalized time $t = 0,\ 0.5, 1.0$ in Figure 4. The network u_θ is parameterized as a ResNet with three layers and each layer has 36 nodes.As we can observe, the initial density gradually moves to the target density under the guidance of the learned control u_θ.

IV. CONCLUSION

In this paper, we develop a novel computational framework for density control. We formulate the control problem with running cost and penalize inaccurate matching to target density using Wasserstein metric. The control problem is reformulated as a saddle-point optimization problem of the control and the dual function of the Wasserstein distance. Both of the control and the dual function are parameterized as neural networks which can handle high-dimensional problems without any spatial discretization or basis representations. We validated the promising performance of this method on several synthetic and real data sets.

REFERENCES

[1] B. Açikmeşe and D. S. Bayard. A markov chain approach to probabilistic swarm guidance. In *2012 American Control Conference (ACC)*, pages 6300–6307. IEEE, 2012.

[2] S. Bandyopadhyay, S.-J. Chung, and F. Y. Hadaegh. Probabilistic and distributed control of a large-scale swarm of autonomous agents. *IEEE Transactions on Robotics*, 33(5):1103–1123, 2017.

[3] B. Bonnet and H. Frankowska. Necessary optimality conditions for optimal control problems in wasserstein spaces. *Applied Mathematics & Optimization*, 84(2):1281–1330, 2021.

[4] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo. Swarm robotics: a review from the swarm engineering perspective. *Swarm Intelligence*, 7:1–41, 2013.

[5] K. Caluya and A. Halder. Wasserstein proximal algorithms for the schr\"{o} dinger bridge problem: Density control with nonlinear drift. *IEEE Transactions on Automatic Control*, 2021.

[6] Y. Chen. Density control of interacting agent systems. *arXiv preprint arXiv:2108.07342*, 2021.

[7] Y. Chen, T. Georgiou, and M. Pavon. On the relation between optimal transport and schrödinger bridges: A stochastic control viewpoint. In *Journal of Optimization Theory and Applications*, 2016.

[8] Y. Chen, T. Georgiou, and M. Pavon. Optimal transport in systems and control. *Annual Review of Control, Robotics, and Autonomous Systems*, 4, 2020.

[9] H. Choi, U. Vaidya, and Y. Chen. A convex data-driven approach for nonlinear control synthesis. *Mathematics*, 9(19):2445, 2021.

[10] L. Dinh, D. Krueger, and Y. Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

[11] M. Dorigo, G. Theraulaz, and V. Trianni. Swarm robotics: Past, present, and future [point of view]. *Proceedings of the IEEE*, 109(7):1152–1165, 2021.

[12] K. Elamvazhuthi and S. Berman. Mean-field models in swarm robotics: A survey. *Bioinspiration & Biomimetics*, 15(1):015001, 2019.

[13] K. Elamvazhuthi, S. Liu, W. Li, and S. Osher. Dynamical optimal transport of nonlinear control-affine systems.

[14] U. Eren and B. Açıkmeşe. Velocity field generation for density control of swarms using heat equation and smoothing kernels. *IFAC-PapersOnLine*, 50(1):9405–9411, 2017.

[15] Z. Feng, G. Hu, Y. Sun, and J. Soon. An overview of collaborative robotic manipulation in multi-robot systems. *Annual Reviews in Control*, 49:113–127, 2020.

[16] C. Frederick, M. Egerstedt, and H. Zhou. Collective motion planning for a group of robots using intermittent diffusion. *Journal of Scientific Computing*, 90(1):1–20, 2022.

[17] H. Gao, W. Lee, W. Li, Z. Han, S. Osher, and H. V. Poor. Energy-efficient velocity control for massive numbers of rotary-wing uavs: A mean field game approach. In *GLOBECOM 2020-2020 IEEE Global Communications Conference*, pages 1–6. IEEE, 2020.

[18] S. Ghadimi, G. Lan, and H. Zhang. Mini-batch stochastic approximation methods for nonconvex stochastic composite optimization. *Math. Program.*, 155(1-2, Ser. A):267–305, 2016.

[19] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *International Conference on Artificial Intelligence and Statistics*, 2010.

[20] I. Gühring and M. Raslan. Approximation rates for neural networks with encodable weights in smoothness spaces. *Neural Networks*, 134:107–130, 2021.

[21] I. Haasler, Y. Chen, and J. Karlsson. Optimal steering of ensembles with origin-destination constraints. *IEEE Control Systems Letters*, 5(3):881–886, 2020.

[22] A. Hakobyan and I. Yang. Wasserstein distributionally robust motion control for collision avoidance using conditional value-at-risk. *IEEE Transactions on Robotics*, 2021.

[23] K. Hoshino. Finite-horizon control of nonlinear discrete-time systems with terminal cost of wasserstein distance. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pages 4268–4274. IEEE, 2020.

[24] D. Inoue, Y. Ito, and H. Yoshida. Optimal transport-based coverage control for swarm robot systems: Generalization of the voronoi tessellation-based method. In *2021 American Control Conference (ACC)*, pages 3032–3037. IEEE, 2021.

[25] S. J Reddi, S. Sra, B. Poczos, and A. J. Smola. Proximal stochastic methods for nonsmooth nonconvex finite-sum optimization. *Advances in neural information processing systems*, 29, 2016.

[26] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, 2014.

[27] J. Lasry and P. Lions. Mean field games. In *Japanese journal of mathematics*, 2007.

[28] Z. Li and J. Li. A simple proximal stochastic gradient method for nonsmooth nonconvex optimization. *Advances in neural information processing systems*, 31, 2018.

[29] C. E. Luis, M. Vukosavljev, and A. P. Schoellig. Online trajectory generation with distributed model predictive control for multi-robot motion planning. *IEEE Robotics and Automation Letters*, 5(2):604–611, 2020.

[30] S. Ma, M. Hou, X. Ye, and H. Zhou. High-dimensional optimal density control with wasserstein metric matching. *arXiv preprint arXiv:2307.13135*, 2023.

[31] P. J. Martin. Description of the navy coastal ocean model version 1.0. Technical Report NRL/FR/7322–00-9962, Naval Research Lab, 2000.

[32] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

[33] D. Onken, L. Nurbekyan, X. Li, S. W. Fung, S. Osher, and L. Ruthotto. A neural network approach for high-dimensional optimal control. *arXiv preprint arXiv:2104.03270*, 2021.

[34] L. C. Pimenta, N. Michael, R. C. Mesquita, G. A. Pereira, and V. Kumar. Control of swarms based on hydrodynamic models. In *2008 IEEE international conference on robotics and automation*, pages 1948–1953. IEEE, 2008.

[35] S. Roy, M. Annunziato, A. Borzì, and C. Klingenberg. A fokker–planck approach to control collective motion. *Computational Optimization and Applications*, 69:423–459, 2018.

[36] C. Sinigaglia, A. Manzoni, and F. Braghin. Density control of large-scale particles swarm through pde-constrained optimization. *IEEE Transactions on Robotics*, 38(6):3530–3549, 2022.

[37] C. Sinigaglia, A. Manzoni, F. Braghin, and S. Berman. Robust optimal density control of robotic swarms. *arXiv preprint arXiv:2205.12592*, 2022.

[38] C. Villani. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.

[39] L. Wang, A. D. Ames, and M. Egerstedt. Safety barrier certificates for collisions-free multirobot systems. *IEEE Transactions on Robotics*, 33(3):661–674, 2017.

[40] I. Yang. Wasserstein distributionally robust stochastic control: A data-driven approach. *IEEE Transactions on Automatic Control*, 66(8):3863–3870, 2020.

[41] Z. Yi, Z. Cao, E. Theodorou, and Y. Chen. Nonlinear covariance control via differential dynamic programming. In *2020 American Control Conference (ACC)*, pages 3571–3576. IEEE, 2020.

[42] S. Zhao, S. Ramakrishnan, and M. Kumar. Density-based control of multiple robots. In *Proceedings of the 2011 American control conference*, pages 481–486. IEEE, 2011.

[43] T. Zheng, Q. Han, and H. Lin. Distributed mean-field density estimation for large-scale systems. *IEEE Transactions on Automatic Control*, 67(10):5218–5229, 2021.

[44] T. Zheng, Q. Han, and H. Lin. Backstepping mean-field density control for large-scale heterogeneous nonlinear stochastic systems. In *2022 American Control Conference (ACC)*, pages 4832–4837. IEEE, 2022.

[45] P. Zhu, W. Dai, W. Yao, J. Ma, Z. Zeng, and H. Lu. Multi-robot flocking control based on deep reinforcement learning. *IEEE Access*, 8:150397–150406, 2020.