

Control of high-dimensional collective dynamics by deep neural feedback laws and kinetic modelling

Giacomo Albi ^a, Sara Bicego ^{b,*}, Dante Kalise ^b

^a Department of Computer Science, University of Verona, Strada le Grazie 15, Verona, 37134, Italy

^b Department of Mathematics, Imperial College London, South Kensington Campus, London, SW72AZ, UK

ARTICLE INFO

Keywords:

Agent-based models
Optimal feedback control
Mean field/multiscale models
High-dimensional approximation
Supervised learning
Kinetic equations

ABSTRACT

Modeling and control of agent-based models is twice cursed by the dimensionality of the problem, as both the number of agents and their state space dimension can be large. Even though the computational barrier posed by a large ensemble of agents can be overcome through a mean field formulation of the control problem, the feasibility of its solution is generally guaranteed only for agents operating in low-dimensional spaces. To circumvent the difficulty posed by the high dimensionality of the state space a kinetic model is proposed, requiring the sampling of high-dimensional, two-agent sub-problems, to evolve the agents' density using a Boltzmann type equation. Such density evolution requires a high-frequency sampling of two-agent optimal control problems, which is efficiently approximated by means of deep neural networks and supervised learning, enabling the fast simulation of high-dimensional, large-scale ensembles of controlled particles. Numerical experiments demonstrate the effectiveness of the proposed approach in the control of consensus and attraction-repulsion dynamics.

1. Introduction

Collective behaviour in agent-based models (ABMs) is of evergrowing interest across various disciplines, including mathematics, physics, biology, and economics. ABMs enable the description of complex phenomena through a general paradigm that combines endogenous interactions between agents with external influences. Their applicability spans diverse areas, such as social sciences [29,46], robotics, and computer science [22]. A fundamental topic of interest in ABMs is the study of pattern formation and self-organization [8,54]. However, beyond self-organization, a fascinating topic arises in relation to the design of external signals or controls to influence a system and inducing a prescribed collective behaviour [19].

Agent-based models encode pairwise agent-to-agent interactions through a balance of attraction and repulsion forces acting over first or second-order dynamics, while the influence of the external world on the system is expressed as a suitable control signal. To make matters more concrete, let us consider a second-order system with N agents in $\mathbb{R}^d$, where the state of the i -th agent is encoded by the pair $s_i = (x_i, v_i) \in \Omega_x \times \Omega_v \subset \mathbb{R}^{2d}$, representing position and velocity, respectively, evolving according to transport-interaction

* Corresponding author.

E-mail addresses: giacomo.albi@univr.it (G. Albi), s.bicego21@imperial.ac.uk (S. Bicego), d.kalise-balza@imperial.ac.uk (D. Kalise).

<https://doi.org/10.1016/j.jcp.2025.114229>

Received 3 April 2024; Received in revised form 13 June 2025; Accepted 6 July 2025

Available online 9 July 2025

0021-9991/© 2025 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

dynamics of the form

$$\begin{aligned}\dot{x}_i(t) &= v_i(t) \\ \dot{v}_i(t) &= \frac{1}{N} \sum_{j=1}^N P(x_i(t), x_j(t))(v_j(t) - v_i(t)) + u_i(t), \quad i = 1, \dots, N.\end{aligned}\quad (1)$$

Here, $P(\cdot, \cdot) : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ denotes an interaction kernel, while $u_i(t)$ is a control signal influencing agent i . The ensemble of control signals is denoted by $\mathbf{u}(t) = (u_1(t), \dots, u_N(t))^T$. The core of the self-organization behaviour of the free dynamics resides in P , which can induce clustering, polarization or alignment, among many others. This self-organization behaviour can be modified by the influence of an external control law. In the framework of dynamic optimization, this control is synthesized by minimizing a cost functional which rewards the convergence of the system towards a cooperative goal, e.g. consensus

$$\bar{\mathbf{u}}(t) = \underset{\mathbf{u}(\cdot)}{\operatorname{argmin}} \left\{ J_N(\mathbf{x}, \mathbf{v}, \mathbf{u}) := \frac{1}{N} \int_0^{+\infty} \sum_{i=1}^N \|v_i - \bar{v}\|^2 + \gamma \ell(u_i) dt \right\}, \quad \gamma > 0, \quad (\text{OCP})$$

where the first term in the cost is promoting consensus towards the target velocity $\bar{v}$, while $\ell : \mathbb{R}^d \rightarrow \mathbb{R}_+ \cup \{0\}$ is a convex function penalizing the energy spent by the control u_i .

The solution of the optimal control problem defined by minimizing (OCP) subject to (1) is *twice cursed* by the dimensionality of the problem: solving the OCP becomes prohibitively expensive for large values of N (as in swarm robotics or collective animal behaviour where hundreds or thousands of agents are present), as well as for high-dimensional state spaces (i.e. $d \gg 1$, as in portfolio optimization). A partial remedy to this problem comes from statistical mechanics by assuming the number of agents $N \rightarrow \infty$, so that the dynamics of the individual-based problem can be approximated by a mean field equation

$$\partial_t f + v \cdot \nabla_x f = -\nabla_v \cdot \left((P[f] + u)f \right), \quad (2)$$

where $f(t, x, v)$ is the probability density of having an agent with state (x, v) at time t , and the mean field interaction force is given by the non-local operator

$$P[f](t, x, v) := \int_{\Omega_x \times \Omega_v} P(x, x_*) (v_* - v) f(t, x_*, v_*) dx_* dv_*. \quad (3)$$

A direct transcription of the objective (OCP) leads to the mean field optimal control problem [28]

$$\min_u J(f, u) := \int_0^{+\infty} \int_{\Omega_x \times \Omega_v} (\|v - \bar{v}\|^2 + \gamma \ell(u)) f(t, x, v) dx dv dt. \quad (\text{MFOC})$$

Even though the formulation using (2) alleviates the curse of dimensionality with respect to N , it leads to a PDE-constrained optimization problem over $2d + 1$ dimensions, which becomes prohibitively expensive already for moderate values of d . The solution of high-dimensional mean field optimal control problems has been addressed using deep learning techniques in [55] in the case of linear dynamics, however, the nonlinear case remains open. In this paper, we propose the synthesis of a feedback control for (2) which is a suboptimal solution to (MFOC). In order to overcome the obstacle posed by the treatment of high-dimensional agents with nonlinear interactions, we resort to modeling the evolution of the agents density f from a kinetic viewpoint, reformulating the mean field controlled dynamics as a Povzner-Boltzmann type equation

$$\partial_t f(t, x, v) + v \cdot \nabla_x f = \lambda \mathcal{Q}_{\eta, u}(f, f)(t, x, v), \quad (4)$$

where the operator $\mathcal{Q}$ takes into account the gain and the loss of particles in (x, v) at time t , due to the motion of individuals via free transport and the velocity changes resulting from the controlled interaction dynamics, see [48, 52]. If we consider the interaction dynamics in (1), reduced to the case of $N = 2$ particles with velocities (v, v_*) , by denoting as (v', v'_*) the update of those velocities after a forward Euler step of length η , we have

$$\mathcal{Q}_{\eta, u}(f, f)(t, x, v) = \int_{\Omega_x \times \Omega_v} \frac{1}{J_\eta} f(t, x, v') f(t, x_*, v'_*) - f(t, x, v) f(t, x_*, v_*) dx_* dv_*, \quad (5)$$

where $(v', v'_*) \mapsto (v, v_*)$ are the pre-interaction velocities that generate the couple (v, v_*) , while J_η is the Jacobian of the binary interactions map $(v, v_*) \mapsto (v', v'_*)$.

We will show that the alternative mesoscopic description of ABMs via (4) is consistent with a sub-optimal, controlled solution of the constrained mean field PDE (2), when assuming high frequency λ and weak interactions η between particles, similarly to the *grazing-collision* limit in kinetic theory [38]. The convenience of such kinetic formulation relies on the reduced computational cost required for its solution via *direct simulation Monte Carlo* (DSMC) methods, [4]. Such a sampling-based technique approximates the solution of the Boltzmann (4) by computing a collection of binary sub-problems for couples of agents sampled from the population density function $f(t, \cdot)$ [48].

Assuming a high interaction frequency between agents requires, at each evolution step, the solution of a large number of reduced 2-agent (OCP). For efficiency purposes, we rely on supervised learning to build an *Artificial Neural Networks* (ANN) approximating

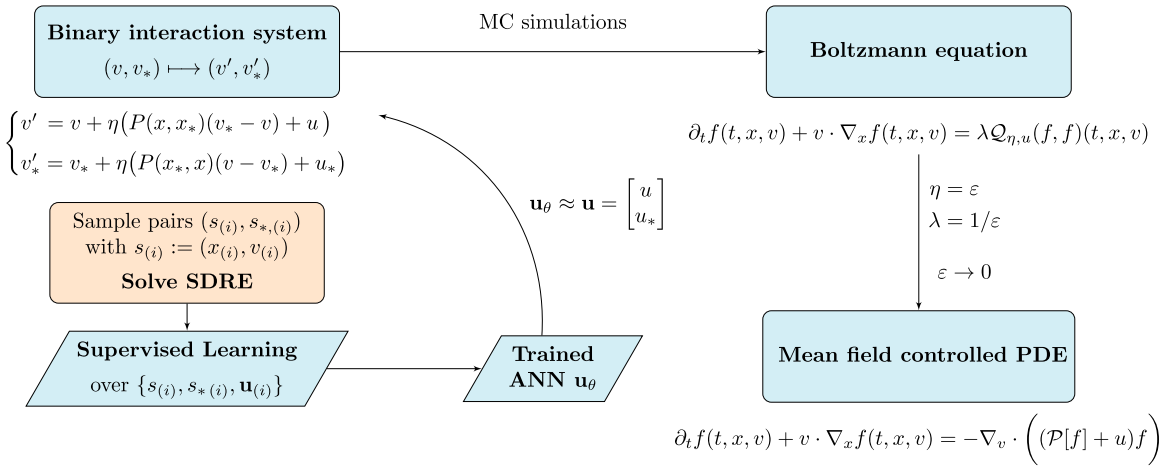


Fig. 1. A diagram summarizing the main building blocks of the proposed numerical method. The mean field optimal control problem is approximated by a kinetic equation, which relies on sampling interactions between pairs of agents. To reconstruct the complete microscopic dynamical system, these interaction equations are coupled with a free transport process (omitted here for simplicity). Since every sample requires the solution of a binary optimal control problem, we resort to supervised learning techniques to build a feedback control law which can be called in the sampling step without solving online optimization problems. We denote by $s_{(i)} := (x_{(i)}, v_{(i)})$ the state of the i -th agent.

the solution of the binary OCP, thus circumventing its online solution at every sampling instance. For training, we generate synthetic data via the solution of the binary OCP using a discrete time *state-dependent Riccati equation* (SDRE) approach [32,63]. The overall procedure is outlined in Fig. 1.

1.0.0.1. Related literature and contributions. Simulation and control of high-dimensional ABMs is a longstanding and challenging topic. The modeling of such systems has been extensively studied within the kinetic research community, aiming at reducing the computational complexity of simulations [12,13,24]. The formulation of a kinetic description for evolutionary models has inspired a flourishing literature in ABMs [16,49,58,59]. In this direction, the authors of [5] proposed a Monte Carlo approach based on binary collision dynamics [48], inspired from plasma physics. The proposed methodology addresses the integration of the mean field formulation of ABMs by means of a Boltzmann scheme, allowing to retrieve the mean field evolution of the system as a limit of 2-agents sub-problems. Such complexity reduction has motivated the extension of this numerical scheme to fit the optimal control framework [4,50,51].

In [2,4], consistency was shown between the feedback controlled, non-local, mean field (2) and the homogeneous non-linear Boltzmann (4) with controlled binary interactions in the quasi-invariant scaling, and where the control action has been optimized by means of *model predictive control* (MPC) [53], or via *dynamic programming* (DP) [11]. Similar approaches can be found also in [6] for leader-follower multi-agent systems. In the MPC case, the controller is to be considered sub-optimal, as it is designed to optimize the OCP up to a reduced horizon. DP instead, leads to optimal solutions for the binary OCP, but it requires the solution of a first-order nonlinear Hamilton-Jacobi-Bellman (HJB) PDE cast in the state-space of the system, which can be of arbitrarily high dimension.

Over the last years, the solution of high-dimensional HJB-PDEs has been addressed with a number of different numerical approaches [10,25,30,44,45], and a flourishing literature on Artificial Neural Networks (ANNs). ANN methods are differentiated between unsupervised learning techniques [21,39,40,43,55] and supervised ones [1,33,62]. Examples of deep learning algorithms for solving PDEs can be found in [61] for high dimensional Boltzmann equations, or in [36,56] for more general applications.

The main contributions of our work, inspired by the aforementioned results, can be summarized as follows:

- Aiming at a further reduction in computational complexity with respect to [2,4], we rely on data-driven approximation models fitted in a supervised learning fashion over synthetic data for the reduced binary OCP. For this, we consider candidate approximation models of both feed-forward (FNN) and recurrent (RNN) neural network type. The architecture of such fitted approximation models conveniently allows for batch evaluations of data, meaning that at each time step the controls can be computed for all the sampled 2-agents sub-systems at once. The precision of the fitted models and their efficiency as the dimensionality of the problem increases have been assessed through numerical tests.
- The dataset guiding the ANNs training phase is collected from synthetic data, obtained from the solution of the infinite horizon (OCP) reduced to the binary case. Unlike the approaches taken in [2], we circumvent the solution of the HJB equation associated to the binary OCP, which would be unfeasible even in moderate agent dimensions. Instead, we rely on a procedure for synthesizing nonlinear feedback controls that combines elements from both DP and MPC: the State-Dependent Riccati equation approach. In particular, we will focus on discrete time settings [18,27].
- A first, intuitive, choice is to consider the feedback control as target variable of the approximation task, as done in [1,26,33,35,53]. Furthermore, we also train networks for approximating the whole of the controlled right hand side of the discrete-time binary

controlled dynamics. For uncontrolled systems, approximation techniques has been studied for detection and approximation of interaction kernels $P(\cdot, \cdot)$ [14,37]. However, to our knowledge, this work has not been yet extended to the controlled framework.

The rest of the paper has been organized as follows. In Section 2 we construct an approximation for the sub-optimal control of the mean field PDE (2) from a kinetic viewpoint. This leads to a Boltzmann description of the dynamics (4), which is then proven to converge to its mean field counterpart when performing a proper scaling of the frequency and strength of interactions. In Section 3 we discuss how to approximate the evolution of system according to the Boltzmann dynamics for the distribution of agents via Monte Carlo simulation: from the current system configuration, we sample a pool of agents, which are then randomly coupled according to statistics of interaction. The post-interaction agents are then considered, so that their sampling distribution models the updated population density. Following this scheme, the evolution of the system is tracked at discrete times. This requires the computation of the feedback control in discrete-time settings, which is addressed in Section 4 by means of a discrete time state-dependent Riccati Equation approach. In Section 5, we discuss a supervised learning approximation to the solution of the binary OCP. Numerical tests are presented in Section 6.

2. A Boltzmann formulation for the control of mean field dynamics

The accurate modeling of self-organization phenomena and optimal control in ABMs requires a large number of interacting individuals, implying the need for the solution of a very high dimensional optimal control problem, which often comes at a prohibitively expensive computational cost. An alternative way to address this problem is to model instead the distribution function $f(t, x, v)$ describing the density of individuals having state variable $(x, v) \in \mathbb{R}^d \times \mathbb{R}^d$ at time $t \geq 0$. The evolution of $f(t, x, v)$ can be characterized by a kinetic equation accounting for the motion of individuals undergoing pairwise interactions, as modeled in (8). Thus, the mean field dynamics can be retrieved by suitable scaling of the interactions, also referred as *quasi-invariant scaling*, or grazing collision limit [48,60]. In particular, the quasi-invariant limit consists of considering an interaction regime where low intensity interactions occur with high frequency. In this regime, the density $f(t, \cdot, \cdot)$ is expected to converge pointwise in time to the solution of the mean field controlled PDE (2), which corresponds to a sub-optimal solution of the mean field optimal control problem (MFOC).

2.1. Binary controlled dynamics

We denote by $(x, v), (x_*, v_*) \in \mathbb{R}^d \times \mathbb{R}^d$ the position and velocity states of two agents in the population, and we assume that they modify their velocity states according to binary interaction maps $(v, v_*) \mapsto (v', v'_*)$ as follows

$$\begin{aligned} v' &= v + \eta P(x, x_*)(v_* - v) + \eta u, \\ v'_* &= v_* + \eta P(x_*, x)(v - v_*) + \eta u_*, \end{aligned} \quad (6)$$

where η is the strength of interaction, $\mathbf{u} = (u, u_*)^\top$ is the forcing term associated to interaction between agents. The goal of such external influence is to steer a couple of agents toward consensus. In particular, we will address the optimal control $\bar{u}$ as the solution of an infinite horizon binary optimal control problem, as follows

$$\bar{\mathbf{u}} = \operatorname{argmin}_{\mathbf{u} \in U_{\text{adm}}} \left\{ \int_0^{+\infty} \|v - \bar{v}\|^2 + \|v_* - \bar{v}_*\|^2 + \gamma (\mathcal{G}(u) + \mathcal{G}(u_*)) dt \right\}, \quad (7)$$

where $U_{\text{adm}} = L^\infty([0, +\infty[; \mathbb{R}^{2d})$ is the space of admissible controls. The numerical procedure for the feedback control synthesis will be addressed in Section 4.

Furthermore, we model the evolution in time of $f(t, x, v)$ with a kinetic integro-differential equation of Boltzmann type [2,47,52]

$$\partial_t f(t, x, v) + v \cdot \nabla_x f(t, x, v) = \lambda \mathcal{Q}_{\eta, u}(f, f)(t, x, v), \quad (8)$$

where the parameter λ encodes the interaction frequency, and the operator $\mathcal{Q}_{\eta, u}(f, f)$ accounts for the gain and loss of particles with state (x, v) at time t

$$\mathcal{Q}_{\eta, u}(f, f) = \mathcal{Q}_{\eta, u}^+(f, f) - \mathcal{Q}_{\eta, u}^-(f, f). \quad (9)$$

In particular, we can express respectively the gain and loss operators as follows

$$\mathcal{Q}_{\eta, u}^+(f, f)(t, x, v) = \int_{\mathbb{R}^d \times \mathbb{R}^d} \frac{1}{J_{\eta, u}} f(t, x, v') f(t, x_*, v'_*) dx_* dv_*, \quad (10a)$$

$$\mathcal{Q}_{\eta, u}^-(f, f)(t, x, v) = \int_{\mathbb{R}^d \times \mathbb{R}^d} f(t, x, v) f(t, x_*, v_*) dx_* dv_*, \quad (10b)$$

where $(v', v'_*) \mapsto (v, v_*)$ are the pre-interaction velocities that generate the couple (v, v_*) , while $J_{\eta, u}$ is the Jacobian of the binary interactions map (6). To avoid the presence of the jacobian we can introduce a test function $\varphi(x, v) \in C_0^2(\mathbb{R}^d \times \mathbb{R}^d)$ and express the Povzner-Boltzmann operator in the weak form as

$$\lambda \langle \mathcal{Q}_{\eta, u}(f, f), \varphi \rangle = \lambda \iint_{\mathbb{R}^{2d} \times \mathbb{R}^{2d}} (\varphi(x, v') - \varphi(x, v)) f f_* dx_* dv_* dx dv, \quad (11)$$

where v' denotes the post-interaction velocity as in (6), and we adopted the shorten notation $f = f(t, x, v)$, and $f_* = f(t, x_*, v_*)$.

Remark 1. Here we considered the unbounded domain in space and velocity, where the operator (9) accounts interactions with constant collision frequency λ , analogously to the Boltzmann equation for Maxwell molecules. In general, the interaction frequency is ruled by a non-linear kernel, where in the Povzner-Boltzmann model it reads as follows

$$\langle Q_{\eta,u}, \varphi \rangle = \iint_{\mathbb{R}^{2d} \times \mathbb{R}^{2d}} B(x, x_*, v, v_*) (\varphi(x, v') - \varphi(x, v)) f_* dx_* dv_* dx dv, \quad (12)$$

where the kernel $B(\cdot)$ in the Povzner approach [52] considers non-local interactions among particles.

If we consider a bounded domain, further conditions should be included in (8). Of particular interest is the case, when the velocity space is bounded, e.g. $v, v_* \in \Omega_v \subset \mathbb{R}^d$. In this situation, we can directly impose in (12) that the boundary is satisfied introducing a non-linear interaction kernel B such as $B \equiv \chi(v' \in \Omega_v) \chi(v'_* \in \Omega_v)$. This latter choice automatically satisfies the boundaries, however such nonlinear kernel causes major difficulties when we are interested in studying asymptotic properties of the model (8), such as mean field approximations. Alternatively, following the approach proposed in [57], one can ask that the discrete interaction (6) preserves the boundary. In this setting, such requirement is possible by properly designing the admissible space of controls U_{adm} in such a way that the boundary cannot be violated. We also observe that the control acts by forcing the state within an admissible position in the bounded domain. Thus, to preserve numerically the bounds a possible strategy is to require a sufficiently small penalty parameter $\gamma > 0$ in the cost functional (7), as for example shown in [59], where an explicit form of the control is obtained.

2.2. Consistency with the mean field formulation

Here we focus on the consistency of the Boltzmann operator (9) with a mean field controlled dynamics of the type (2), in particular introducing a *quasi-invariant optimality limit* we can regularize such operator, considering a regime where interactions strength is low and frequency is high. This technique, analogous to the grazing collision limit in plasma physics, has been thoroughly studied in [60] and specifically for first order models in [7,20], and allows to pass from Boltzmann (8) to a mean field equation [4,6]. In what follows we consider the change of notation for the control in the binary dynamics (6)

$$u \rightarrow u_\eta(x, v, x_*, v_*), \quad u_* \rightarrow u_\eta(x_*, v_*, x, v)$$

to give explicit dependence on the parameters and the state variables, since we focus on feedback type controls, and we introduce the following assumptions

- (i) the system (6) constitutes an invertible changes of variables from (v, w) to (v', w') ;
- (ii) there exists an integrable function $u(x, v, x_*, v_*)$ such that the following limit is well defined

$$\lim_{\eta \rightarrow 0} u_\eta(x, v, x_*, v_*) = u(x, v, x_*, v_*). \quad (13)$$

Hence, in accordance with [16] and with [2], where an analogous of this argument is furnished for the controlled dynamics in stochastic settings, we state the following theorem

Theorem 1. Consider the Boltzmann-type (8), with $\eta, \lambda > 0$ and the control $u_\eta \in U_{\text{adm}}$, where U_{adm} is the class the admissible controls. Furthermore, assume the kernel function $F_\eta(\cdot) \in L^2_{\text{loc}}(\mathbb{R}^{2d} \times \mathbb{R}^{2d})$ for all $\eta > 0$, where

$$F_\eta(x, v, x_*, v_*) = P(x, x_*)(v_* - v) + u_\eta(x, v, x_*, v_*),$$

and define the parameter $\varepsilon > 0$ to introduce the quasi-invariant scaling as follows

$$\eta = \varepsilon, \quad \lambda = 1/\varepsilon, \quad (14)$$

which links the strength and the frequency of the interactions in the Boltzmann-type (8). Thus, if $f^\varepsilon(x, v, t)$ is a solution for the scaled equation (8), for $\varepsilon \rightarrow 0$ $f^\varepsilon(t, x, v)$ converges pointwise, up to a subsequence, to $f(t, x, v)$ where f satisfies the following controlled mean field equation,

$$\partial_t f + v \cdot \nabla_x f = -\nabla_v \cdot ((P[f] + U[f])f) \quad (15)$$

with initial data $f^0(x, v) = f(0, x, v)$ and where $P[\cdot]$ represents the interaction kernel (3) and the control is such that

$$U[f](t, x, v) = \int_{\mathbb{R}^d \times \mathbb{R}^d} u(x, v, x_*, v_*) f(t, x_*, v_*) dx_* dv_* \quad (16)$$

where $u(x, v, x_*, v_*)$ is defined as the limiting value in (13).

We report the proof of this result in the Appendix A, as a reformulation of the result proposed in [2].

2.3. Full state controlled binary dynamics

A further generalization consists in introducing a binary interaction dynamics where, differently from the previous section, we consider a binary exchange of information for the full states of the agents $s = (x, v)^\top$, $s_* = (x_*, v_*)^\top \in \mathbb{R}^{2d}$. The interacting process defining $(s, s_*) \rightarrow (s', s'_*)$ is described by a binary rules of the following type

$$\begin{aligned} s' &= s + \eta G(s, s_*) + \eta H u \\ s'_* &= s_* + \eta G(s_*, s) + \eta H u_* \end{aligned} \quad (17)$$

where the controls are respectively $u = u(s, s_*)$, $u_* = u_*(s, s_*)$, and the operators G and H are defined accordingly to (6) as follows

$$G(s, s_*) = \begin{pmatrix} 0_d & \mathbb{I}_d \\ 0_d & -P(x, x_*)\mathbb{I}_d \end{pmatrix} \begin{pmatrix} x \\ v \end{pmatrix} + \begin{pmatrix} 0_d & 0_d \\ 0_d & P(x, x_*)\mathbb{I}_d \end{pmatrix} \begin{pmatrix} x_* \\ v_* \end{pmatrix}, \quad H = \begin{pmatrix} 0_d \\ \mathbb{I}_d \end{pmatrix}. \quad (18)$$

Hence, the kinetic density $f(t, s)$ evolves according to a Boltzmann-type model of the following type

$$\partial_t f(t, s) = \lambda \hat{Q}_{\eta, u}(f, f)(t, s), \quad (19)$$

with frequency $\lambda > 0$, and where, differently from the formulation in (8), the transport term is now encoded in the interaction operator $\hat{Q}(f, f)(t, s)$. The interaction operator $\hat{Q}(f, f)(t, s)$ is now defined as follows

$$\hat{Q}_{\eta, u}(f, f)(t, s) = \int_{\mathbb{R}^{2d}} \left(\frac{1}{\hat{J}_{\eta, u}} f(t, 's) f(t, 's_*) - f(t, s) f(t, s_*) \right) ds, \quad (20)$$

where $\hat{J}_{\eta, u}$ is the jacobian of the binary rule (17), with $('s, 's_*)$ the precollisional states.

In this framework, we can retrieve the consistency with mean field model (15) via Theorem 1, considering now as a kernel function the following

$$F_\eta(s, s_*) = \eta G(s, s_*) + \eta H u(s, s_*), \quad (21)$$

and assuming $F_\eta(\cdot) \in L^2_{loc}(\mathbb{R}^{2d} \times \mathbb{R}^{2d}; \mathbb{R}^{2d})$. Then, in the quasi-invariant scaling (14), and taking the limit $\varepsilon \rightarrow 0$ we expect pointwise convergence of the kinetic model (19) to the following mean field equation

$$\partial_t f(t, s) = -\nabla_s \cdot \left(f(t, s) \int_{\mathbb{R}} (G(s, s_*) + H u(s, s_*)) f(t, s_*) ds_* \right). \quad (22)$$

Recalling that $\nabla_s = (\nabla_x, \nabla_v)^\top$ and that the operators G, H are defined as in (18), we have that (22) is equivalent to the mean field (15). This result follows the same steps as the proof of Theorem (1), in this regard further details can be found in Appendix A. We stress that the main differences, with respect to the results of the previous section, consist in treating the transport term as part of the interaction dynamics, and requiring that (21) is L^2_{loc} . This last requirement, useful for the validity of the consistency Theorem 1, is in general more restrictive. Nonetheless the full state binary interaction (17), in a reformulated version, allow to provide an efficient control synthesis for the second-order dynamics. This aspect will be discussed in more details in Section 4.

3. Asymptotic monte carlo methods for constrained mean field dynamics

We provide a fast simulation numerical scheme for the Povzner-Boltzmann-type controlled equation, in the asymptotic regime, reminiscent of Direct Simulation Monte Carlo methods (DSMCs), used in plasma physics, [5,12,13] and later adapted to collective dynamics [5,57].

First of all, considering a splitting of the transport and collisional term of the Boltzmann-type (8) in the asymptotic regime in two different steps [5]:

$$\begin{cases} \partial_t f = -v \cdot \nabla_x f & \text{transport} \\ \partial_t f = \varepsilon^{-1} Q_{\eta, u}(f, f). & \text{interaction} \end{cases} \quad (23)$$

The purpose of the splitting scheme (23) is to focus the discussion on the convergence of the collisional term, since the free transport process coincides with the mean field formulation and the Boltzmann one.

In the previous section, consistency between the time evolution of the agents' population has been proven between the mean field model for the population density function, and a scaled Boltzmann description for the dynamics. The latter modeling of $f(t, x, v)$ is guided by the reduced microscopic binary interactions (6) between agents. This motivates the resorting to Monte Carlo simulation techniques for the approximation of the population density function under the Boltzmann formulation [2,5,47,48].

The evolution (23) of f in $[0, T]$ can be modeled by means of a Forward Euler method, for which we define a time step Δt , and discrete times $t_n = n\Delta t$ for $n = 0, \dots, N_T$

$$\partial_t f(t_n, x, v) \approx \frac{f(t_{n+1}, x, v) - f(t_n, x, v)}{\Delta t}. \quad (24)$$

In the spirit of DSMC methods, we can design a stochastic simulation scheme where we sample N_s agents from f^0 and we approximate the time variation of $\partial_t f$ as the evolution of the sampled distribution according to the post transport/interaction states.

Concerning the free transport process, the time evolution amounts to the exact free flow at time t_n of sample particles $\{(x_i^n, v_i^n)\}_{i=1}^{N_s}$ and evolves as follows

$$x_i^{n+1} = x_i^n + \Delta t v_i^n \quad i = 1, \dots, N_s. \quad (25)$$

The collisional term in the Boltzmann-type (23), can be rewritten in dependence of the gain and loss components of the operator $Q_{\varepsilon, u}$:

$$\partial_t f(t, x, v) = \frac{1}{\varepsilon} \left(Q_{\varepsilon, u}^+(f, f)(t, x, v) - \rho f(t, x, v) \right), \quad (26)$$

where $\rho > 0$ represent the total mass

$$\rho = \int_{\mathbb{R}^{2d}} f(t, x_*, v_*) dx_* dv_* . \quad (27)$$

Assuming f to be a probability density function we will consider $\rho = 1$. The Forward Euler scheme for (26), with the notation $f^n = f(t_n, x, v)$ reads

$$f^{n+1} = \left(1 - \frac{\Delta t}{\varepsilon}\right) f^n + \frac{\Delta t}{\varepsilon} Q_{\varepsilon, u}^+(f^n, f^n), \quad (28)$$

since f^n is a probability density, thanks to mass conservation $Q_{\varepsilon, u}^+$ is again probability density function. Moreover, under the restriction $\Delta t \leq \varepsilon$, also f^{n+1} is also a probability density. (28) can be interpret as follows: an agent with state (x, v) has probability $(1 - \Delta t/\varepsilon) \in [0, 1]$ to avoid collision with other agents in each time interval $[t, t + \Delta t]$. When, instead, the collision does happen (event with probability $\Delta t/\varepsilon \in [0, 1]$), the evolution follows the interaction law $Q_{\varepsilon, u}^+$ described by the scaled binary interaction, then a sampled pair of agents at time t_n

$$(x, v) := (x_i^n, v_i^n), \quad (x_*, v_*) := (x_j^n, v_j^n)$$

evolves according to (6) as follows

$$\begin{aligned} v_i^{n+1} &= v_i^n + \varepsilon P(x_i^n, x_j^n) (v_j^n - v_i^n) + \varepsilon u(x_i^n, v_i^n, x_j^n, v_j^n), \\ v_j^{n+1} &= v_j^n + \varepsilon P(x_j^n, x_i^n) (v_i^n - v_j^n) + \varepsilon u(x_j^n, v_j^n, x_i^n, v_i^n). \end{aligned} \quad (29)$$

In what follows we consider the asymptotic regime with $\Delta t = \varepsilon$, where the binary interactions (29) are equivalent to a Forward Euler scheme for the two agent dynamics, and the particles that interacts at each iteration are maximized according to the scheme (28). The choice of ε is of paramount importance to approximate consistently the mean-field model (15) according to Theorem 1. Nonetheless, there is a trade-off between the number of samples N_s and the size of ε , indeed it is possible to see that below a certain threshold $\varepsilon_{N_s}^*$ there is no further improvement in approaching the mean field model. We refer the reader to [5,48] for a detailed discussion of these methods, and to [15] for a rigorous convergence analysis of stochastic particle dynamics in this framework.

The asymptotic Monte-Carlo algorithm can be formalized as the procedure outlined in Algorithm 4, which is comparable to Algorithm 1 in [2]. This approach differs from the conventional Nanbu scheme in plasma physics, which was originally developed for free particle systems lacking external control inputs, and it does not require the introduction of a mesh.

Algorithm 1: MPC-DSDRE approach.

$\mathbf{z}(0) \leftarrow \mathbf{s}$	initial condition
for $n = 0, \dots, N_T$ do	
$\mathbf{s} \leftarrow \mathbf{z}(t_n)$	read current state
$\mathbf{A} \leftarrow \mathbf{A}(\mathbf{s}), \mathbf{B} \leftarrow \mathbf{B}(\mathbf{s})$	freeze the system accordingly
solve (45) for Π ;	
$\mathbf{u} \leftarrow -(\mathbf{R} + \mathbf{B}^\top \Pi \mathbf{B})^{-1} \mathbf{B}^\top \Pi \mathbf{A} \mathbf{s}$	approximated feedback control
$\mathbf{z}(t_{n+1}) \leftarrow \mathbf{s} + \Delta t(\mathbf{A} \mathbf{s} + \mathbf{B} \mathbf{u})$	the control system evolves for Δt

Up to this moment, the discussion overlooked the derivation of the forcing terms u in the interaction (29). These control variable is meant to be of feedback type, as they only depend on the current agents' states. Nevertheless, the discrete-time nature of the interaction map (29), embeds within the control variable a dependency on the time-step Δt , which in turn is related to the parameter ε , see for example [2,4]. In the following section, we address the solution of the discrete time formulation (25) and (29) of the binary interaction control problem.

4. Infinite horizon optimal control of binary dynamics

In this section we study the solution of the infinite horizon optimal control problem for the reduced 2-agent system.

We recall some fundamental notions on optimal control for discrete-time systems [41], to then present its numerical approximation using a discrete-time State-Dependent Riccati Equation approach.

We begin by reformulating (29) in a general control-affine discrete time systems of the form:

$$\mathbf{z}(t_{n+1}) = \mathbf{A}(\mathbf{z}(t_n)) + \mathbf{B}(\mathbf{z}(t_n)) \mathbf{u}(t_n), \quad (30)$$

where $\mathbf{z}(t_n) \in \mathbb{R}^\kappa$ and $\mathbf{u}(t_n) \in \mathbb{R}^\mu$ denote the state of the system and the control signal at time $t_n = n\Delta t$, such that $\mathbf{z}(t_n) := (s^n, s_q^n)^\top \in \mathbb{R}^\kappa$, with $\kappa = d$ or $2d$ depending on whether the agent dynamics are of first or second order, respectively. The control vector is defined as $\mathbf{u}(t_n) := (u^n, u_q^n)^\top \in \mathbb{R}^\mu$, with $\mu = 2d$.

The state-to-state map $\mathcal{A}(\mathbf{z}) : \mathbb{R}^K \rightarrow \mathbb{R}^\mu$ and the control operator $B(\mathbf{z}) : \mathbb{R}^K \rightarrow \mathbb{R}^{K \times \mu}$ are assumed to be $C^1(\mathbb{R}^K)$, satisfying $\mathcal{A}(\mathbf{0}_K) = \mathbf{0}_K$ and $B(\mathbf{0}_K) = \mathbf{0}_{K \times \mu}$. Note that binary systems of the form (29) fit this setting. Given $Q \in \mathbb{R}^{K \times K}$, $Q \geq 0$ and $R \in \mathbb{R}^{\mu \times \mu}$, $R > 0$, we are interested in the infinite horizon optimal control problem

$$\min_{\mathbf{u}(\cdot)} \mathcal{J}(\mathbf{u}; \mathbf{s}) := \sum_{n=0}^{+\infty} \mathbf{z}(t_n)^\top Q \mathbf{z}(t_n) + \mathbf{u}(t_n)^\top R \mathbf{u}(t_n), \quad (31)$$

subject to the dynamics (30) with initial state $\mathbf{z}(t_0) = \mathbf{s}$. We look for a solution $\tilde{\mathbf{u}}$ to (31) in feedback form, that is, an optimal control map $\tilde{\mathbf{u}} : \mathbb{R}^K \rightarrow \mathbb{R}^\mu$ which is expressed a function of the state, $\tilde{\mathbf{u}}(t_n) = \tilde{\mathbf{u}}(\mathbf{z}(t_n))$. The computation of an optimal feedback law follows a dynamic programming argument, for which we define $V(\mathbf{s}) : \mathbb{R}^K \rightarrow \mathbb{R}$ as the optimal cost-to-go departing from $\mathbf{s}$:

$$V(\mathbf{s}) = \min_{\mathbf{u}(\cdot)} \mathcal{J}(\mathbf{u}; \mathbf{s}), \quad (32)$$

where V satisfies the Bellman equation

$$V(\mathbf{s}) = \min_{\mathbf{u} \in \mathbb{R}^\mu} \left(\mathbf{s}^\top Q \mathbf{s} + \mathbf{u}^\top R \mathbf{u} + V(\mathcal{A}(\mathbf{s}) + B(\mathbf{s})\mathbf{u}) \right), \quad \text{for all } \mathbf{s} \in \mathbb{R}^K. \quad (33)$$

From this point onwards, as we work globally in the state space, $\mathbf{z}$ and $\mathbf{s}$ are treated indistinctly.

In order to better describe the difficulties related to the solution of this optimal control problem, we first focus on the linear quadratic case, where the optimal solution is computed via the discrete-time linear quadratic regulator (LQR). The optimality of the feedback law is ensured by a direct link between the LQR solution and the dynamic programming one, obtained from (33). In non-linear settings, the parallelism between the nonlinear QR and DP is broken, together with the optimality of the solution.

4.1. Discrete-time linear quadratic regulator

The discrete time linear quadratic problem is a particular instance of the optimal control problem (31) when $\mathcal{A}(\mathbf{s}) = A\mathbf{s}$, $A \in \mathbb{R}^{K \times K}$, and $B(\mathbf{s}) = B \in \mathbb{R}^{K \times \mu}$. Under these assumptions, we make the ansatz $V(\mathbf{s}) = \mathbf{s}^\top \Pi \mathbf{s}$, so that the Bellman (33) becomes

$$\mathbf{s}^\top \Pi \mathbf{s} = \min_{\mathbf{u} \in \mathbb{R}^\mu} \left(\mathbf{s}^\top Q \mathbf{s} + \mathbf{u}^\top R \mathbf{u} + (A\mathbf{s} + B\mathbf{u})^\top \Pi (A\mathbf{s} + B\mathbf{u}) \right) \quad \text{for all } \mathbf{s} \in \mathbb{R}^K. \quad (34)$$

Solving the equation above leads to an optimal feedback of the form

$$\tilde{\mathbf{u}} = \tilde{\mathbf{u}}(\mathbf{s}) = -(R + B^\top \Pi B)^{-1} B^\top \Pi A \mathbf{s}, \quad (35)$$

where $\Pi \in \mathbb{R}^{K \times K}$ is the unique positive definite solution of the discrete-time algebraic Riccati equation (DARE):

$$\Pi = Q + A^\top \Pi A - A^\top \Pi B (R + B^\top \Pi B)^{-1} B^\top \Pi A. \quad (36)$$

Remark 2. Consider a discrete first-order binary system as in (29) with constant interaction kernel $P(x, x_*) = p$, with individual states $v, v_* \in \mathbb{R}^d$, target $\tilde{v} = 0$ and $\varepsilon = \Delta t$. In this case, the optimal control problem (31) reads

$$\min_{\mathbf{u}(\cdot)} \Delta t \sum_{n=0}^{+\infty} \|v^n\|^2 + \|v_*^n\|^2 + \gamma (\|u^n\|^2 + \|u_*^n\|^2) \quad (37)$$

$$\text{s.t.} \quad \begin{cases} v^{n+1} &= v^n + \Delta t (p(v_*^n - v^n) + u^n) \\ v_*^{n+1} &= v_*^n + \Delta t (p(v^n - v_*^n) + u_*^n). \end{cases} \quad (38)$$

This formulation corresponds to a linear-quadratic control problem by setting $\mathbf{s} = (v, v_*)^\top$, $\mathbf{u} = (u, u_*)^\top$ and

$$A = \mathbb{I}_{2d} + \Delta t \tilde{A}, \quad \tilde{A} = \begin{pmatrix} -p & p \\ p & -p \end{pmatrix} \otimes \mathbb{I}_d, \quad B = \Delta t \mathbb{I}_{2d} \quad Q = \Delta t \mathbb{I}_{2d}, \quad R = \Delta t \gamma \mathbb{I}_{2d}.$$

Similarly as in Theorem 1, we are interested in the asymptotic limit for $\Delta t \rightarrow 0$, in which case (36) becomes

$$2\tilde{A}\Pi - \frac{1}{\gamma}\Pi^2 + \mathbb{I}_{2d} = \mathbb{O}_{2d}. \quad (39)$$

Exploiting to the symmetric structure of the two agent interaction, Π can be reduced to diagonal and off-diagonal components, see e.g. [3,31], obtaining the following

$$\Pi = \begin{pmatrix} \pi_D & \pi_O \\ \pi_O & \pi_D \end{pmatrix} \otimes \mathbb{I}_d, \quad \pi_D = \frac{\sqrt{\gamma}}{2} \left(1 - 2\sqrt{\gamma}p + \sqrt{1 + 4\gamma p^2} \right), \quad \pi_O = \sqrt{\gamma} - \pi_D. \quad (40)$$

The limiting optimal feedback law $\tilde{\mathbf{u}}(\mathbf{s}) = (\tilde{u}, \tilde{u}_*)$ is given by

$$\tilde{\mathbf{u}}(\mathbf{s}) = -\frac{1}{\gamma} \Pi \mathbf{s} = -\frac{1}{\gamma} \begin{pmatrix} \pi_D v + \pi_O v_* \\ \pi_O v + \pi_D v_* \end{pmatrix}, \quad (41)$$

from where it follows that the individual feedback laws are given by

$$\tilde{u}(v, v_*) = -\frac{1}{\gamma}(\pi_D v + \pi_O v_*) = \tilde{u}_*(v_*, v),$$

Equivalently, notice that the discretized control computed in (35) in this case reads as follows

$$\tilde{\mathbf{u}}_{\Delta t}(\mathbf{s}) = -(\Delta t \gamma \mathbb{I}_{2d} + \Delta t^2 \Pi)^{-1} \Delta t \Pi(\mathbb{I}_{2d} + \Delta t \tilde{A})\mathbf{s}$$

and in the limit for $\Delta t \rightarrow 0$ we obtain exactly the same expression as in (41).

In the limiting kinetic equation this corresponds to the mean field control model

$$\partial_t f + v \cdot \nabla_x f = -\nabla_v \cdot \left(f \int_{\mathbb{R}^{2d}} (p(v_* - v) - \frac{1}{\gamma}(\pi_D v + \pi_O v_*)) f(t, x_*, v_*) dx_* dv_* \right). \quad (42)$$

Notice that the structure of $\Pi \in \mathbb{R}^{2d \times 2d}$ in (40) implies isotropic action of the control in the d -dimensional state space.

4.2. Discrete-time non-linear quadratic regulator

It is feasible to apply LQR to a linearization of the system (38). However, if we want to compute a law accounting for nonlinearities, then we compromise the convenient connection between the DARE and the Bellman equation. Moreover, the computational cost required for the solution consistently increases.

In order to extend LQR to the nonlinear case, we begin by re-arranging the difference (30) in semilinear state-dependent form

$$\mathbf{z}(t_{n+1}) = A(\mathbf{z}(t_n))\mathbf{z}(t_n) + B(\mathbf{z}(t_n))\mathbf{u}(t_n), \quad (43)$$

where $A(\mathbf{z}) = A(\mathbf{z})\mathbf{z}$ and $B(\mathbf{z}) = B(\mathbf{z})$. Note that this semi-linearization is not unique for systems of order greater than 1. In what follows, we further assume pointwise controllability, i.e. $\forall \mathbf{z} \in \Omega$ the pair $(A(\mathbf{z}), B(\mathbf{z}))$ is controllable.

Similarly to the LQR design, the feedback control policy can be calculated as

$$\begin{aligned} \tilde{\mathbf{u}}(\mathbf{z}(t_n)) &= -K(\mathbf{z})\mathbf{z} \\ K(\mathbf{z}) &= (B(\mathbf{z})^\top \Pi(\mathbf{z}) B(\mathbf{z}) + R)^{-1} B(\mathbf{z})^\top \Pi(\mathbf{z}) A(\mathbf{z}) \end{aligned} \quad (44)$$

where the argument $\mathbf{z}$ for the operators denotes dependency of the current state $\mathbf{z}(t_n)$ with discrete time scale Δt , and $\Pi(\mathbf{z})$ is the solution of a DARE with state-dependent coefficients (DSDRE):

$$\Pi(\mathbf{z}) = Q + A(\mathbf{z})^\top \Pi(\mathbf{z}) A(\mathbf{z}) - A(\mathbf{z})^\top \Pi(\mathbf{z}) B(\mathbf{z}) (R + B(\mathbf{z})^\top \Pi(\mathbf{z}) B(\mathbf{z}))^{-1} B(\mathbf{z})^\top \Pi(\mathbf{z}) A(\mathbf{z}). \quad (45)$$

As discussed in [65], similarly to the continuous-time case, the discrete-time SDRE is derived by assuming a quadratic ansatz for the value function in the Bellman equation and by freezing the matrix $\Pi(\mathbf{z})$ at the current state, thus neglecting its variation between time steps. The state dependency of the feedback operator $K(\mathbf{z}) = K(\mathbf{z}(t_n))$ suggests the need for consecutive sequential solutions $\Pi(\mathbf{z}) = \Pi(\mathbf{z}(t_n))$ of (45) at every discrete time t_n along a trajectory. Thus, for high dimensions of the state space $\Omega \subset \mathbb{R}^n$, the exact solution of the DSDRE (45) comes at a cumbersome computational cost. Between the several numerical approaches that have been proposed to address this, we cite Taylor series method [9] and interpolation for $\Pi(\mathbf{z})$ [32], while we refer the interested reader to [42] for a more exhaustive review. In what follows, we rely on a discrete version of the SDRE approach proposed in [32].

4.3. Discrete-time SDRE approach

We aim at circumventing the computational challenge of solving (45) at each time step, by realizing the DSDRE feedback law in a model predictive control fashion: given the current state $\mathbf{z}(t_n)$ of the system, we assume the operator $\Pi(\mathbf{z})$ to be a positive definite matrix in $\Pi \in \mathbb{R}^{n \times n}$, meaning that (45) reduces to its algebraic form (36), where the state dependencies in (45) are neglected by accordingly freezing all the operators at the current configuration. The resulting feedback variable leads to a suboptimal approximation of the controlled trajectory between t_n and t_{n+1} , after which the procedure is repeated by freezing the system at the updated state $\mathbf{z}(t_{n+1})$.

The main computational bottleneck still persisting with this approach is the availability of a sufficiently fast solver for (36). Efficiency in retrieving the DARE solution is key in the settings under consideration. The consistency between the mean field and the kinetic dynamics only holds for $\varepsilon \ll 1$ which in turn is associated with high frequency sampling, i.e. large numbers of couple of interacting particles in the Monte Carlo simulation. At each time step, we aim at conveniently approximate the controlled post-interaction states for a large number of paired agents, living in an arbitrarily high dimensional state space $\Omega \subset \mathbb{R}^n$. We address this task by means of supervised learning approximation, relying on models within the family of Artificial Neural Networks.

5. Neural networks and supervised learning approximation of feedback laws

The computational method proposed in this paper models the time evolution of the population density $f(t, x, v)$ according to the Forward Euler scheme (28), where the updated density is to be approximated by the sampling distribution of *controlled* post-interaction states in a Monte Carlo fashion.

In this section, we provide an efficient approximating feedback map for the controlled binary dynamics via Feedforward and Recurrent Neural Networks (FNN/RNN). We briefly define these architectures and supervised learning framework for training [23]. After that, we will focus on the problem of interest, discussing synthetic data generation for feedback laws.

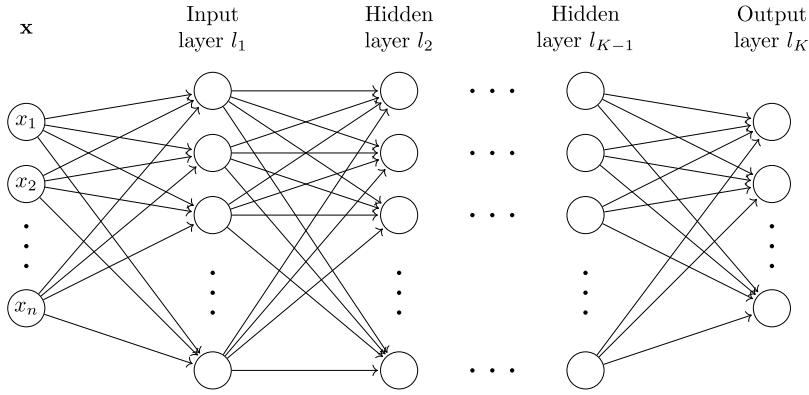


Fig. 2. A fully connected feedforward neural network, visualization of equation (46).

5.1. Feedforward neural networks

These architectures approximate general functions ψ through a sequence of layered transformations $l_1, \dots, l_K$, resulting in $\tilde{\psi} = \psi_\theta \approx \psi$ where

$$\psi_\theta(\mathbf{x}) = l_K \circ \dots \circ l_2 \circ l_1(\mathbf{x}), \quad l_k(\mathbf{x}) = \sigma_k(\mathbf{W}_k \mathbf{x}_k + \mathbf{b}_k), \quad (46)$$

with information flowing from the input nodes to the output ones in a unidirectional path, avoiding any cycles or loops, as shown in Fig. 2. Each layer applies a nonlinear activation function σ_k element-wise to a linear transformation of its input $\mathbf{x}_k$. Assuming the k -th layer to have n_k neurons, the parameters $\mathbf{W}_k \in \mathbb{R}^{n_{k-1} \times n_k}$ and $\mathbf{b}_k \in \mathbb{R}^{n_k}$ represent the weight matrix and bias vector for the k -th layer, respectively. We use the same activation function for all neurons within a given layer, with $\sigma_1 = \sigma_K$ being the identity function.

In a supervised learning environment, the trainable parameters $\theta = \{\mathbf{W}_k, \mathbf{b}_k\}_{k=1}^K$ are then to be computed as minimizers of a suitable *loss function* measuring the approximation error within a set of sampled data (*training set*) $\mathcal{T} = \{\mathbf{x}_{(i)}, \psi_{(i)}\}_{i=1}^{N_s}$ where $\psi_{(i)} := \psi(\mathbf{x}_{(i)})$:

$$\theta^* = \operatorname{argmin}_{\theta} \mathcal{L}(\psi, \psi_\theta), \quad \mathcal{L}(\psi, \psi_\theta) = \sum_{\mathcal{T}} l(\psi_{(i)}, \psi_\theta(\mathbf{x}_{(i)})). \quad (47)$$

In particular, we consider

$$\mathcal{L}(\psi, \psi_\theta) = MSE(\psi, \psi_\theta), \quad l(\psi_{(i)}, \psi_\theta(\mathbf{x}_{(i)})) = \frac{\|\psi_{(i)} - \psi_\theta(\mathbf{x}_{(i)})\|^2}{N_s}. \quad (48)$$

where we denoted by MSE the mean squared error as follows

$$MSE(\psi, \psi_\theta) = \frac{1}{N_t} \sum_{\mathcal{T}_v} \frac{\|\psi_{(j)} - \psi_\theta(\mathbf{x}_{(j)})\|^2}{\|\psi_{(j)}\|^2}.$$

The number of layers K , the number of neurons per layer, the set of activation functions $\{\sigma_k\}_{k=2}^{K-1}$ in the hidden layers, and eventual other parameters represent further degrees of freedom defining the *architecture* Θ of the FNN. They are to be computed as result of the *hyper-parameter tuning phase*, which amounts to choose, between trained models $\psi_\theta[\Theta]$ having different architectures, the one performing better to the error measure

$$\Theta^* = \operatorname{argmin}_{\Theta} \sqrt{MSE(\psi, \psi_\theta[\Theta])}, \quad (49)$$

where $\mathcal{T}_v = \{\mathbf{x}_{(j)}, \psi_{(j)}\}_{j=1}^{N_v}$ is the *validation set* such that $\mathcal{T} \not\subseteq \mathcal{T}_v$.

5.2. Recurrent neural networks

Recurrent neural networks are a class of models which can be obtained from FNNs by allowing loop connections between layers. This architecture is particularly well-suited for processing sequential data. However, given that our input lacks temporal structure, we focus on *one-to-one* architectures.

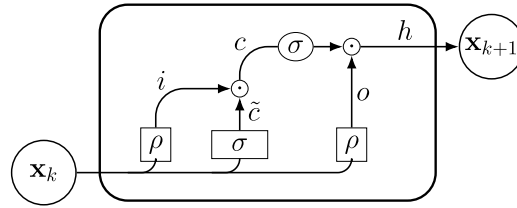


Fig. 3. One-to-one LSTM cell, visualization of equation (50).

Here we extend FNNs into RNNs by allowing hidden layers to be one-to-one *Long Short Term Memory* (LSTM) cells which – for every input $\mathbf{x}$ – generate output h according to the following system of equations

$$\begin{aligned}
 i &= \rho(W_i \mathbf{x} + b_i) && \text{inputgate} \\
 \tilde{c} &= \sigma(W_c \mathbf{x} + b_c) && \text{candidatevalue} \\
 c &= i \odot \tilde{c} && \text{cellvalue} \\
 o &= \rho(W_o \mathbf{x} + b_o) && \text{outputgate} \\
 h &= o \odot \sigma(c) && \text{finaloutput}
 \end{aligned} \tag{50}$$

where (assuming $\mathbf{x}_k \in \mathbb{R}^{n_k}$), the weight matrices $W_i, W_c, W_o \in \mathbb{R}^{n_k \times n_{k+1}}$ and the bias vectors $b_i, b_c, b_o \in \mathbb{R}^{n_{k+1}}$ are the trainable parameters, $\sigma(\cdot)$ is the activation function, and $\rho(\cdot)$ is the recurrent activation function. The number κ denotes the number of neurons in the cell. A visual representation of the flux of information through a LSTM cell can be found in Fig. 3.

The training and architecture selection for this kind of RNNs consists of the same procedure described in (47) and (49), with the exception that the k -th layer is replaced by a LSTM cell, mapping input $\mathbf{x}_k$ to output $h = \mathbf{x}_{k+1}$, with parameters $\mathbf{W}_k = (W_i, W_c, W_o)^\top$ and $\mathbf{b}_k = (b_i, b_c, b_o)^\top$.

5.3. Synthetic data generation

Both the training and hyper-parameter tuning phases for the neural network approximation rely on the availability of datasets containing input data points coupled with the corresponding target function evaluations. Since our goal is to approximate the feedback law for the reduced binary controlled dynamics, we consider two distinct approaches:

Control approximation: The first approach treats the feedback control law $\mathbf{u}$, i.e. we build a network $\mathbf{u}_\theta(\mathbf{s}, \Delta t) \approx \tilde{\mathbf{u}}(\mathbf{s}, \Delta t)$, where $\tilde{\mathbf{u}}$ is the optimal control obtained from the SDRE approach described in Section 4.

State update approximation: In this case, the neural network directly learns the mapping from current states to their controlled updates, i.e., $\mathbf{s}'_\theta(\mathbf{s}, \Delta t) \approx \mathcal{A}(\mathbf{s}) + \mathcal{B}(\mathbf{s})\tilde{\mathbf{u}}(\mathbf{s}, \Delta t)$, bypassing the explicit computation of the control law.

The training loss for both approaches uses the mean squared error (MSE) as defined in (47). In Section 6, we compare these two approaches in numerical tests, examining their precision and computational efficiency.

To create training data and establish a benchmark for precision, we solve the optimal control problem using the discrete-time SDRE approach. The difference equations for the binary system at the discrete time t_n hold as follows:

$$\begin{cases}
 x^{n+1} = x^n + \Delta t v^n \\
 x_*^{n+1} = x_*^n + \Delta t v_*^n \\
 v^{n+1} = v^n + \Delta t [P(x^n, x_*^n)(v_*^n - v^n) + u(x^n, x_*^n, v^n, v_*^n)] \\
 v_*^{n+1} = v_*^n + \Delta t [P(x_*^n, x^n)(v^n - v_*^n) + u_*(x^n, x_*^n, v^n, v_*^n)]
 \end{cases} \tag{51}$$

being $(x, v), (x_*, v_*) \in \Omega \subset \mathbb{R}^{2d}$ the states for a couple of interacting agents. We introduce the notation $\mathbf{s} = (x, x_*, v, v_*)^\top \in \mathbb{R}^{4d}$, $\mathbf{u} = (u, u_*)^\top \in \mathbb{R}^{2d}$, for which the difference equation reads as in (30)

$$\mathbf{s}^{n+1} = \mathcal{A}(\mathbf{s}^n) + \mathcal{B}(\mathbf{s}^n)\mathbf{u}^n, \tag{52}$$

for $\mathcal{A}(\mathbf{s}) = (\mathbb{I}_{4d} + \Delta t G(\mathbf{s}))\mathbf{s}$ and $\mathcal{B}(\mathbf{s}) = \Delta t H(\mathbf{s})$, with G and H defined as in (18). From this, comes the Δt -dependency of the feedback law $\mathbf{u}$: the DSDRE parameters $\mathcal{A}, \mathcal{B}$ are dependent of the time-step, as will be the DARE solution associated to the system frozen at the current configuration.

The generation of datasets $\mathcal{T}_u, \mathcal{T}_{s'}$ for the supervised learning approximation of the feedback law $\mathbf{u}_\theta$, and post-interaction states $\mathbf{s}'_\theta$ is summarized in Algorithm 2. The number N_s of samples in the datasets is discussed in Section 6 specifically for each numerical test, whilst the reliability and generalizability of the sampled data are ensured through cross validation of the trained models over an unseen *test set* $\mathcal{T}_t \not\subseteq \mathcal{T} \cup \mathcal{T}_v$ also generated through Algorithm 2. The addition of the time step Δt to the models' input allows to train the networks to provide accurate approximations even when applied to adaptive time-step integration techniques.

Algorithm 2: Synthetic data generation.

```

1   $\{s_{(i)}, \Delta t_{(i)}\}_{i=1}^{N_s}$  i.i.d.  $N_s$  uniform samples in  $\Omega^2 \times [0, T]$ 
2  for  $i = 1, \dots, N_s$  do
3       $A \leftarrow \mathbb{I}_{4d} + \Delta t_{(i)} G(s_{(i)});$ 
4       $B \leftarrow \Delta t_{(i)} H(s_{(i)})$   $G, H$  as in (18) solve (45) for  $\Pi$ ;
5       $u_{(i)} \leftarrow -(R + B^T \Pi B)^{-1} B^T \Pi A s_{(i)}$  feedback control
6       $s'_{(i)} \leftarrow s_{(i)} + \Delta t_{(i)} (A s_{(i)} + B u_{(i)})$  post-interaction state
7
8   $\mathcal{T}_u = \{s_{(i)}, \Delta t_{(i)}, u_{(i)}\}_{i=1}^{N_s}$  dataset for  $u_\theta$ 
9   $\mathcal{T}_{s'} = \{s_{(i)}, \Delta t_{(i)}, s'_{(i)}\}_{i=1}^{N_s}$  dataset for  $s'_\theta$ 

```

Remark 3. As the controlled density evolution is carried by a Monte Carlo type method, the computational efficiency of evaluating a batch of controlled particle interactions is crucial. Our choice of control/state architectures is motivated by its favorable computational complexity. The evaluation cost of a single point through FNNs is $O(n)$, n being the number of neurons in the network. Furthermore, in our non-sequential framework, the evaluation cost of a one-to-one LSTM cell with n neurons is comparable to the one of a FNN with $4n$ neurons. This linear scaling in both architectures ensures efficient batch evaluation, as the computational cost grows proportionally with network complexity. This efficiency is particularly advantageous for large-batch evaluation, associated with $\varepsilon \ll 1$.

6. Numerical tests

In this section we will assess the proposed methodology over consensus control problems for two high dimensional (both in d and in N) ABMs. We aim at modeling the time evolution of the agents' distribution $f(t, x, v)$ via Monte Carlo simulation of the approximated binary post interaction states. In this section we will compare two different approaches: the approximation of the feedback control map u_θ , acting in the binary system (51), and the direct approximation of the controlled dynamics s'_θ . Since the approximation of the post-interactions positions s'_x is direct, the neural network model is restricted to post-interaction velocities s'_v , that is $s'_\theta = (s'_x, s'_v)^\top$. Algorithm 3 [5] summarizes the proposed numerical procedure for the density evolution once an approximated optimal feedback law $u_\theta \approx \bar{u}$ has been constructed. When a neural network s'_θ is built, we replace line 6 in the algorithm with

Algorithm 3: Monte Carlo simulation for Boltzmann dynamics.

```

1   $\{(x, v)_i\}_{i=1}^N \sim f(t_0, x, v)$  i.i.d.  $N$  samples from the initial distribution
2  for  $n = 0, \dots, T$ ,  $t_n = n \cdot \Delta t$  do
3      select  $\{(i_k, j_k)\}_{k=1}^{N//2}$  random pairs of agents without repetitions
4       $\{s_k^n\}_{k=1}^{N//2}, s_k^n = (x_{i_k}^n, x_{j_k}^n, v_{i_k}^n, v_{j_k}^n)$   $N//2$  couples of agents
5      for  $k = 1, \dots, N//2$  do
6           $s_k^{n+1} \leftarrow \mathcal{A}(s_k^n) + \mathcal{B}(s_k^n) \bar{u}(s_k^n, \Delta t)$  controlled dynamics (51)
7           $(x_{i_k}^{n+1}, v_{i_k}^{n+1}, x_{j_k}^{n+1}, v_{j_k}^{n+1}) \leftarrow s_k^{n+1};$ 

```

$$s_k^{n+1} \leftarrow s'_\theta(s_k^n). \quad (53)$$

The assumption of constant time-step Δt can be easily relaxed, as both the approximants u_θ and s'_θ take Δt as input. In the following numerical tests, the NN training has been done via *Adam* optimizer [34] with a learning rate $\alpha = 0.01$ over batches of 100 samples.

6.1. Test 1: Sznajd model

To illustrate the relationship between the control strategy derived at the supervised learning kinetic approximation level and its mean-field counterpart, we consider a preliminary numerical example in a simplified setting. We focus on consensus control for a one-dimensional first-order Sznajd model for opinion dynamics [64]. At the mean-field level, the control problem reads as a first-order version of (MFOC):

$$\begin{aligned}
 \min_u \mathcal{J}(f, u) &:= \int_0^T \int_\Omega (\|x - \bar{x}\|^2 + \gamma \|u\|^2) f(t, x_*) dx dt, \\
 \text{subject to } \partial_t f + \partial_x \left[(\mathcal{P}[f] + u) f \right] &= 0,
 \end{aligned} \quad (54)$$

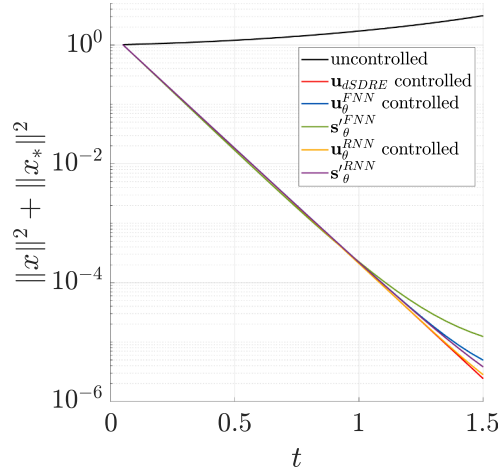


Fig. 4. Convergence to consensus for two randomly sampled interacting agents with states $x, x_* \in \Omega$, when controlled via dSDRE feedback law vs. the approximation models.

where $\bar{x}$ denotes the target configuration, and the interaction operator is defined as

$$P[f](t, x) = \int_{\Omega} P(x, x_*) (x_* - x) f(t, x) dx_*, \quad P(x, x_*) = \beta(1 - x^2), \quad (55)$$

for $\beta \in \mathbb{R}$. In particular, we consider $\Omega = [-1, 1]$, representing the opinion space of a population of voters ranging between two extreme positions represented by $\{-1, 1\}$, and a negative interaction coefficient $\beta = -1$, which induces polarization behavior. The control energy penalization coefficient is fixed to $\gamma = 0.05$.

This simplified framework enables the numerical solution of the mean-field optimal control problem, which would be computationally prohibitive in the higher-dimensional, second-order models considered in the subsequent numerical tests.

At the binary interaction level, the problem reads

$$\min_{\mathbf{u}(\cdot)} \frac{\Delta t}{2} \sum_{n=0}^{+\infty} (\|\mathbf{x}^n\|^2 + \gamma \|\mathbf{u}^n\|^2), \quad \text{s.t.} \quad \begin{cases} x^{n+1} = x^n + \Delta t (\beta(1 - (x^n)^2)(x_* - x) + u^n) \\ x_*^{n+1} = x_*^n + \Delta t (\beta(1 - (x_*^n)^2)(x - x_*) + u_*^n) \end{cases} \quad (56)$$

for $\mathbf{u}^n := (u^n, u_*^n)^\top$ and $\mathbf{x} = (x^n, x_*^n)^\top$. The synthetic data generation procedure associates $N_s = 10^5$ uniform samples of interacting couples of agents $\mathbf{x}_{(i)}^n \in \Omega$ to their associated dSDRE feedback control $\mathbf{u}_{(i)}^n$ and the controlled state update $\mathbf{x}_{(i)}^{n+1}$ obtained with interaction strenght $\varepsilon = 0.01$, for $i = 1, \dots, N_s$. With a ratio of 80/20, we designate those samples to form the training set $\mathcal{T}$ and the validation set $\mathcal{T}_v$ respectively. We test the following architectures:

u_θ^{FNN} : $K = 1$ hidden layers with 100 neurons, and $\sigma(\mathbf{x}) = \log(\exp(\mathbf{x}) + 1)$

u_θ^{RNN} : $K = 1$ LSTM cell with 100 neurons, $\sigma(\mathbf{x}) = \tanh(\mathbf{x})$, $\rho(\mathbf{x}) = (1 + \exp(-\mathbf{x}))^{-1}$

$\mathbf{x}'_\theta^{FNN}$: $K = 2$ hidden layers with 60 neurons per layer, and $\sigma(\mathbf{x}) = \tanh(\mathbf{x})$

$\mathbf{x}'_\theta^{RNN}$: $K = 1$ LSTM cell with 100 neurons, $\sigma(\mathbf{x}) = \tanh(\mathbf{x})$, $\rho(\mathbf{x}) = \log(\exp(\mathbf{x}) + 1)$

The error comparison for the binary-controlled evolution of a randomly selected pair of interacting agents is presented in Fig. 4, where all approximation models demonstrate similar performance to the dSDRE control.

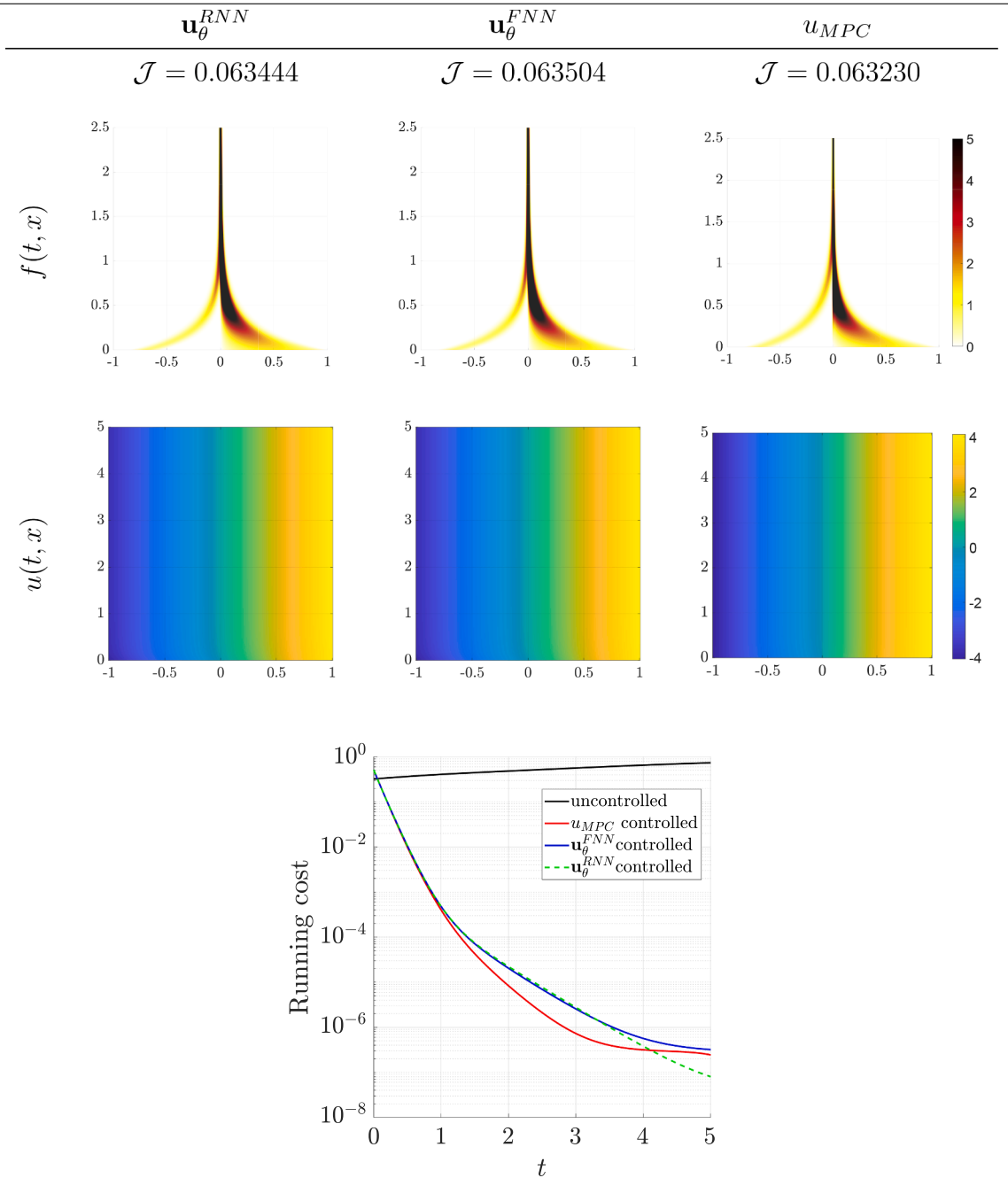
We conclude this preliminary example by quantifying the performance gap between the solution of (54) and its neural network-accelerated kinetic approximation, both implemented with a time step $\Delta t = \varepsilon = 0.01$. This comparison offers a more concrete understanding of the suboptimality of the control strategy at the kinetic level with respect to the MFOC. The approximation of the mean field control law under the numerical procedure proposed in this paper is retrieved as

$$\int_{\Omega} \mathbf{u}_\theta(x, x_*) f(t, x_*) dx_* \approx \mathbf{u}(t, x). \quad (57)$$

In Table 1, we compare the MPC solution of problem (54) with the forward integration of the mean-field dynamics controlled via (57), where the control input $\mathbf{u}_\theta$ is given by the previously trained neural networks $\mathbf{u}_\theta^{FNN}$ and $\mathbf{u}_\theta^{RNN}$. The optimality conditions arising from problem (54) are approximated with a first-order semi-Lagrangian scheme, as done in [2], but neglecting diffusion. Moreover, to mirror the feedback nature of the control resulting from the numerical routine discussed so far, the solver is applied in an MPC fashion, where the optimality conditions are solved iteratively over a reduced time horizon $T = 0.5s$ [53]. Using the same semi-Lagrangian scheme, we simulate the forward dynamics controlled via the kinetic approximation models through (57).

Table 1

Comparison of the mean field optimal control problem solved via the proposed numerical procedure against the MPC routine relying on the first-order semi-Lagrangian scheme discussed in [2]. For the three cases, we display the resulting total cost (top), controlled evolution (middle), and mean field control (bottom), with u_{MPC} retaining the best performance.

**Fig. 5.** Comparison of the time evolution of the running cost for the MFOC (54).

The table reports comparable outcomes in terms of cost functionals, density evolutions, and control profiles across the three approaches. The mean-field control, denoted here by u_{MPC} , retains the best performance. It is worth noting that even u_{MPC} should be regarded as close-to-optimal for an infinite horizon cost. This is because it is computed as the numerical solution of the first-order necessary optimality conditions, implemented in a receding horizon fashion over a sequence of finite-horizon MFOCs. In Fig. 5, we compare the running cost evolution, i.e. the integral w.r.t. time in (54), which is consistent with the results reported in Table 1.

6.2. Test 2: Cucker-Smale model

We proceed by testing the proposed methodology in a consensus control problem for non-linear and non-local dynamics governing the evolution of a N -agent system. In its reduced binary semilinear formulation (51), the model is written for the state $\mathbf{s} = (x, x_*, v, v_*)^\top \in \mathbb{R}^{2d} \times \mathbb{R}^{2d}$, with an interaction kernel given by

$$P(x, x_*) = \frac{1}{1 + \|x - x_*\|^2}. \quad (58)$$

The control variable $\mathbf{u} = (u, u_*) \in \mathbb{R}^{2d}$ is here computed as

$$\bar{\mathbf{u}} = \underset{u(\cdot)}{\operatorname{argmin}} \frac{1}{2} \int_0^{+\infty} \|v - \bar{v}\|^2 + \|v_* - \bar{v}\|^2 + \gamma \|\mathbf{u}\|^2 dt, \quad (59)$$

with $d = 15$, $\gamma = 0.01$, and target velocity $\bar{v} = \frac{v + v_*}{2}$.

As discussed before, the key ingredient of the proposed methodology is a NN approximation of either the control $\mathbf{u}_\theta$ or the (controlled) update $\mathbf{s}'_\theta$ of the reduced binary problem. In both cases, the synthetic data have been generated from $N_s = 10^5$ uniform samples of interacting couples of agents $\{\mathbf{s}_{(i)}\}_{i=1}^{N_s}$ within $\Omega \times \Omega = [-5, 5]^{4d}$, together with their associated (sub)optimal DSDRE feedback control for the discrete-time infinite horizon OCP (44) starting from each $\mathbf{s}_{(i)}$ for the sampled time-step $\Delta t_{(i)} \in [0, 1]$. We highlight that while the dynamics and controls are defined in the entire space, we consider a finite computational domain Ω for sampling.

Furthermore, we notice that we can write the state penalty term as

$$\|v - \bar{v}\|^2 = \|v - Mv\|^2 \quad (60)$$

for $v = (v, v_*)^\top$ and a suitable block matrix $M \in \mathbb{R}^{2d \times 2d}$ defined as

$$M = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & & & \mathbf{0} \\ \frac{1}{2} & \frac{1}{2} & & & \\ & & \ddots & & \\ \mathbf{0} & & & \frac{1}{2} & \frac{1}{2} \\ & & & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \quad (61)$$

It follows that

$$\begin{aligned} \|v - Mv\|^2 &= \langle v - Mv, v - Mv \rangle = v^\top \mathbb{I}_{2d} v + v^\top M^\top M v - 2v^\top M v \\ &= v^\top \left(\mathbb{I}_{2d} + M^\top M - 2M \right) v. \end{aligned} \quad (62)$$

Thus, we can write the cost (59) in quadratic form (31) w.r.t. linear operators

$$Q = \mathbb{I}_{2d} + M^\top M - 2M, \quad R = \frac{\gamma}{2} \cdot \mathbb{I}_{2d}. \quad (63)$$

Once the problem has been written in semi-linear form, we rely on the DSDRE approach (described in algorithm 1) for the generation of a dataset collecting N_s samples of coupled states, time-steps, associated feedback laws, and controlled state updates.

We test the following architectures:

u_θ^{FNN} : $K = 1$ hidden layers with 100 neurons, and $\sigma(\mathbf{x}) = \log(\exp(\mathbf{x}) + 1)$

u_θ^{RNN} : $K = 1$ LSTM cell with 100 neurons, $\sigma(\mathbf{x}) = \max(0, \mathbf{x})$, $\rho(\mathbf{x}) = (1 + \exp(-\mathbf{x}))^{-1}$

$\mathbf{s}'_{v,\theta}^{FNN}$: $K = 3$ hidden layers with 100 neurons per layer, and $\sigma(\mathbf{x}) = \max(0, \mathbf{x})$

$\mathbf{s}'_{v,\theta}^{RNN}$: $K = 1$ LSTM cell with 100 neurons, $\sigma(\mathbf{x}) = \max(0, \mathbf{x})$, $\rho(\mathbf{x}) = (1 + \exp(-\mathbf{x}))^{-1}$

The behaviour of the approximated controlled binary dynamics for a single couple of agents is shown in Fig. 6, where we compare the true DSDRE solution with our NN approximations. As the trajectories evolve, the approximated states deviate from the DSDRE closed-loop. However, this does not significantly affect the MC simulation, as a sampled pair of agents will only interact for the duration of a single time step. The goodness of fit of the trained models measured in a *test set* $\mathcal{T}_t \not\subseteq \mathcal{T} \cup \mathcal{T}_v$ is presented in Table 4.

In Fig. 7, we show the evolution of the density of agents' velocities according to a Monte Carlo simulation of binary controlled dynamics, as described in Algorithm 3. All the approximation models equally succeed in steering the density distribution to concentration profiles. Snapshots of the mean field distributions at specific time intervals show how similar the consensus configurations are across the trained models.

So far, the reliance on approximants for the binary controlled state update has been motivated in terms of efficiency. In Tables 2 and 3, we present CPU times¹ (in seconds) for the MC simulation with the different models, compared with the DSDRE solution $\mathbf{u}$.

¹ The experiments have been executed in MATLAB R2022a installed on a machine with Intel Core i7-10700 processor running at 2.90GHz.

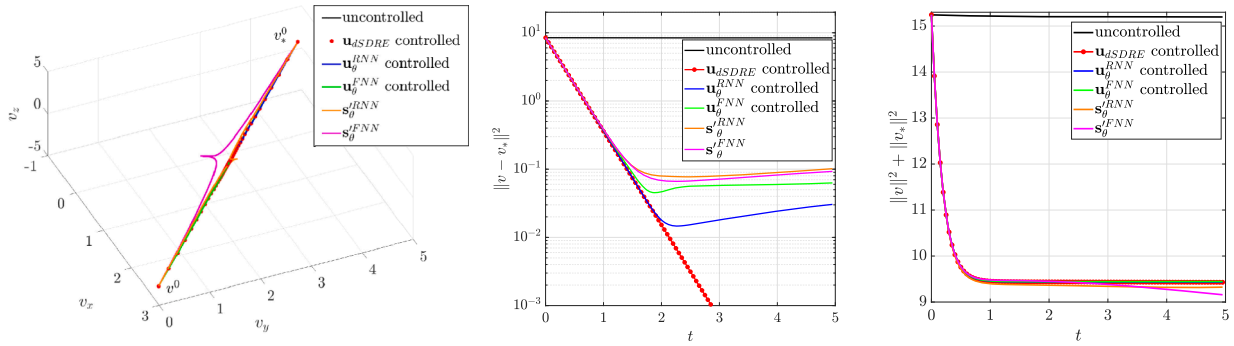


Fig. 6. Evolution of the two interacting agents' in the velocity space restricted to the first 3 dimensions (left); convergence to consensus in logarithmic scale (centre); consensus configurations for the different approximators (right).

Table 2

CPU times (seconds) for coupled agents in $\mathbb{R}^{4d}$, $d = 15$, when considering different number of MC samples. The omitted records exceeded a time threshold $t_{\max} = 24h$.

$d = 15$	$N_s = 10^2$	$N_s = 10^3$	$N_s = 10^4$	$N_s = 10^5$
s_{θ}^{FNN}	0.045515	0.227243	2.097497	24.060892
s_{θ}^{RNN}	0.067627	0.283442	3.210856	34.087541
u_{θ}^{FNN}	0.293578	2.447205	21.754862	2.2594×10^2
u_{θ}^{RNN}	0.340493	2.225319	22.486736	2.2559×10^2
u	2.3866×10^2	2.3738×10^3	-	-

Table 3

CPU times (seconds) when considering 10^4 MC samples of coupled agents in $\mathbb{R}^{4d}$, with varying d . The omitted records exceeded a time threshold $t_{\max} = 24h$.

$N_s = 10^4$	$d = 3$	$d = 7$	$d = 10$	$d = 15$	$d = 30$
s_{θ}^{FNN}	1.048226	1.212633	1.390813	2.142041	2.617840
s_{θ}^{RNN}	2.033726	2.243084	2.493256	3.210856	3.893368
u_{θ}^{FNN}	7.712628	11.006977	15.041731	21.754862	70.172311
u_{θ}^{RNN}	7.734224	11.224325	15.991421	22.486736	70.564372
u	1.1979×10^3	5.2136×10^3	-	-	-

Table 4

Goodness of fit for Tests 1 and 2 in terms of: *coefficient of determination* r^2 , *mean squared error* and *mean percentage error*. $\mathcal{T}_i$ is a collection of sampled states $s^{(i)} \in (\Omega \times \Omega)$ and related target DSDRE values for $i = 1, \dots, 10^5$.

Model	Test 1			Test 2		
	r^2	MSE	MRE %	r^2	MSE	MRE %
s_{θ}^{FNN}	0.99998	0.075252	0.38964	0.9994	0.038274	0.60308
s_{θ}^{RNN}	0.99999	0.0069192	0.3739	0.9998	0.012784	0.20956
u_{θ}^{FNN}	0.99996	0.045596	0.63555	0.9997	7.6882	3.4018
u_{θ}^{RNN}	0.99998	0.018018	0.40136	0.99979	5.5492	2.7483

The improved efficiency achieved through approximation is crucial, as the alignment between the mean field dynamics and their kinetic approximation relies on high-frequency sampling ($\epsilon \ll 1$). The evolution of the agents' distribution is approximated along a sequence of discrete times $t_n = n\Delta t$, $\Delta t = 0.05$, $n = 1, \dots, 100$. The tables address the computational cost associated to sampling a number of controlled binary interactions in the MC simulation, and the dimensionality d of the agents physical space, for a total of $4d$ dimensions. The reliance on NN approximation models allows for a speedup of 2 to 3 orders of magnitude. For $d \geq 7$ and $N_s \geq 10^4$, the computational cost resulting from the use of u_{θ} displays a linear growth, whilst s_{θ}^i performs even better. CPU times for the true DSDRE controlled dynamics exceed 24 h.

6.3. Test 3: Quasi-Morse potential

As a second numerical test, we consider the consensus control problem analysed in [17] for the interacting particles of a second order system of agents in the physical space $\mathbb{R}^3$. The velocities are here governed by both a self-propulsion force, expressed in the $i - th$

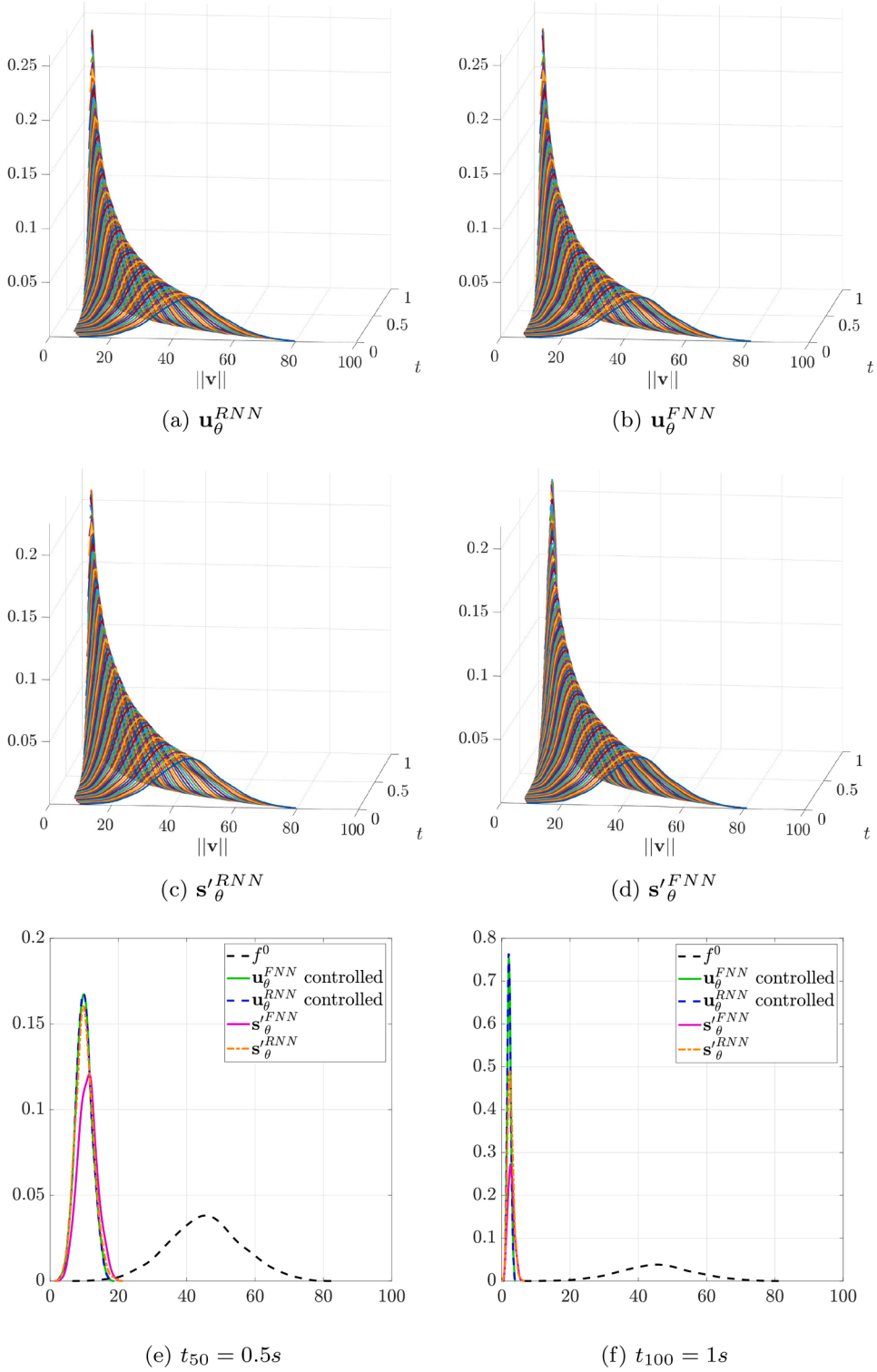


Fig. 7. Density of agents' velocities in norm for Test 1. Comparison of the time evolution of the MC pdf obtained considering different approximation approaches for the controlled binary interactions. In the bottom row, we display a comparison of the different approximated densities at two different discrete times.

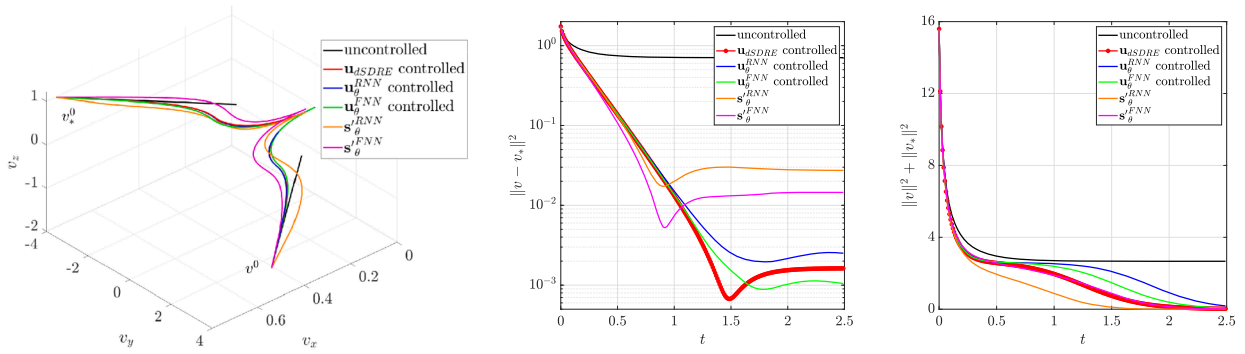


Fig. 8. Evolution (in velocities) of a couple of approximately controlled interacting agents under the different models, versus the uncontrolled dynamics and the dSDRE-controlled ones (left); convergence to consensus in logarithmic scale (centre); consensus configurations for the different approximators (right).

agent by the term $(\alpha - \beta \|v_i(t)\|^2)v_i(t)$, for fixed $\alpha \geq 0$, $\beta > 0$, and an attraction-repulsion force acting through the pairwise interaction potential W . We consider a radial potential of the form

$$W(x) = V(\|x\|) - C V\left(\frac{\|x\|}{l}\right), \quad V(r) := -\exp\left\{-\frac{r^p}{p}\right\} \quad (64)$$

with $C = 0.6$, $p = 1.5$, $l = 0.5$, $\alpha = 2$, $\beta = 1.5$, in a similar configuration as in [17]. Accordingly, for the couple of interacting agents $s = (x, x_*, v, v_*)$ we define

$$P_x(x, x_*) = \|x - x_*\|^{(p-2)} \left(\frac{C}{l^p} e^{-\frac{\|x - x_*\|^p}{p l^p}} - e^{-\frac{\|x - x_*\|^p}{p}} \right) \quad (65)$$

$$P_v(v) = (\alpha - \beta \|v(t)\|^2), \quad (66)$$

with which we write the reduced binary dynamics in semilinear form as

$$A = \begin{bmatrix} 0_{2d} & \mathbb{I}_{2d} \\ A_x & A_v \end{bmatrix}, \quad \begin{bmatrix} A_x \\ A_v \end{bmatrix}_{i,j} = \begin{cases} -P_x(x, x_*) & \text{if } i = j \\ P_x(x, x_*) & \text{if } m \text{ odd, } j = i + 1 \\ P_x(x, x_*) & \text{if } m \text{ even, } j = i - 1 \\ 0 & \text{otherwise} \end{cases} \quad (67)$$

where A_v is the diagonal matrices with diagonal vector the component-wise application of $P_v(\cdot)$ to the vector with d repetitions of v, v_* . Moreover, the cost operators for the consensus goal Q, R hold as in (63).

As discussed for the previous numerical example, we aim at approximating the binary control u , resulting from the DSDRE approach, and the related controlled state update s' . We train the following models: u_{θ}^{FNN} : $K = 3$ hidden layers (100 neurons per layer), $\sigma(x) = \log(1 + e^x)$

u_{θ}^{RNN} : $K = 5$ (100 neurons per layer) with l_1 LSTM cell, $\sigma(x) = \rho(x) = \log(1 + e^x)$

$s'_{v,\theta}^{FNN}$: $K = 4$ (100 neurons per layer), $\sigma(x) = \log(1 + e^x)$

$s'_{v,\theta}^{RNN}$: $K = 3$ (100 neurons per layer), l_1 LSTM cell, $\sigma_1(x) = \rho(x) = \log(1 + e^x)$,

$$\sigma_2(x) = \sigma_3(x) = \text{elu}(x) = \begin{cases} e^x - 1 & \text{if } x \leq 0 \\ x & \text{if } x > 0 \end{cases}$$

The goodness of fit of these models outside $\mathcal{T} \cup \mathcal{T}_v$ is displayed in Table 4, whilst a comparison of the approximately controlled binary system is displayed in Fig. 8, where the dynamics of a couple of agents with states randomly sampled in $[-4, 4]^{12}$ evolve throughout 40 discrete time intervals of length $\Delta t = 0.02s$. Whilst all the NN models lead to a similar final state, those approximating the feedback design perform closer to the reference DSDRE solution. As shown in Fig. 9, the controlled evolution converges to concentrated density profiles, exhibiting slight differences in mean and variance among the various models. This can also be noticed from the Fig. 10, which depicts the density of agents along the first two dimensions, together with the associated vector field of the velocities.

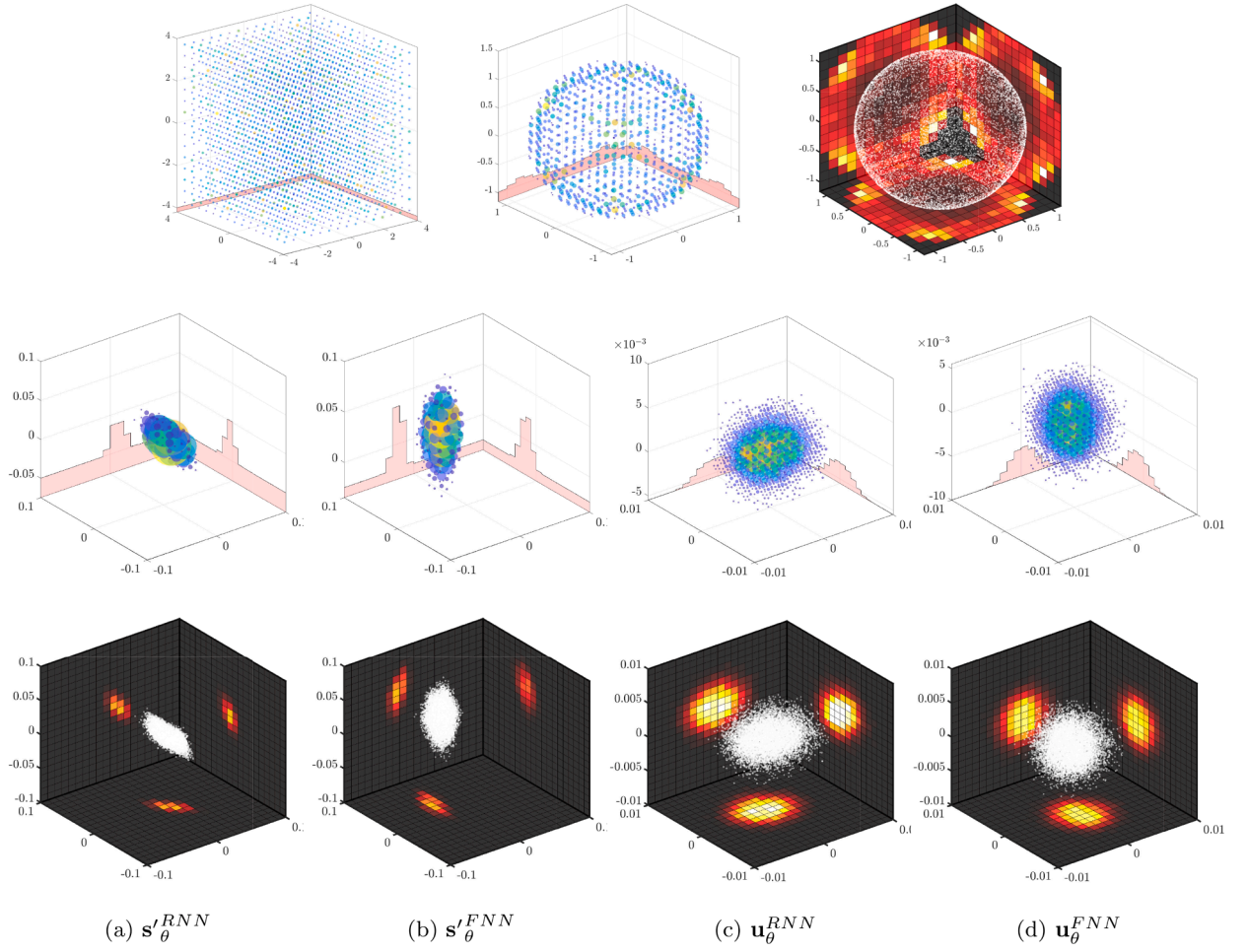


Fig. 9. Density of the system in the velocity space. First row: density of the uncontrolled system at initial and final times $t \in \{0s, 2s\}$. Second row: controlled system configuration at time $T = 2s$, with projection of partial densities along the first and second dimension. Third row: configuration of the controlled system at the final time $T = 2s$, with projection of partial densities along all the three dimensions.

7. Conclusions

We have developed a novel computational method for mitigating the curse of dimensionality arising in the optimal control of large-scale, high-dimensional, agent-based models. The key ingredients of the proposed approach are: the use of a kinetic model to reduce the complexity associated to the particle ensemble to the sampling of two-agent subproblems, and the synthesis of control laws for the two-agent system by means of deep neural networks, supervised learning, and a discrete-time state-dependent Riccati equation approach.

Our numerical experiments validate the effectiveness of the approach in controlling consensus and attraction-repulsion dynamics in high-dimensional interacting particle systems. The use of neural network approximation models for fast feedback synthesis allows for a speedup of 2 to 3 orders of magnitude compared to solving a binary optimal control problem every time two interacting agents are sampled.

The proposed framework can be extended to construct a deep neural network that directly predicts the post-interaction states, or to different control problems, provided that a solver for synthetic data generation is available; moreover, the same approach can be applied to other types of dynamics, as long as they can be simulated or approximated through suitable numerical methods.

Potential future research directions include the generalization of the proposed approach to the H_{∞} robust control framework, which could further enhance the applicability of the method to real-world scenarios where robustness to uncertainties in the agents' interaction forces is crucial. Additionally, exploring the integration of adaptive time-stepping techniques with the neural network approximation models could lead to faster and more accurate simulations of the controlled agent-based system. Finally, investigating the scalability of the proposed approach to even higher-dimensional problems and larger agent populations could enable applications in consensus-based optimization and mean field limits of neural networks.

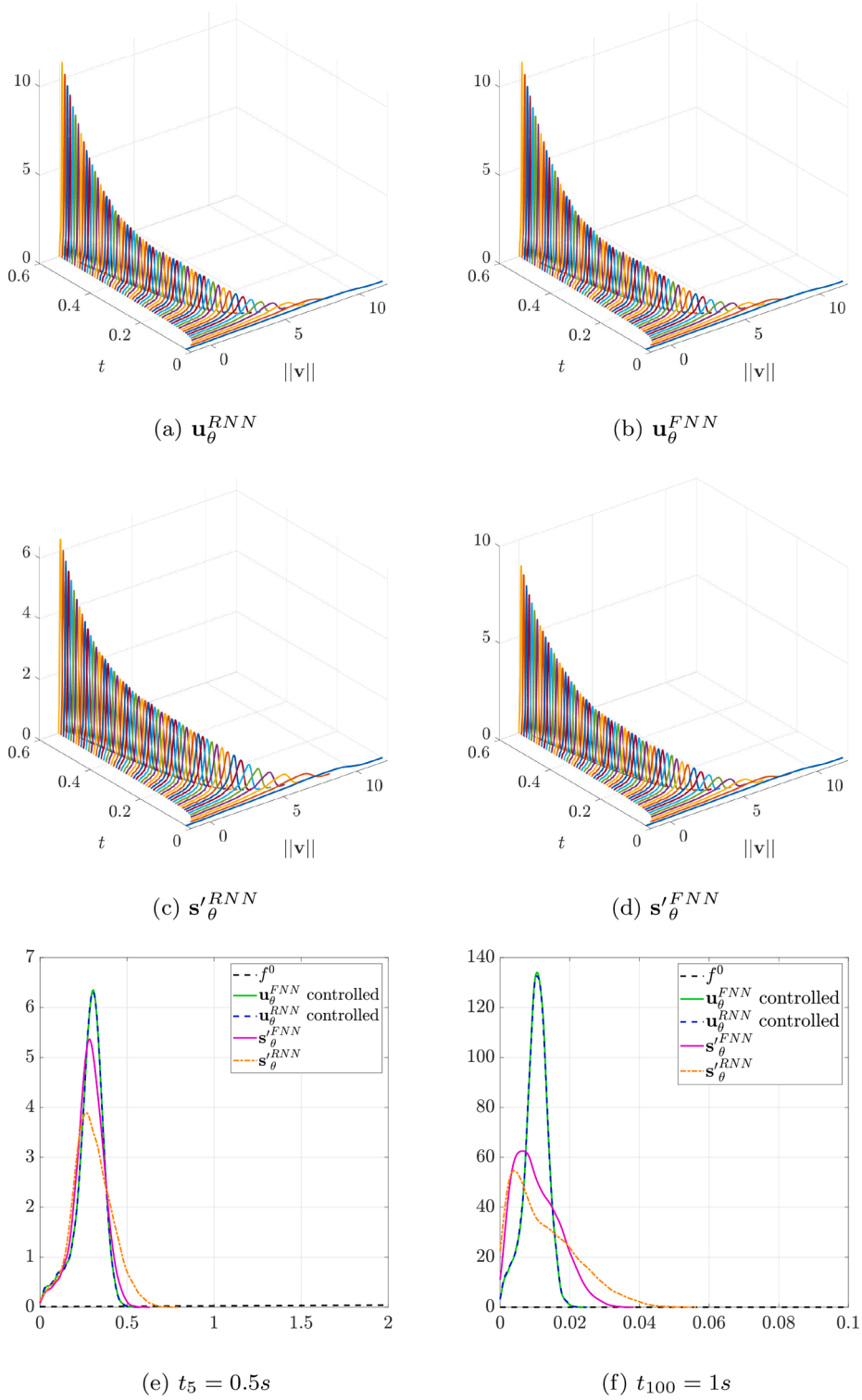


Fig. 10. Density of agents' velocities in norm for test 2. Comparison of the time evolution of the MC pdf obtained considering different approximation approaches for the controlled binary interactions. In the bottom row, we display a comparison of the different approximated densities at two different discrete times.

CRedit authorship contribution statement

Giacomo Albi: Writing – review & editing, Formal analysis; **Sara Bicego:** Writing – original draft, Software; **Dante Kalise:** Writing – review & editing, Supervision, Formal analysis, Project administration.

Data availability

Data will be made available on request.

Declaration of interests

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Giacomo Albi reports financial support was provided by European Union. Dante Kalise reports financial support was provided by Engineering and Physical Sciences Research Council. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

GA is member of Indam GNCS and thanks the support of MUR-PRIN Project 2022 PNRR No. P2022JC95T, “Data-driven discovery and control of multi-scale interacting artificial agent systems”, and of MUR-PRIN Project 2022 No. 2022N9BM3N “Efficient numerical schemes and optimal control methods for time-dependent partial differential equations” financed by the European Union - Next Generation EU. This research was supported by the UK Engineering and Physical Sciences Research Council (EPSRC) grant EP/T024429/1

Appendix A. Proof of Theorem 2.1

A.0.0.1. Step 1. Let us introduce a test function $\varphi \in C_0^2(\mathbb{R}^d \times \mathbb{R}^d)$, is the state space hosting the N_d agents of the system. We consider the following weak formulation of the Boltzmann (8)

$$\begin{aligned} \frac{d}{dt} \langle f, \varphi \rangle + \langle f, v \cdot \nabla_x \varphi \rangle &= \lambda \langle \mathcal{Q}_{\eta, u}(f, f), \varphi \rangle \\ &= \lambda \iint_{\mathbb{R}^{2d} \times \mathbb{R}^{2d}} (\varphi(x, v') - \varphi(x, v)) f(t, x_*, v_*) dx_* dv_* dx dv \end{aligned} \quad (\text{A.1})$$

A.0.0.2. Step 2. By definition of the binary post-interaction dynamics (6), we can explicitly write $v' - v$ as

$$v' - v = \eta P(x, x_*)(v_* - v) + \eta u_{\eta}(x, v, x_*, v_*) =: \eta F_{\eta}(x, v, x_*, v_*) \quad (\text{A.2})$$

where we introduce the function $F_{\eta}(x, v, x_*, v_*)$ for the controlled dynamics binary dynamics. Moreover, we expand $\varphi(v')$ inside the operator (A.1) in Taylor series of $v' - v$ up to the second order, obtaining

$$\varphi(x, v') - \varphi(x, v) = \eta F_{\eta}(x, v, x_*, v_*) \cdot \nabla_v \varphi(x, v) + R^{\varphi}(\eta^2) \quad (\text{A.3})$$

where the term $R^{\varphi}(\eta^2)$ represents the reminder of the Taylor expansion as follows

$$\begin{aligned} R^{\varphi}(\eta^2) &= \sum_{i,j=1}^d ((\partial_v^{(i,j)} \varphi(x, v) - \partial_v^{(i,j)} \varphi(x, \bar{v}))(v' - v)_i (v' - v)_j) \\ &= \eta^2 \sum_{i,j=1}^d ((\partial_v^{(i,j)} \varphi(x, v) - \partial_v^{(i,j)} \varphi(x, \bar{v}))(F_{\eta})_i (F_{\eta})_j) =: \eta^2 \tilde{R}_{\eta}^{\varphi}(x, v, x_*, v_*) \end{aligned} \quad (\text{A.4})$$

where $\bar{v} = (1 - \theta)v' + \theta v$ for some $\theta \in [0, 1]$ and we use the multi-index notation for the second order partial derivatives of φ .

A.0.0.3. Step 3. Embedding splitting (A.3) into the interacting operator for the weak form (A.1), and introducing the quasi invariant scaling (14), i.e. $\lambda = 1/\varepsilon$, $\eta = \varepsilon$ we have

$$\frac{1}{\varepsilon} \langle \mathcal{Q}_{\varepsilon, u}(f, f), \varphi \rangle = \iint_{\mathbb{R}^{2d} \times \mathbb{R}^{2d}} F_{\varepsilon}(x, v, x_*, v_*) \cdot \nabla_v \varphi(x, v) f f_* dx_* dv_* dx dv + \varepsilon \mathcal{R}_{\varepsilon}^{\varphi}, \quad (\text{A.5})$$

with reminder term

$$\mathcal{R}_{\varepsilon}^{\varphi} = \iint_{\mathbb{R}^{2d} \times \mathbb{R}^{2d}} \tilde{R}_{\varepsilon}^{\varphi}(x, v, x_*, v_*) f f_* dx_* dv_* dx dv. \quad (\text{A.6})$$

For $\varepsilon \rightarrow 0$, assuming that the reminder vanishes to zero, and integrating by parts the scaled weak form of the Povzner-Boltzmann model (A.1) we have

$$\left\langle \partial_t f + v \cdot \nabla_x f + \nabla_v \cdot \left[f \int_{\mathbb{R}^{2d}} (P(x, x_*)(v_* - v) + u(x, v, x_*, v_*)) f(t, x_*, v_*) dx_* dv_* \right], \varphi \right\rangle = 0$$

for every $\varphi \in C^2(\mathbb{R}^d \times \mathbb{R}^d)$. Hence, in strong form, we retrieve consistency of the scaled Povzner-Boltzmann model (8) with controlled mean field model (15). Hence, in the next step we show that the reminder is bounded and the previous limit hold true for $\varepsilon \rightarrow 0$.

A.0.0.4. Step 4. Finally we provide estimate on the reminder of the Taylor expansion, to show that such term vanishes for $\varepsilon \rightarrow 0$. We observe that

$$\|\mathcal{R}_\varepsilon^\Phi\| \leq \frac{C}{2} \iint_{\mathbb{R}^{2d} \times \mathbb{R}^{2d}} \|F_\varepsilon(x, v, x_*, v_*)\|^2 f(t, x, v) f(t, x_*, v_*) dx dv dx_* dv_*$$

where C is the bounding constant of $C_0^2(\mathbb{R}^d \times \mathbb{R}^d)$, and since $F_\varepsilon(x, v, x_*, v_*) \in L_{loc}^2$ we can conclude that the quasi-invariant limit holds true. $\square$

Remark 4. The proof provided here can be readapted seamlessly in the case of the full state binary dynamics (17) by assuming that $F_\eta(s, s_*) := G(s, s_*) + Hu(s, s_*) \in L_{loc}^2(\mathbb{R}^{2d} \times \mathbb{R}^{2d})$. More precisely, in Step 1, we consider the test function $\Phi \in C_0^2(\mathbb{R}^{2d})$ and the weak form of (19) as follows

$$\frac{d}{dt} \langle f, \Phi \rangle = \lambda \iint_{\mathbb{R}^{2d} \times \mathbb{R}^{2d}} (\Phi(s') - \Phi(s)) f(t, s_*) f(t, s) ds_* ds, \quad (\text{A.7})$$

where $s' = (x', v')$ is the post-collisional state in (17). In Step 2, by Taylor's expansion around $s = (x, v)$, we retrieve

$$\Phi(s') - \Phi(s) = \eta(G(s, s_*) + Hu(s, s_*)) \cdot \nabla_s \Phi(s) + \eta^2 \tilde{R}_\eta^\Phi, \quad (\text{A.8})$$

with reminder term defined now as

$$\tilde{R}_\eta^\Phi(s, s_*) := \sum_{i,j=1}^{2d} ((\partial_v^{(i,j)} \Phi(s) - \partial_v^{(i,j)} \Phi(\tilde{s}))(F_\eta(s, s_*)_i (F_\eta(s, s_*)_j)), \quad (\text{A.9})$$

where $\tilde{s} = (1 - \theta)s + \theta s'$, $\theta \in [0, 1]$, similarly to (A.4). Then in Step 3, introducing the scaling (14) and using (A.8) in (A.7) we have

$$\frac{d}{dt} \langle f, \Phi \rangle = \iint_{\mathbb{R}^{2d} \times \mathbb{R}^{2d}} ((G(s, s_*) + Hu(s, s_*)) \cdot \nabla_s \Phi(s)) f(t, s_*) f(t, s) ds_* ds + \varepsilon \mathcal{R}_\varepsilon^\Phi, \quad (\text{A.10})$$

where the reminder $\mathcal{R}_\varepsilon^\Phi$ is defined as in (A.6), as integrating $\tilde{R}_\varepsilon^\Phi(s, s_*)$. Thus, integrating by parts, we have that for any test function Φ is

$$\left\langle \partial_t f + \nabla_s \cdot \left(f \int_{\mathbb{R}^{2d}} (G(s, s_*) + Hu(s, s_*)) f(s_*) ds_* \right), \Phi \right\rangle = 0. \quad (\text{A.11})$$

Finally, the reminder vanishes in the limit $\varepsilon \rightarrow 0$ following the same argument of Step 4.

Appendix B. Asymptotic constrained symmetric Nanbu algorithm

Algorithm 4: .

$\{(x_i^0, v_i^0)\}_{i=1}^{N_s} \sim f^0$ i.i.d.	N_s samples from the initial distribution
for $h = 0, \dots, N_T - 1$, $t_n = n \cdot \Delta t$ do	
select $\{(i_k, j_k)\}_{k=1}^{N_s/2}$ random pairs of agents without repetitions	
for $k = 1, \dots, N_s/2$ do	
Compute $(v_{i_k}^{n+1}, v_{j_k}^{n+1})$ according to (29)	interaction
for $i = 1, \dots, N_s$ do	
Compute x_i^{n+1} according to (25)	transport

For the generalized binary dynamics (17) an analogous stochastic simulation technique can be designed, where in this case interaction and transport are updated simultaneously considering the particle states at time t_n as follows $s_i^n = (x_i^n, v_i^n)^\top$, $s_j^n := (x_j^n, v_j^n)^\top$.

References

- [1] G. Albi, S. Bicego, D. Kalise, Gradient-augmented supervised learning of optimal feedback laws using state-dependent Riccati equations, *IEEE Cont. Syst. Lett.* 6 (2022) 836–841. <https://doi.org/10.1109/LCSYS.2021.3086697>.
- [2] G. Albi, Y.P. Choi, M. Fornasier, D. Kalise, Mean field control hierarchy, *Appl. Math. Optim.* 76 (2017) 93–135. <https://doi.org/10.1007/s00245-017-9429-x>.
- [3] G. Albi, M. Herty, D. Kalise, C. Segala, Moment-driven predictive control of mean-field collective dynamics, *SIAM J. Control Optim.* 60 (2022) 814–841. <https://doi.org/10.1137/21M1391559>.
- [4] G. Albi, M. Herty, L. Pareschi, Kinetic description of optimal control problems and applications to opinion consensus, *Commun. Math. Sci.* 13 (2015) 1407–1429. <https://doi.org/10.4310/CMS.2015.v13.n6.a3>.
- [5] G. Albi, L. Pareschi, Binary interaction algorithms for the simulation of flocking and swarming dynamics, *Multiscale Model. & Simul.* 11 (2013) 1–29. <https://doi.org/10.1137/120868748>.
- [6] G. Albi, L. Pareschi, M. Zanella, Boltzmann-type control of opinion consensus through leaders, *Philos. Trans. R. Soc A: Math., Phys. Eng. Sci.* 372 (2014) 20140138. <https://doi.org/10.1098/rsta.2014.0138>.
- [7] G. Aletti, G. Naldi, G. Toscani, First order continuous models of opinion formation, *SIAM J. Appl. Math.* 67 (2007) 837–853. <https://doi.org/10.1137/060658679>.
- [8] A. Amirkhani, A.H. Barshooi, Consensus in multi-agent systems: a review, *Artif. Intell. Rev.* 55 (2022). <https://doi.org/10.1007/s10462-021-10097-x>.
- [9] G.C. Andreas Wernli, Suboptimal control for the nonlinear quadratic regulator problem, *Automatica* 11 (1975) 75–84. [https://doi.org/10.1016/0005-1098\(75\)90010-2](https://doi.org/10.1016/0005-1098(75)90010-2).
- [10] B. Azmi, D. Kalise, K. Kunisch, Optimal feedback law recovery by gradient-augmented sparse polynomial regression, *J. Mach. Learn. Res.* 22 (2021), Paper No. 48, 32. <http://jmlr.org/papers/v22/20-755.html>.
- [11] D.P. Bertsekas, *Dynamic programming and optimal control. vol. i*, Athena Scientific, Belmont, MA, 4th edition, 2017.
- [12] G. Bird, *Molecular gas dynamics and the direct simulation of gas flows*, Clarendon Press, 1994. v. 1
- [13] A.V. Bobylev, K. Nanbu, Theory of collision algorithms for gases and plasmas based on the Boltzmann equation and the Landau-Fokker-Planck equation, *Phys. Rev. E* 61 (2000) 4576. <https://doi.org/10.1103/PhysRevE.61.4576>.
- [14] M. Bongini, M. Fornasier, M. Hansen, M. Maggioni, Inferring interaction rules from observations of evolutive systems I: the variational approach, *Math. Models Methods Appl. Sci.* 27 (2017) 909–951. <https://doi.org/10.1142/S0218202517500208>.
- [15] G. Borghi, L. Pareschi, Wasserstein convergence rates for stochastic particle approximation of Boltzmann models, (2025). [arXiv preprint arXiv:2504.10091](https://arxiv.org/abs/2504.10091)
- [16] J.A. Carrillo, M. Fornasier, J. Rosado, G. Toscani, Asymptotic flocking dynamics for the kinetic Cucker-Smale model, *SIAM J. Math. Anal.* 42 (2010) 218–236. <https://doi.org/10.1137/090757290>.
- [17] J.A. Carrillo, D. Kalise, F. Rossi, E. Trélat, Controlling swarms toward flocks and mills, *SIAM J. Control Optim.* 60 (2022) 1863–1891. <https://doi.org/10.1137/21M1404314>.
- [18] I. Chang, J. Bentsman, Constrained discrete-time state-dependent riccati equation technique : a model predictive control approach, *Proceedings of the IEEE Conference on Decision and Control* (2013) 5125–5130. <https://doi.org/10.1109/CDC.2013.6760694>.
- [19] F. Chen, W. Ren, On the control of multi-agent systems: a survey, *SYS 6* (2019) 339–499. Publisher: Now Publishers, Inc. <https://doi.org/10.1561/26000000019>.
- [20] S. Cordier, L. Pareschi, G. Toscani, On a kinetic model for a simple market economy, *J. Stat. Phys.* 120 (2005) 253–277. <https://doi.org/10.1007/s10955-005-5456-0>.
- [21] J. Darbon, G.P. Langlois, T. Meng, Overcoming the curse of dimensionality for some Hamilton-Jacobi partial differential equations via neural network architectures, *Res. Math. Sci.* 7 (2020), Paper No. 20 50. <https://doi.org/10.1007/s40687-020-00215-6>.
- [22] K.V.R. Devi, B.S. Smitha, S. Lakhanpal, R. Kalra, V.A. Sethi, S.K. Thajil, A review: swarm robotics: cooperative control in multi-agent systems, *E3S Web Conf.* 505 (2024) 03013. <https://doi.org/10.1051/e3sconf/202450503013>.
- [23] R. DeVore, B. Hanin, G. Petrova, Neural network approximation, *Acta Numerica* 30 (2021) 327–444. <https://doi.org/10.1017/S09662492921000052>.
- [24] G. Dimarco, Q. Li, L. Pareschi, B. Yan, Numerical methods for plasma physics in collisional regimes, *J. Plasma Phys.* 81 (2015) 305810106. <https://doi.org/10.1017/S0022377814000762>.
- [25] S. Dolgov, D. Kalise, K.K. Kunisch, Tensor decomposition methods for high-dimensional Hamilton-Jacobi-Bellman equations, *SIAM J. Sci. Comput.* 43 (2021) A1625–A1650. <https://doi.org/10.1137/19M1305136>.
- [26] S. Dolgov, D. Kalise, L. Saluzzi, Data-driven tensor train gradient cross approximation for Hamilton-Jacobi-Bellman equations, *SIAM J. Sci. Comput.* 45 (2023) A2153–A2184. <https://doi.org/10.1137/22M1498401>.
- [27] A.S. Dutka, A.W. Ordys, M.J. Grimble, Optimized discrete-time state dependent Riccati equation regulator, *Proceedings of the 2005, American Control Conference* 4 (2005) 2293–2298. <https://doi.org/10.1109/ACC.2005.1470311>.
- [28] M. Fornasier, F. Solombrino, Mean-field optimal control, *ESAIM Control Optim. Calc. Var.* 20 (2014) 1123–1152. <https://doi.org/10.1051/cocv/2014009>.
- [29] B. Geng, S. Brahma, T. Wimalajeewa, P.K. Varshney, M. Rangaswamy, Prospect theoretic utility based human decision making in multi-agent systems, *IEEE Trans. Signal Process.* 68 (2020) 1091–1104. <https://doi.org/10.1109/TSP.2020.2970339>.
- [30] A. Gooran Orimi, S. Effati, M.H. Farahi, Approximate solution of the Hamilton-Jacobi-Bellman equation, *J. Math. Model.* 10 (2022) 71–91. <https://doi.org/10.22124/jmm.2021.18386.1579>.
- [31] M. Herty, L. Pareschi, S. Steffensen, Mean-field control and Riccati equations, *Netw. Heterogen. Media* 10 (2015) 699–715. <https://doi.org/10.3934/nhm.2015.10.699>.
- [32] H.T. Banks, B.M. Lewis, H.T. Tran, Nonlinear feedback controllers and compensators: a state-dependent Riccati equation approach, *Comput. Optim. Appl.* 37 (2007) 177–218. <https://doi.org/10.1007/s10589-007-9015-2>.
- [33] W. Kang, Q. Gong, T. Nakamura-Zimmerer, F. Fahroo, Algorithms of data generation for deep learning and feedback design: a survey, *Phys. D* 425 (2021), Paper No. 132955 10. <https://doi.org/10.1016/j.physd.2021.132955>.
- [34] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, *CoRR abs/1412.6980* (2015). <https://doi.org/10.48550/arXiv.1412.6980>.
- [35] K. Kunisch, D. Walter, Semiglobal optimal feedback stabilization of autonomous systems via deep neural network approximation, *ESAIM Control Optim. Calc. Var.* 27 (2021), Paper No. 16 59. <https://doi.org/10.1051/cocv/2021009>.
- [36] I. Lagaris, A. Likas, D. Fotiadis, Artificial neural networks for solving ordinary and partial differential equations, *IEEE Trans. Neural Networks* 9 (1998) 987–1000. <https://doi.org/10.1109/72.712178>.
- [37] F. Lu, M. Maggioni, S. Tang, Learning interaction kernels in heterogeneous systems of agents from multiple trajectories, *Found Comput. Math.* 22 (2022). <https://doi.org/10.1007/s10208-021-09521-z>.
- [38] S. McNamara, W.R. Young, Kinetics of a one-dimensional granular medium in the quasielastic limit, *Phys. Fluids A* 5 (1993) 34–45. <https://doi.org/10.1063/1.858896>.
- [39] Y. Meng, R. Zhou, A. Mukherjee, M. Fitzsimmons, C. Song, J. Liu, Physics-informed neural network policy iteration: algorithms, convergence, and verification, *PMLR* (2024). 235:35378–35403.
- [40] K.M. Na, C.H. Lee, Physics-informed deep learning approach to solve optimal control problem, *AIAA SCITECH 2024 Forum*. <https://doi.org/10.2514/6.2024-0945>.
- [41] D.S. Naidu, R.C. Dorf, *Optimal control systems*, CRC Press, Inc., USA, USA, 2003. <https://doi.org/10.1201/9781315214429>.
- [42] S.R. Nekoo, Tutorial and review on the state-dependent riccati equation, *J. Appl. Nonlinear Dyn.* 8 (2019) 109–166. <https://doi.org/10.5890/JAND.2019.06.001>.
- [43] D. Onken, L. Nurbekyan, X. Li, S.W. Fung, S. Osher, L. Ruthotto, A neural network approach for high-dimensional optimal control applied to multiagent path finding, *IEEE Trans. Control Syst. Technol.* 31 (2023) 235–251.
- [44] M. Oster, L. Sallandt, R. Schneider, Approximating optimal feedback controllers of finite horizon control problems using hierarchical tensor formats, *SIAM J. Sci. Comput.* 44 (2022) B746–B770. <https://doi.org/10.1137/21M1412190>.

- [45] M. Oster, L. Saluzzi, T. Wenzel, A comparison study of supervised learning techniques for the approximation of high dimensional functions and feedback control, *Dyn. Games Appl.* 15 (2025) 454–480. <https://doi.org/10.1007/s13235-024-00610-6>.
- [46] Papadopoulou, Furtbauer, O'Bryan, Garnier, Georgopoulou, Bracken, Christensen, King, Dynamics of collective motion across time and species, *Philos. Trans. R. Soc. B: Biol. Sci.* 378 (2023). <https://doi.org/10.1098/rstb.2022.0068>.
- [47] L. Pareschi, G. Russo, An introduction to Monte Carlo method for the Boltzmann equation, *ESAIM: Proceedings* 10 (2001) 35–75. <https://doi.org/10.1051/PROC:2001004>.
- [48] L. Pareschi, G. Toscani, *Interacting multiagent systems. Kinetic equations and Monte Carlo methods*, Oxford: Oxford University Press, 2014.
- [49] L. Pareschi, G. Toscani, Wealth distribution and collective knowledge: a Boltzmann approach, *Philos. Trans. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci.* 372 (2014) 20130396, 15. <https://doi.org/10.1098/rsta.2013.0396>.
- [50] L. Pareschi, T. Trimborn, M. Zanella, Mean-field control variate methods for kinetic equations with uncertainties and applications to socioeconomic sciences, *Int. J. Uncertain. Quantif.* 12 (2022) 61–84. <https://doi.org/10.1615/Int.J.UncertaintyQuantification.2021037960>.
- [51] B. Piccoli, F. Rossi, E. Trélat, Control to flocking of the kinetic cucker-smale model, *SIAM J. Math. Anal.* 47 (2015) 4685–4719. <https://doi.org/10.1137/140996501>.
- [52] A.J. Povzner, On the boltzmann equation in the kinetic theory of gases, *Mat. Sb. (N.S.)* 58(100) (1962) 65–86.
- [53] S.V. Raković, W.S. Levine (Eds.), *Handbook of Model Predictive Control*, Birkhäuser/Springer, Cham, 2019. Control Engineering, <https://doi.org/10.1007/978-3-319-77489-3>.
- [54] F. Rossi, S. Bandyopadhyay, M. Wolf, M. Pavone, Review of multi-agent algorithms for collective behavior: a structural taxonomy, *IFAC-PapersOnLine* 51 (2018) 112–117. IFAC Workshop on Networked & Autonomous Air & Space Systems NAASS 2018. <https://doi.org/10.1016/j.ifacol.2018.07.097>.
- [55] L. Ruthotto, S.J. Osher, W. Li, L. Nurbekyan, 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 (2020) 9183–9193. <https://doi.org/10.1073/pnas.1922204117>.
- [56] J. Sirignano, K. Spiliopoulos, DGM: a deep learning algorithm for solving partial differential equations, *J. Comput. Phys.* 375 (2018) 1339–1364. <https://doi.org/10.1016/j.jcp.2018.08.029>.
- [57] G. Toscani, Kinetic models of opinion formation, *Comm. Math. Sci.* 4 (2006) 481–496. <https://doi.org/10.4310/CMS.2006.v4.n3.a1>.
- [58] A. Tosin, M. Zanella, Boltzmann-type models with uncertain binary interactions, *Commun. Math. Sci.* 16 (2018) 963–985. <https://doi.org/10.4310/CMS.2018.v16.n4.a3>.
- [59] A. Tosin, M. Zanella, ©2021. Boltzmann-type description with cutoff of follow-the-leader traffic models, in: *Trails in kinetic theory—foundational aspects and numerical methods*, volume 25 of SEMA SIMAI Springer Ser., Springer Cham (2021), pp. 227–251. https://doi.org/10.1007/978-3-030-67104-4_8.
- [60] C. Villani, On a new class of weak solutions to the spatially homogeneous Boltzmann and Landau equations, *Arch. Ration. Mech. Anal.* 143 (1998) 273–307. <https://doi.org/10.1007/s002050050106>.
- [61] T. Xiao, M. Frank, Using neural networks to accelerate the solution of the Boltzmann equation, *J. Comput. Phys.* 443 (2021). Paper No. 110521 22. <https://doi.org/10.1016/j.jcp.2021.110521>.
- [62] Y. Zhao, J. Han, Offline supervised learning v.s. online direct policy optimization: a comparative study and a unified training paradigm for neural network-based optimal feedback control, *Physica D: Nonlinear Phenom.* 462 (2024) 134130. <https://doi.org/10.1016/j.physd.2024.134130>.
- [63] T. Çimen, State-dependent riccati equation (SDRE) control: a survey, *IFAC Proceedings Volumes* 41 (2008) 3761–3775. 17th IFAC World Congress. <https://doi.org/10.3182/20080706-5-KR-1001.00635>.
- [64] K. Sznajd-Weron, J. Sznajd, Opinion evolution in closed community, *Intern. J. Modern Phys. C* 11 (6) (2000) 1157–1165. World Scientific Pub Co Pte Ltd. 1793–6586.
- [65] I. Chang, J. Bentsman, Constrained discrete-time state-dependent Riccati equation technique: a model predictive control approach, in: *Proceedings of the 52nd IEEE Conference on Decision and Control*, 2013, pp. 5125–5130. Florence, Italy. <https://doi.org/10.1109/CDC.2013.6760694>.