ODEs Learn to Walk: ODE-Net based Data-Driven Modeling for Crowd Dynamics

Chen Cheng
Shanghai Jiao Tong University
Shanghai, China
chengchen0301@sjtu.edu.cn

Jinglai Li
University of Birmingham
Edgbaston, United Kingdom
J.Li.10@bham.ac.uk

ABSTRACT

Predicting the behaviors of pedestrian crowds is of critical importance for a variety of real-world problems. Data driven modeling, which aims to learn the mathematical models from observed data, is a promising tool to construct models that can make accurate predictions of such systems. In this work, we present a data-driven modeling approach based on the ODE-Net framework, for constructing continuous-time models of crowd dynamics. We discuss some challenging issues in applying the ODE-Net method to such problems, which are primarily associated with the dimensionality of the underlying crowd system, and we propose to address these issues by incorporating the social-force concept in the ODE-Net framework. Finally application examples are provided to demonstrate the performance of the proposed method.

KEYWORDS

crowd dynamics; data-driven modeling; ODE-Net; social force

ACM Reference Format:
Chen Cheng and Jinglai Li. 2024. ODEs Learn to Walk: ODE-Net based Data-Driven Modeling for Crowd Dynamics. In *Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), Auckland, New Zealand, May 6 – 10, 2024*, IFAAMAS, 9 pages.

1 INTRODUCTION

Collective motion of pedestrians is a highly common phenomenon in urban life, and understanding the dynamics of pedestrian crowds is essential for a large variety of applications, ranging from safety management [9, 11] to robot navigation [18, 19]. Modeling the behaviors of pedestrian crowds has attracted considerable attention in multiple disciplines such as physics, social science and artificial intelligence, and various models have been proposed in the past decades. Due to the complexity of the crowd dynamics, driving the mathematical models that can accurately predict the crowd behaviors is an extremely challenging task. To this end a particularly promising remedy is to develop mathematical models with the assistance of related data, an approach often referred to as data-driven modeling [16].

Within the context of crowd dynamics modeling, we here discuss two main strategies behind the data driven methods. The first strategy assumes that the crowd dynamics follows a specific mathematical model that is usually derived based on physics, but all or some of the model parameters are not available; one then estimates these parameters by fitting the observation data into the model. Examples of such methods include [12, 13, 17, 20], among some others. While this type of methods are conceptually straightforward and relatively easy to implement, their performance is ultimately limited by the mathematical models adopted. The second strategy offers more flexibility: namely it does not impose a specific mathematical model; rather, it learns the model (often represented by an artificial neural network) directly from the data with machine learning techniques. While their implementation is usually more complicated, the machine-learning based methods are much less restrictive than the first kind and can potentially obtain very accurate model, provided that high-quality data are available.

In the past a few years, various efforts have been made to the machine learning based data driven modeling, e.g., [1–3, 21]. To the best of our knowledge, most of these existing methods are designed to learn crowd dynamics models that are discrete in time, largely because the discrete-time models can be naturally formulated with a deep neural network such as the recurrent neural network (RNN). On the other hand, there is strong desire to develop continuous-time models, as they can be used to predict the crowd behaviors at any time of interest. The ODE-Net method, first proposed in [6], has gained attention as a tool to learn continuous-time models of physical systems [5, 10, 15, 22]. Simply speaking, ODE-Net formulates the system of interest as an ordinary differential equation system, which is represented by a deep neural network, and learned from the data. The ODE-Net method, however, can not be directly applied to the crowd dynamics, and we summarise three main challenges of it, all associated with the crowd size (or equivalently the dimensionality of the system): first and foremost, due to the high training cost, ODE-Net generally has difficulty dealing with systems of high dimensions, rendering it especially unsuited for large-size crowds; secondly, in reality the size of a crowd may vary in time, with pedestrians entering or leaving the scene of interest, and such a system can not be easily modeled by ODE-Net; finally, the model obtained by ODE-Net cannot be used to predict crowds whose size is different from the training system, which makes its application very limited. In this work we propose an ODE-Net based method to learn the crowd dynamics models from data, where the aforementioned issues are addressed by incorporating underlying physical knowledge of the dynamics into the ODE-Net model. In particular, we adopt the concept that the crowd is a physical system driven by social and psychological forces as is in the so-called social force model (SFM) [8], and then learn those force functions from data. The resulting social force

Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), N. Alechina, V. Dignum, M. Dastani, J.S. Sichman (eds.), May 6 – 10, 2024, Auckland, New Zealand. © 2024 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

based method allows one to learn the models from data for large-scale and variable-size crowds, and also use the learned models to predict the behaviors of crowds of any sizes.

The rest of the paper is organized as follows. In Section 2 we present the social force based ODE-Net method, and in Section 3 we demonstrate the performance of the proposed method by applying it to data generated from two commonly used crowd dynamics models. Finally Section 4 offers some conclusions and discussions.

2 METHODOLOGIES

2.1 ODE-Net for crowd dynamics

We start by introducing the ODE-Net from a deep neural network perspective. Traditional deep neural networks, such as residual networks, build complicated transformations by composing a sequence of transformations to a hidden state:

$$\frac{z_{t+\delta_t} - z_t}{\delta_t} = h_t\left(z_t\right), \quad \delta_t = 1, \tag{1}$$

where $h_t\left(z_t\right)$ is a function parameterized by a neural network. These iterative updates can be interpreted as an Euler discretization of a continuous transformation. In contrast to traditional deep neural networks where $\delta_t = 1$ is fixed, ODE-Net [6] introduced a continuous version in which $\delta_t \to 0$. As a result, Eq. (1) becomes

$$\frac{\mathrm{d}z(t)}{\mathrm{d}t} = h(z(t), t). \tag{2}$$

In this continuous framework, training the networks becomes to learn the function $h(z, t)$ and next we will discuss how to learn this function.

First we assume that the function $h(z, t)$ is represented by a neural network $h_\theta(z, t)$ parameterized by θ, and we have observed data at t_0 and t_1, denoted as $\hat{z}(t_0)$ and $\hat{z}(t_1)$ respectively. Starting from the input layer $\hat{z}(t_0)$, the output layer $z(t_1)$ can be defined by the solution to this ODE initial value problem at some time t_1:

$$z(t_1) = \hat{z}(t_0) + \int_{t_0}^{t_1} h_\theta(z(t), t)dt, \tag{3}$$

and the time from t_0 to t_1 is referred to as the integration time of the data point. Eq. (3) can be computed using an off-the-shelf differential equation solver and we write it as,

$$z(t_1) = \text{ODESolve}\left(\hat{z}(t_0), h_\theta, t_0, t_1\right). \tag{4}$$

The network parameters θ are computed by iteratively minimizing a prescribed loss function $L(\hat{z}(t_1), z(t_1))$, which measures the difference between the observed data $\hat{z}(t_1)$ and the model prediction $z(t_1)$. An interesting feature of this method is that the gradient of the loss function with respect to θ can be computed using the adjoint sensitivity method, which is more memory efficient than directly backpropagating through the integrator [6].

As has been discussed earlier, ODE-Net allows us to construct a continuous-time model for the crowd dynamics. Namely, let $z(t)$ represents the state of the crowd at time t and as a result Eq. (2) becomes the governing equation of the crowd dynamics; suppose that we have observations of the crowd flow $\hat{z}(t)$, and we can use the training process described above to learn the function $h(z, t)$ (or more precisely its neural network representation $h_\theta(z, t)$).

Though the application of ODE-Net to crowd dynamics is conceptually straightforward, the implementation is highly challenging. When applied to crowd dynamics, z represents the state of motion of the entire crowd that may consist of a large number of particles (i.e., pedestrians, and throughout the paper we use these two terms interchangeably), and it follows that z can be of very high dimensions since the dimensionality of z is proportional to the size of the crowd. In this case, learning a high-dimensional function $h(z, t)$ can be prohibitively difficult: it may require a massive amount of training data which may not be available in practice, and the computational cost for training such a complex model can be exceedingly high. In addition, as one can see, in the formulation described above, the dimensionality of z needs to be fixed, which often does not meet the reality, as in most situations people may enter or leave the scene of interest and the dimensionality of z varies over time. More importantly, as the dimensionality of z is fixed, once the model is learned from the data, it can only be used to predict systems of the same number of particles, a serious limitation of the usefulness of the method. To address these issues, we propose to address the dimensionality issue by incorporating the social force (SF) concept into the ODE-Net method, which is detailed in Section 2.2.

2.2 Social-force based ODE-Net

Suppose that we consider a crowd of N particles and we can write the state variable $\boldsymbol{z} = (z_1, ..., z_N)^T$ where z_n represents the state of motion of particle n for each $n = 1...N$. In particular we have $z_n = (x_n, v_n)$ where x_n and v_n are respectively the position and the velocity of particle n. We also introduce the notations $\boldsymbol{x} = (x_1, ..., x_N)^T$ and $\boldsymbol{v} = (v_1, ..., v_N)^T$. Now according to the Newton's second law, model (2) can be re-written as

$$\begin{bmatrix} \dot{\boldsymbol{x}} \\ \dot{\boldsymbol{v}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{v} \\ \mathcal{M}^{-1}\boldsymbol{f} \end{bmatrix}, \tag{5}$$

where

$$\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{v}) = \begin{pmatrix} f_1(\boldsymbol{x}, \boldsymbol{v}) \\ ... \\ f_N(\boldsymbol{x}, \boldsymbol{v}) \end{pmatrix}$$

with $f_n(\boldsymbol{x}, \boldsymbol{v})$ being the force applied to particle n and the matrix $\mathcal{M} = \text{diag}[m_1, ..., m_N]$ with m_n being the "mass" of particle n. With formulation (5), the original ODE-Net problem is turned into learning the force function $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{v})$ and estimating the mass matrix $\mathcal{M}$, where one can see that learning function $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{v})$ is by far the more challenging task here.

It is important to note that in such problems $\boldsymbol{f}$ and $\mathcal{M}$ should not be understood as the usual physical forces and masses respectively. Rather, following the assumption of the social force model [8], $\boldsymbol{f}$ represents the socio-psychological forces driven by personal motivations and environmental constraints, and the mass matrix $\mathcal{M}$ characterizes how easy or difficult to change the velocity of each pedestrian. At this point the force field $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{v})$ is still a high dimensional function for large crowd size N, and further simplification is needed to make the learning problem feasible.

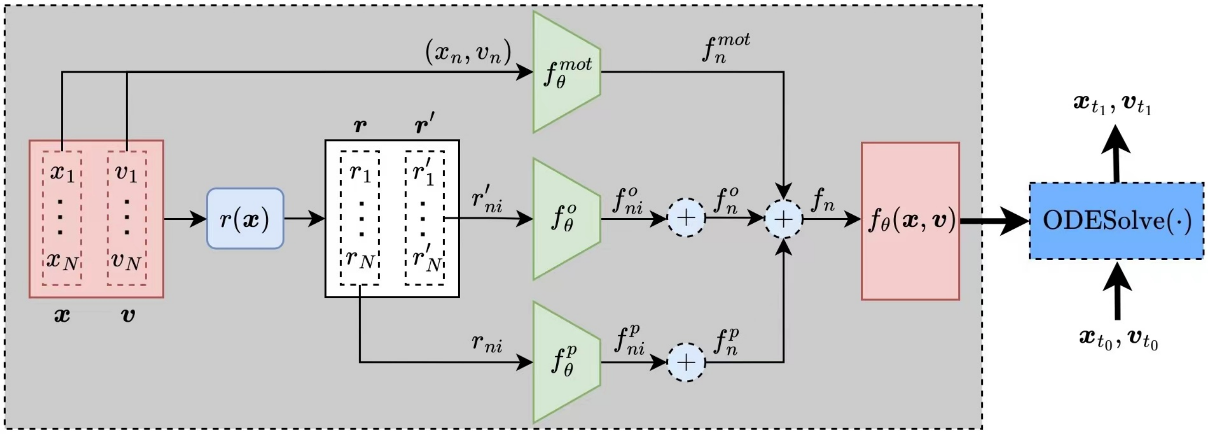

Figure 1: Graphic representation of the SF-based ODE-Net. The shaded box shows that how the social force is computed. The ODE system defined by the force via Eq. (10)**, enters the ODE solver together with the initial states x_{t_0}, v_{t_0}, yielding predictions of the states x_{t_1}, v_{t_1}.**

We now introduce further assumptions to simplify the force function. First we assume that the total force applied to each particle/pedestrian consists of two parts:

$$f_n = f_n^{mot} + f_n^{int}, \tag{6}$$

where f_n^{mot} is the force generated by personal motivation to reach certain desired state of motion, and f_n^{int} is the force caused by the interactions with other particles and the environments (e.g., obstacles). The total interaction force is further written as,

$$f_n^{int} = \sum_{j(\neq n)=1}^{N} f_{nj}^p + \sum_{w=1}^{W} f_{nw}^o, \tag{7}$$

where f_{nj}^p is the interaction force between pedestrians n and j and f_{nw}^o between pedestrian n and the w-th obstacles (assuming there are W obstacles in total). We now need to deal with both the motivation and the interaction forces. We first assume that the personal motivation force depends on the particle's state of motion:

$$f_n^{mot} = f_\theta^{mot}(x_n, v_n, d), \tag{8}$$

where d represents some environmental factors that also affect the motivation force, and $f_\theta^{mot}(\cdot)$ is an artificial neural network parametrized by θ. Next we consider the interaction force f^{int}. To this end, it is common to assume that pedestrians psychologically tend to keep a distance between each other and avoid hitting obstacles. As such, the two interacting forces can be written as,

$$f_{nj}^p = f_\theta^p(r_{nj}, u_{nj}), \tag{9a}$$

$$f_{nw}^o = f_\theta^o(r_{nw}, u_{iw}), \tag{9b}$$

where $r_{nj} = x_j - x_n$ is the relative location of particle j to particle n, and $u_{nj} = v_j - v_n$ is the relative velocity of particle j to particle n, and r_{nw} and u_{nw} are defined in the same way for the obstacles. Note here that v_w is usually 0 in practice.

Under these assumptions, the total force function is completely determined by f_θ^{mot}, f_θ^p and f_θ^o. *Importantly the dimensionality of these three functions is independent on the crowd size N, and therefore the learning problem for constructing the ODE-Net model is of fixed dimensionality regardless of how large the crowd is.* Relatedly the resulting model can be applied to a crowd of any size once the functions are learned.

2.3 Implementation of SF-based ODE-Net

In this section we discuss how to implement the SF based ODE-Net method for crowd dynamics. Simply speaking, one just inserts the social force functions f_θ^{mot}, f_θ^p and f_θ^o into the ODE model (5) via Eqs. (6)–(9), and then trains the resulting ODE-Net with the algorithms described in Section 2.1. In what follows we provide further implementation details.

In our numerical implementation in this work we make the following assumptions:

a) the masses of all particles are the same; this is of course a simplification, but it is important for the application of the learned model, as we typically do not have the knowledge of the "mass" of each pedestrian when applying the learned ODE-Net model;
b) a particle is only interacted with its K nearest neighbors, a measure imposed to reduce the computational cost;
c) the motivation force of a particle only depends on the velocity and position of it and no environmental factors are explicitly included;
d) the interaction force between two particles (or a particle and an obstacle) only depends on the relative position of the objectives.

It is important to note that any of these assumptions can be removed or modified without affecting the implementation procedure described here – for example the interaction force can also depend on the relative velocity between particles.

The governing model (5) can be represented by a collection of the models for each particle in the form of:

$$\begin{bmatrix} \begin{bmatrix} \dot{x}_1 \\ \dot{v}_1 \end{bmatrix} \\ \dots \\ \begin{bmatrix} \dot{x}_N \\ \dot{v}_N \end{bmatrix} \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} v_1 \\ f_\theta(x_1, v_1, r_1, r_1')/m_1 \end{bmatrix} \\ \dots \\ \begin{bmatrix} v_N \\ f_\theta(x_N, v_N, r_N, r_N')/m_N \end{bmatrix} \end{bmatrix}, \tag{10}$$

where

$$f_\theta(x_n, v_n, r_n, r_n') = f_\theta^{mot}(x_n, v_n) + f_\theta^{p}(r_n) + f_\theta^{o}(r_n'),$$
$$r_n = (r_{n1}, ..., r_{nK}), \quad r_n' = (r_{n1}, ..., r_{nW}).$$

Here r_n collects the relative locations of the K nearest particles to the n-th particle (i.e., $r_{ni} = x_i - x_n$), and r_n' are those of the W obstacles, which can be easily computed by a deterministic function $r(\boldsymbol{x})$. Note that in the present formulation, we omit the relative speed u_n due to the experimental phenomenon that physical forces caused by bumping and body compression are infrequent in particle interactions. In other words, the interaction force is dominated by psychological forces, which only depend on r_n according to the SFM. Following [22] we provide the graphic representation of the SF-based ODE-Net in Figure 1.

When training the ODE-Net model, all the collected data are organised in a way such that each data point consists of the states of the crowd at the initial and the end times: $\{[(\hat{x}_0^i, \hat{v}_0^i), (\hat{x}_1^i, \hat{v}_1^i)]\}_{i=1}^M$. Each data point is plugged into the procedure described in Section 2.1 to update the force functions f_θ^{mot}, f_θ^{p} and f_θ^{o}. It is worth mentioning that it is not necessary for all the data points to be collected for the same crowd; rather we just need that the crowd system remains the same within the integration time for each data point: namely, no particle enters or leaves the scene of interest from the initial time to the end time of a data point. We reinstate that this is possible thanks to the fact that all the data points are used to update the same force functions.

3 NUMERICAL EXPERIMENTS

In this section, we conduct numerical experiments to demonstrate the performance of the proposed method. Specifically we consider a typical scenario where a crowd of individuals leave a room via a single exit, as is shown in Figure 2, and synthetic data is used, which means that the true models are available for validating the results. In our experiments we first generate data from a specific computer model under the aforementioned scenario, and then learn the underlying model from the generated data using the proposed ODE-Net based method. Finally the behavior of the learned model is compared with that of the true model to assess its performance.

We generate the data with two representative models: one is continuous in time and the other is discrete. The continuous-time model adopted is the SFM mentioned earlier, and the discrete time one is the Optimal Reciprocal Collision Avoidance (ORCA) model [4], both of which are widely used in describing crowd dynamics.

3.1 Learning the social force model

We first test with the data generated from SFM. The parameter values used in our simulation are presented in Table 1, largely following [8]. Training data is the simulated trajectories of 5 pedestrians,

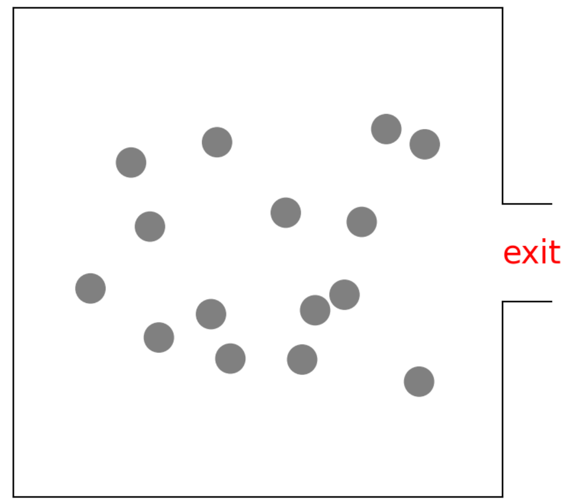

Figure 2: Schematic plot of the simulation scenario. Pedestrians are depicted by dots and walls (i.e., obstacles) are depicted by lines. All pedestrians are moving towards the center of the exit.

but as has been mentioned, the learned ODE-Net model can be applied to crowds of any size. The loss function used in all our experiments is

$$L(\boldsymbol{x}_{t_1}, \boldsymbol{v}_{t_1}; \hat{\boldsymbol{x}}_{t_1}, \hat{\boldsymbol{v}}_{t_1}) = \|\boldsymbol{x}_{t_1} - \hat{\boldsymbol{x}}_{t_1}\|_1 + \|\boldsymbol{v}_{t_1} - \hat{\boldsymbol{v}}_{t_1}\|_1,$$

where $\boldsymbol{x}_{t_1}, \boldsymbol{v}_{t_1} = \text{ODESolve}(\boldsymbol{x}_{t_0}, \boldsymbol{v}_{t_0}, f_\theta(\boldsymbol{x}, \boldsymbol{v}), t_0, t_1)$ are outputs of the ODE-Net. The training process is terminated after 30 epochs. Once the ODE model is learned, we use it to predict the dynamics of a new crowd system, and compare the results with those of the actual model.

Table 1: List of parameter values in SFM.

Variable	Value	Description
H_0	10 m	side length of the room
N	5(20)	number of pedestrians
m	80 kg	mass of pedestrians
v^p	1.0 m/s	desired velocity
τ	0.5 s	acceleration time
r	0.3 m	radius of pedestrians
A	2×10^3 N	interaction strength
B	0.08 m	interaction range
k	1.2×10^5 kg/s^2	bump effect
κ	2.4×10^5 kg/(m · s)	friction effect
Δt	0.001 s	time step in simulation

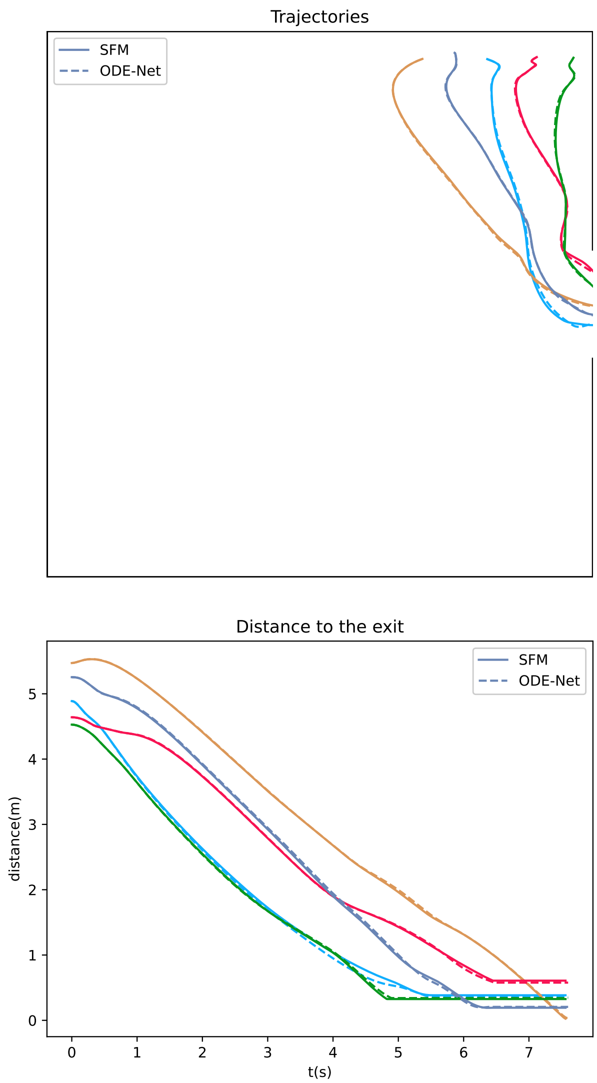

Figure 3: Comparison of SFM and the learned ODE-Net model, for a crowd of 5 pedestrians. Top: pedestrian trajectories generated with SFM (solid) and ODE-Net (dotted). Bottom: each pedestrian's distance to the exit plotted against time, where the results of SFM and of ODE-Net are shown as solid and dotted lines respectively.

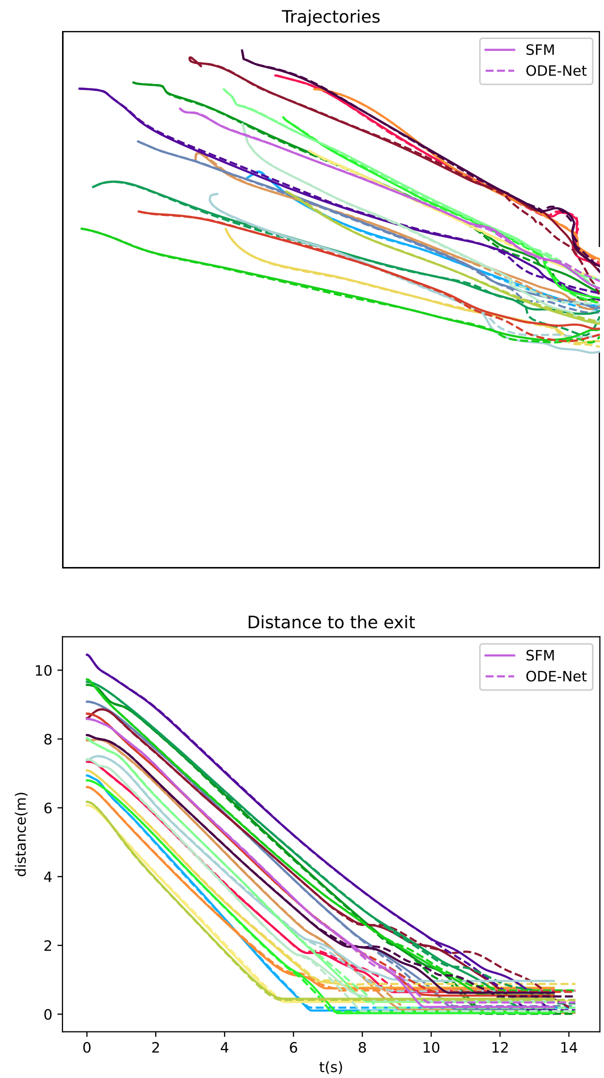

Figure 4: Comparison of SFM and the learned ODE-Net model, for a crowd of 20 pedestrians. Top: pedestrian trajectories generated with SFM (solid) and ODE-Net (dotted). Bottom: each pedestrian's distance to the exit plotted against time, where the results of SFM and of ODE-Net are shown as solid and dotted lines respectively.

We demonstrate such a comparison in Figure 3: the top figure shows the trajectories of the particles predicted by the actual SFM and the learned ODE-Net model, and the bottom one shows the particle's distance to the exit as a function of time, representing the velocity information of the particles. One can see from the figures that the results of the two models agree very well, indicating that ODE-Net can effectively learn the behaviors of the actual model in this case. We then test the learned ODE-Net model with a crowd of 20 particles, where the results are shown in Figure 4. Once again we observe good agreement between the results of the two models, suggesting that the ODE-Net trained with a crowd of 5 particles can be used to make predictions of a much larger crowd. That said, we do observe discrepancy in some trajectories near the exit, which is likely due to the physical contacts between pedestrians when they are very close to each other, and such physical contacts are not taken into account in our present social force based model. Finally, as has been mentioned earlier, compared to the behaviors of each individual particles, it is more important to examine if the

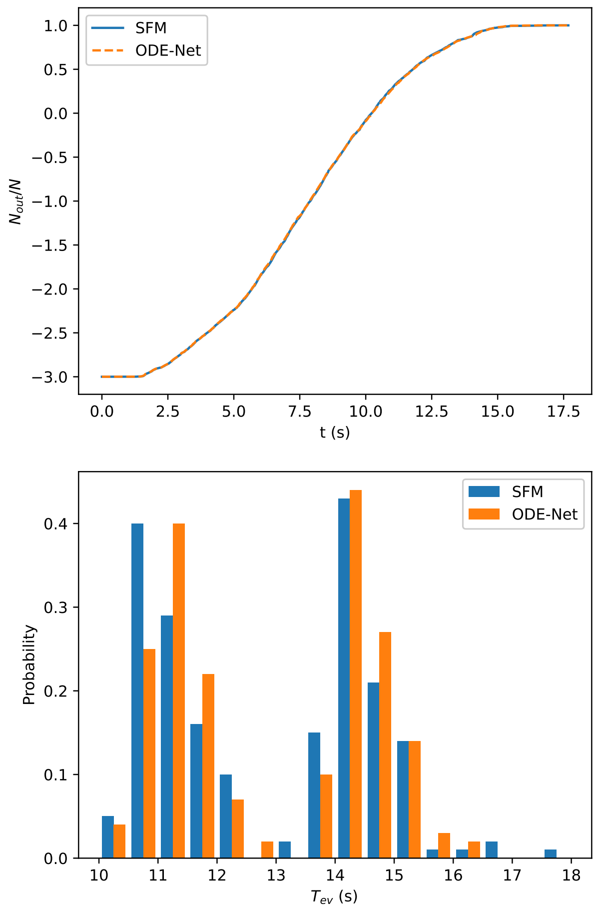

Figure 5: Comparison of SFM and the learned ODE-Net model, for a crowd of 20 pedestrians. Top: ICE plotted against time, averaged over 200 simulations. Bottom: the histograms of the evacuation time T_{ev} obtained from 200 simulations of both models.

learned model can correctly predict important statistical or collective behaviors of the crowd, as that is often what such models are used for. To this end, we consider the following two representative statistical quantities. First, we track the instantaneous collective escape (ICE) rate [7], which is defined as the percentage of the pedestrians who have successfully exited the room at a given time: N_{out}/N where N_{out} is the number of escaped pedestrians. We repeat the simulations of a crowd of 20 particles 200 times with random initial locations, and plot the average ICE rate as a function of time t in Figure 5 (top). One can see that the result of SFM and that of the learned ODE-Net model look nearly identical. Another statistical quantity that we consider is the evacuation time T_{ev}, defined as the time for all the pedestrians to leave the room [7]. We also perform 200 simulations with different initial locations, calculate the evacuation time for each simulation, and plot the histogram of it in Figure 5 (bottom). Note that, the initial locations are chosen in a way that the resulting histogram is bi-modal, to test if ODE-Net can capture this feature. One can see from the figure that, the ODE-Net model does reproduce the bi-modal feature of the histogram. Finally it is important to note that, since the actual SFM is also based on an ODE system (therefore continues-time) and the social-force concept, the ODE-Net performs very well in this experiment, thanks to the similarity between the actual model and ODE-Net. To further test the ODE-Net method, next we apply it to a discrete-time model.

3.2 Learning the ORCA model

In this section, we apply ODE-Net to the ORCA model, which is discrete in time and therefore conceptually different from ODE-Net. As a discrete-time model, at each time step, ORCA allows each pedestrian to determine independently the optimal moving velocities and move accordingly [4]. Briefly speaking, the ORCA model assumes that each pedestrian can obtain the relative distance and velocity with respect to every neighboring pedestrian at a certain time step. Based on the information, the pedestrian computes a collision-free velocity for the next step of movement, by solving a constrained optimization problem. Specifically the velocity should be the one that is closest to a prescribed target velocity, subject to the constraint that it will not cause collision with any other pedestrians or obstacles in a finite time horizon. The details of the ORCA model can be found in [4] and the parameter values used in our simulation are presented in Table 2.

Table 2: List of parameter values in ORCA.

Variable	Value	Description
H_0	10 m	side length of the room
N	5(20)	number of pedestrians
r	0.3 m	radius of pedestrians
v^p	1.0 m/s	preferred velocity
τ	0.05 s	time horizon
Δt	0.01 s	time step in simulation
maxNeighbors	10	max number of neighbors
neighborDist	2.5 m	max distance of neighbors

The training data are generated from the ORCA model for a crowd of 5 pedestrians, and the training procedure is the same as that is used in the first example. As is in the first example, our first test is to apply the obtained ODE-Net model to a crowd of 5 pedestrians. Figure 6 compares the trajectories and the distances to the exit predicted by both models, in which one can see that the results agree quite well with each other. Next in Figure 7, we show the average ICE rate and the histogram of the evacuation time, both obtained from 200 repeated trials. The plots illustrate that, although the discrepancies between the statistical results of the two models are larger than those in the first example, the results largely agree with each other, demonstrating the ability of the ODE-Net model to predict the crowd behaviors in this case.

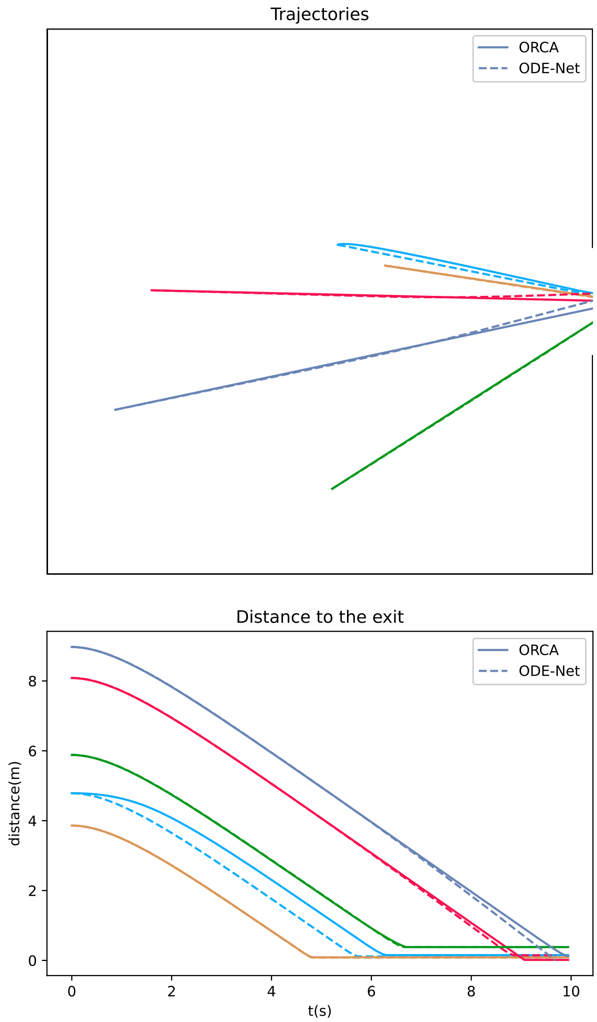

Figure 6: Comparison of ORCA and the learned ODE-Net model, for a crowd of 5 pedestrians. Top: pedestrian trajectories generated with ORCA (solid) and ODE-Net (dotted). Bottom: each pedestrian's distance to the exit plotted against time, where the results of ORCA and of ODE-Net are shown as solid and dotted lines respectively.

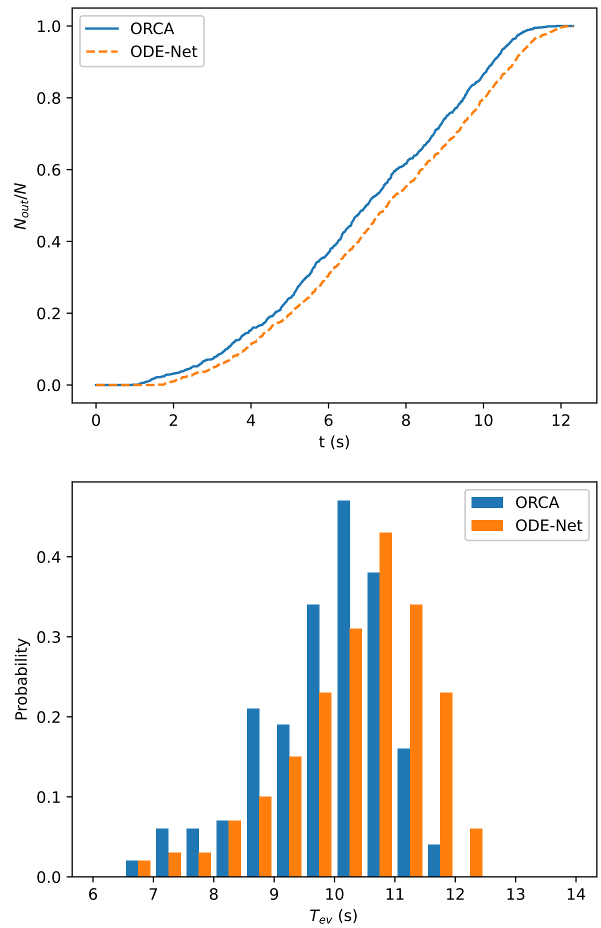

Figure 7: Comparison of ORCA and the learned ODE-Net model, for a crowd of 5 pedestrians. Top: ICE plotted against time, averaged over 200 simulations. Bottom: the histograms of the evacuation time T_{ev} obtained from 200 simulations of both models.

Next we consider a scenario that is more challenging for the obtained ODE-Net model, where we apply it to a crowd of 20 pedestrians that are closely spaced at the beginning. The particle trajectories and the distances to the exit are plotted in Figure 8, where we see that the difference between the results of the two models become more substantial than that in the previous case, especially for the distance to the exit that represents the velocity information of the particles. Similar conclusions can be drawn from Figure 9, which shows the ICE rate and the histogram of the evacuation time. In particular, we have found that while the ODE-Net captures the bimodal feature of the histogram, it seems to predict less variation in the results than the ORCA model. We believe that the larger discrepancy in this example is due to the fact that the ODE-Net and the ORCA models are different in nature: first and foremost, one model is continuous in time and the other is discrete; moreover, how the interactions between particles take place in the two models is fundamentally different. We expect that the agreement can be improved by designing specific network structures according to the ORCA model, which is subject to further investigation.

4 CONCLUSION

In this work we present an ODE-Net based method to learn continuous-time models of crowd dynamics from data. In particular, we formulate the pedestrians as particles driven by several socio-psychological forces, which are learned from the data. With the proposed method, we are able to learn models for large-scale and variable-size crowds, and use the learned model to make predictions for crowds that are

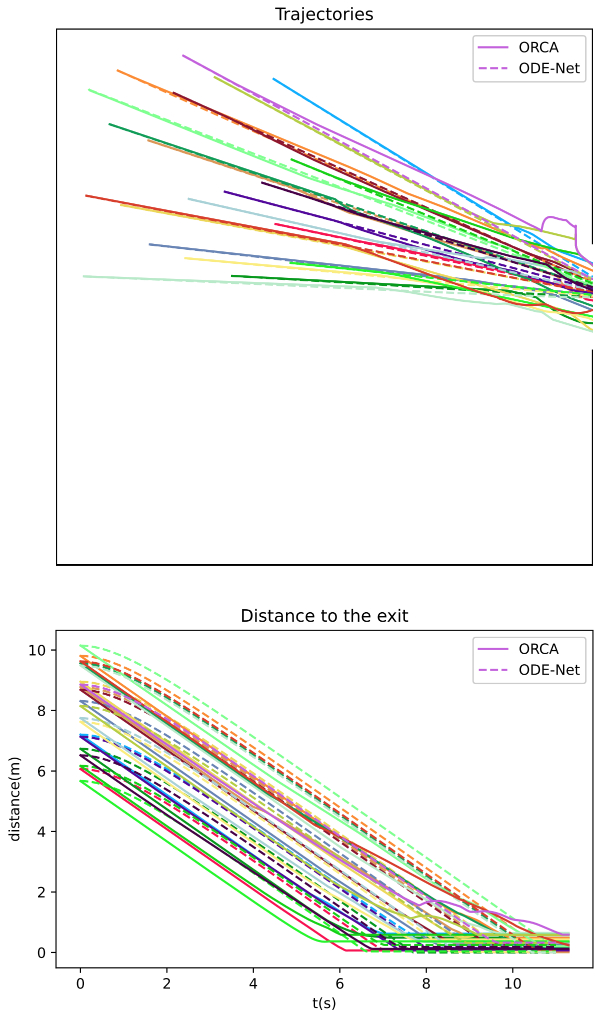

Figure 8: Comparison of ORCA and the learned ODE-Net model, for a crowd of 20 pedestrians. Top: pedestrian trajectories generated with ORCA (solid) and ODE-Net (dotted). Bottom: each pedestrian's distance to the exit plotted against time, where the results of ORCA and of ODE-Net are shown as solid and dotted lines respectively.

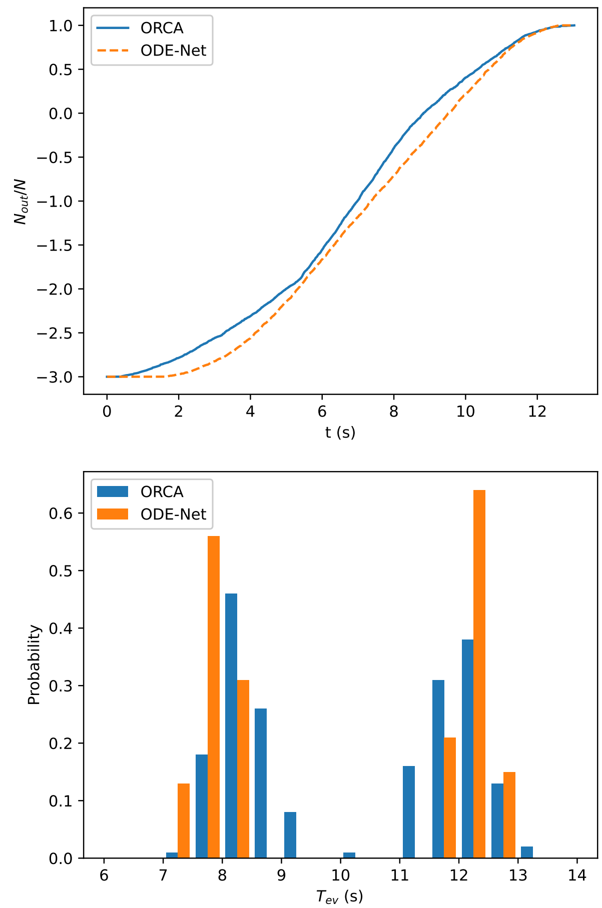

Figure 9: Comparison of ORCA and the learned ODE-Net model, for a crowd of 20 pedestrians. Top: ICE plotted against time, averaged over 200 simulations. Bottom: the histograms of the evacuation time T_{ev} obtained from 200 simulations of both models.

larger or smaller than the training system. The performance of the proposed method is demonstrated by applying it to the synthetic data generated by two popular crowd dynamics models – SFM and ORCA. We believe that the proposed method can be useful in a range of applications, such as urban safety planning and human-robotic interaction.

Several potential improvements of the proposed method are possible. First, considering the limited view field of humans, interaction force between pedestrians might also depend on their facing directions. In particular, pedestrians would pay more attention to pedestrians in front of them than those behind them [12], and such an effect should be taken into account when constructing the interaction force model. Second we here assume that the personal motivation forces depend only on the particle's velocity and position, which is certainly a simplification, and as is discussed in Section 2.2, environmental factors should also be taken into account. Finally, empirical studies show that a large fraction of people in a crowd move in small groups, such as friends walking together [14], the effect of which is not considered in the present work. To this end, it is desirable for the method to be able to learn the group effects from data, and enhance the performance of the resulting model. We plan to investigate these potential improvements in future studies.

REFERENCES

[1] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. 2016. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 961–971.

[2] Javad Amirian, Wouter Van Toll, Jean-Bernard Hayet, and Julien Pettré. 2019. Data-driven crowd simulation with generative adversarial networks. In *Proceedings of the 32nd International Conference on Computer Animation and Social*

Agents. 7–10.

[3] Alessandro Antonucci, Gastone Pietro Rosati Papini, Paolo Bevilacqua, Luigi Palopoli, and Daniele Fontanelli. 2022. Efficient Prediction of Human Motion for Real-Time Robotics Applications With Physics-Inspired Neural Networks. *IEEE Access* 10 (2022), 144–157.

[4] Jur van den Berg, Stephen J Guy, Ming Lin, and Dinesh Manocha. 2011. Reciprocal n-body collision avoidance. In *Robotics research*. Springer, 3–19.

[5] Ricky TQ Chen, Brandon Amos, and Maximilian Nickel. 2021. Learning Neural Event Functions for Ordinary Differential Equations. In *International Conference on Learning Representations*.

[6] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. 2018. Neural ordinary differential equations. *Advances in neural information processing systems* 31 (2018).

[7] Chen Cheng, Jinglai Li, and Zhenwei Yao. 2021. Ordering leads to multiple fast tracks in simulated collective escape of human crowds. *Soft Matter* 17 (2021), 5524–5531. Issue 22.

[8] Dirk Helbing, Illés Farkas, and Tamás Vicsek. 2000. Simulating dynamical features of escape panic. *Nature* 407, 6803 (2000), 487–490.

[9] Dirk Helbing and Anders Johansson. 2011. *Pedestrian, Crowd and Evacuation Dynamics*. Springer New York, New York, NY, 697–716.

[10] Zijie Huang, Yizhou Sun, and Wei Wang. 2020. Learning continuous system dynamics from irregularly-sampled partial observations. *Advances in Neural Information Processing Systems* 33 (2020), 16177–16187.

[11] Anders Johansson, Dirk Helbing, Habib Z Al-Abideen, and Salim Al-Bosta. 2008. From crowd dynamics to crowd safety: a video-based analysis. *Advances in Complex Systems* 11, 04 (2008), 497–527.

[12] Anders Johansson, Dirk Helbing, and Pradyumn K Shukla. 2007. Specification of the social force pedestrian model by evolutionary adjustment to video tracking data. *Advances in complex systems* 10, supp02 (2007), 271–288.

[13] Mehdi Moussaïd, Dirk Helbing, Simon Garnier, Anders Johansson, Maud Combe, and Guy Theraulaz. 2009. Experimental study of the behavioural mechanisms underlying self-organization in human crowds. *Proceedings of the Royal Society B: Biological Sciences* 276, 1668 (2009), 2755–2762.

[14] Mehdi Moussaïd, Niriaska Perozo, Simon Garnier, Dirk Helbing, and Guy Theraulaz. 2010. The walking behaviour of pedestrian social groups and its impact on crowd dynamics. *PloS one* 5, 4 (2010), e10047.

[15] Yulia Rubanova, Ricky T. Q. Chen, and David K Duvenaud. 2019. Latent Ordinary Differential Equations for Irregularly-Sampled Time Series. In *Advances in Neural Information Processing Systems*, Vol. 32. Curran Associates, Inc.

[16] D. Solomatine, L.M. See, and R.J. Abrahart. 2008. *Data-Driven Modelling: Concepts, Approaches and Experiences*. Springer Berlin Heidelberg, Berlin, Heidelberg, 17–30.

[17] Ming Tang and Hongfei Jia. 2011. An approach for calibration and validation of the social force pedestrian model. In *Proceedings 2011 International Conference on Transportation, Mechanical, and Electrical Engineering (TMEE)*. IEEE, 2026–2031.

[18] Peter Trautman and Andreas Krause. 2010. Unfreezing the robot: Navigation in dense, interacting crowds. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 797–803.

[19] Pete Trautman, Jeremy Ma, Richard M Murray, and Andreas Krause. 2015. Robot navigation in dense human crowds: Statistical models and experimental studies of human–robot cooperation. *The International Journal of Robotics Research* 34, 3 (2015), 335–356.

[20] David Wolinski, S J. Guy, A-H Olivier, Ming Lin, Dinesh Manocha, and Julien Pettré. 2014. Parameter estimation and comparative evaluation of crowd simulations. In *Computer Graphics Forum*, Vol. 33. Wiley Online Library, 303–312.

[21] Shuai Yi, Hongsheng Li, and Xiaogang Wang. 2016. Pedestrian behavior understanding and prediction with deep neural networks. In *European Conference on Computer Vision*. Springer, 263–279.

[22] Yaofeng Desmond Zhong, Biswadip Dey, and Amit Chakraborty. 2020. Symplectic ODE-Net: Learning Hamiltonian Dynamics with Control. In *International Conference on Learning Representations*.