PI-NeuGODE: Physics-Informed Graph Neural Ordinary Differential Equations for Spatiotemporal Trajectory Prediction

Zhaobin Mo
zm2302@columbia.edu
Department of Civil Engineering and Engineering Mechanics, Columbia University
New York, USA

Yongjie Fu
yf2578@columbia.edu
Department of Civil Engineering and Engineering Mechanics, Columbia University
New York, USA

Xuan Di*
sharon.di@columbia.edu
Department of Civil Engineering and Engineering Mechanics, Columbia University
New York, USA

ABSTRACT

It is challenging to predict a group of individuals' spatiotemporal trajectories in continuous time and space, due to various environmental and intrinsic factors. Especially, social dynamics such as driving or crowding behaviors could be hard to predict due to heterogeneous and complex mapping from high-dimensional inputs to an output driven by the decision-making processes of other agents. To tackle this challenge, neural ordinary differential equations (neural ODEs) have been developed to predict continuous-time long-term dynamics with constant memory cost and high computational efficiency. Furthermore, scientific communities have developed a rich set of physics models to describe how individuals interactively make decisions. With a rapidly growing trend of employing physics-informed deep learning (PIDL) for dynamical systems in science and engineering, its application to social dynamics is understudied. This paper aims to develop an integrated framework, named "PI-NeuGODE," that encodes physics models, complemented by symbolic regression, into neural ODEs. In the proposed model, physics informs the training of neural ODEs, while neural ODEs guide knowledge discovery. Symbolic regression is used to uncover physics knowledge from complex data. We further use graph neural networks to learn the topological interaction of individuals. The proposed method is tested on two applications, human driving and platooning, as well as crowding, which demonstrate the algorithmic accuracy and efficiency against baselines including existing social deep learning models.

KEYWORDS

Neural Ordinary Differential Equations; Physics-Informed Deep Learning; Graph Neural Network; Symbolic Regression

ACM Reference Format:
Zhaobin Mo, Yongjie Fu, and Xuan Di. 2024. PI-NeuGODE: Physics-Informed Graph Neural Ordinary Differential Equations for Spatiotemporal Trajectory Prediction. 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.

*This author is also affiliated with the Data Science Institute of Columbia University, New York, USA.

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).

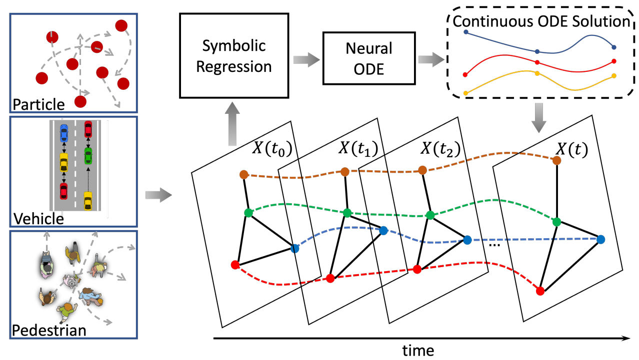


Figure 1: Physics-informed graph neural ordinary differential equations (PI-NeuGODE). The multi-agent dynamics are captured by graph neural networks (for spatial interaction) and neural ordinary equations (for temporal evolution). The symbolic regression is used to infer the underlying physics that informs the training of the data-driven counterpart.

1 INTRODUCTION

Prediction of a group of interacting individuals' dynamic trajectories has various applications, such as autonomous fleet management [12, 19], crowd management[44], traffic signal control [26] and ramp metering [22, 23]. Challenges involved in such a prediction lie in the multi-agent nonlinear interaction for a long-term prediction horizon, because human behaviors could suffer from stochasticity and instability [10] arising from the multi-agent interaction. For example, in a social environment, individuals' spatiotemporal trajectories are governed not only by their own underlying decision-making processes, but also by various external factors such as surrounding actors' constantly changing behaviors and stochastic environmental features. Nevertheless, individuals interact in a nonlinear fashion that complicates the learning of a mapping from environmental features to an individual output. Furthermore, social dynamics could be non-stationary or non-Markovian, especially in a multi-agent system when one's action is affected by surrounding agents, which makes long-term (or n-step ahead) predictions challenging.

Many methods have been introduced to tackle the aforementioned non-linearity and long-term prediction challenges, which broadly bifurcate into two branches. The **physics-based** method makes assumptions about the underlying physics that governs observations, e.g. partial differential equations (PDE), and uses mathematical equations to formulate the pedestrian interactions and intentions. For example, the Social Force Model (SFM) [16] assumes pedestrians navigate through their surroundings by responding to a "social force", which is a combination of external and internal forces. These forces include factors like social interactions, repulsion from obstacles, and attraction towards a destination. Apart from the social force assumption, another widely used assumption is that pedestrians make decisions mainly by following neighbouring velocities and avoiding collision, such as the velocity obstacle model [41].

In contrast to the physics-based method, the **data-driven** method does not assume any prior knowledge of the underlying process governing the pedestrian dynamics. Instead, it utilizes machine learning methods to uncover the underlying patterns directly from the observed data. For example, graph neural networks, such as message passing networks [5], are widely used to capture spatial interactions. For temporal dynamics, Long Short-Term Memory (LSTM) [1] networks and attention mechanisms [14, 42] are broadly employed. Notably, neural ordinary differential equations (Neural ODE) [6] has demonstrated its proficiency in capturing dynamic system patterns, particularly in its ability to extrapolate and address the long-term prediction challenge.

Both the physics-based and data-driven methods have their pros and cons. The former is data-efficient and interpretable but may struggle with generalizing to unobserved data. The latter is generalizable but may be incapable of offering deductive insights. Also, its performance relies on huge amounts of training samples, which is sometimes inaccessible for real-world data. The fusion of these two method categories, known as **physics-informed deep learning** (PIDL) [11, 32], combines the merits of both the physics-based and data-driven methods and supplements their respective limitations. PIDL has been applied to multiple applications, such as car-following modeling [24, 27], crowd simulation [44], and traffic state estimation [17, 25, 35–37]. While promising, the performance of PIDL requires the assumption of the underlying physics, which is usually not available in the real-world scenario. The problem of unknown underlying physics prior knowledge is worse for multi-agent systems [13]. A potential remedy is symbolic regression, which is capable of inferring the mathematical equations that govern the dynamics. Symbolic regression has demonstrated its efficacy across diverse domains, such as materials science [3, 4] and astronomy[8].

This paper proposes, **physics-informed graph neural ordinary differential equations** (**PI-NeuGODE**), for simultaneous prediction of multi-agent spatiotemporal trajectories accounting for the topological interaction over a long-term horizon. Under this framework, neural ODEs are trained to make long-term prediction of individual trajectories interacting over graphs, of which the derivatives are substituted by physics-informed deep neural networks (DNN) complemented by symbolic regression.

2 BACKGROUND AND RELATED WORK

2.1 Long-term prediction using neural ODEs

Neural ordinary differential equations (neural ODEs) are developed to predict continuous-time long-term dynamics, with constant memory cost and high computational efficiency [6]. Since its inception, neural ODEs have become a powerful tool to model complex dynamical systems. The underlying idea is to parameterize the derivative of a system's state with a neural network (NN).

A neural ODE consists of an ordinary differential equation (ODE) of the form:

$$\frac{dx}{dt} = f_\theta(x(t), t, \theta), \tag{1}$$

where $x(t)$ is the system state at time t, $\frac{dx}{dt}$ is the time derivative, $f_\theta(x(t), t, \theta)$ is parameterized by an NN with parameters θ. The loss function to train the NN is defined as:

$$\begin{aligned} L(x(t)) &= L(x(t_0)) + \int_{t_0}^{t} f_\theta(x(t), t, \theta)dt \\ &= L(ODESolve(x(t_0), f_\theta, t_0, t, \theta)). \end{aligned} \tag{2}$$

A numerical integration method, such as the Euler method or the Runge-Kutta method, is used to solve the differential equation and produce a prediction.

2.1.1 Message passing neural networks (MPNNs). MPNNs are a class of graph neural networks (GNNs) that leverage the principles of message passing algorithms to perform graph-based tasks. In MPNNs, each node passes messages to its neighbours, and these messages are used to update the node representations and aggregate information from the surrounding nodes.

Mathematically, MPNNs are modeled as functions that operate on a graph $G = (V, E)$, where V is the set of nodes and E is the set of edges. Each node is initialized with states $x_i(t_0)$, where $i \in \{1, \dots, N\}$ and N is the totoal number of agents. At iteration t, the node state $x_i(t+1)$ is updated as follows:

$$x_i(t+1) = f_u(x_i(t), \sum_{\substack{j \in \{1,\dots,N\} \\ (i,j) \in E}} m_{ij}(t)), \tag{3}$$

where $f_u(\cdot)$ is the node update function, and $m_{ij}(t)$ is the message passed from node j to node i at iteration t. The messages $m_{ij}(t)$ are computed as follows:

$$m_{ij}(t) = f_m(x_i(t), x_j(t), e_{ij}), \tag{4}$$

where e_{ij} is the edge information between nodes i and j and $f_m(\cdot)$ is the message passing function. The process continues until a stopping criterion is met, such as a maximum number of iterations or convergence.

2.1.2 Graph neural ODE (GODE). GODE leverages the strengths of graph neural networks (GNNs) and neural ODEs to model the complex interactions in dynamic systems. MPNN captures the spatial correlation between nodes, while the neural ODE captures the temporal evolution of the system. These two components are coupled by replacing f in Eq. 1 with the message passing process defined in Eq. 4:

$$\dot{x}_i(t) = f_\theta(x_i(t), f_m(x_i(t), x_j(t), e_{ij}(t)), t, \theta). \tag{5}$$

Further details on the implementation of this approach will be discussed in Sec. 3.

2.2 Physics-informed deep learning (PIDL)

PIDL [32] leverages the pros of both physics-based and data-driven approaches while compensating for the cons of each. Physics-based approach refers to scientific hypotheses of what underlying physics governs observations, like the first principle, which is data-efficient and interpretable but may not well capture complex data patterns.

In contrast, the data-driven approach does not bear any prior knowledge of how things work and how different quantities are correlated. Instead, they rely on machine learning techniques such as deep neural networks (DNN) to learn and infer patterns from huge amounts of training samples, but require inductive bias for unseen data. Recent years have seen a rapidly growing trend of applying PIDL to dynamical systems in science and engineering, for its power in robust prediction [2, 9, 20].

To incorporate PIDL into neural ODE, the derivative of a dynamical system contains both the known (represented by physics) and unknown information (represented by PUNN). [28, 38] further incorporates stochasticity into the prediction. These studies, however, are not focused on social dynamics but more on physical processes.

2.3 Symbolic regression

Symbolic regression aims to discover mathematical expressions to match a given dataset [34]. When combined with DNNs informed by physics, symbolic regression demonstrates its robustness in discovering part of physics equations [30, 33, 40, 44].

The mathematical functions are usually represented as a general expression with variables and operators. The task of symbolic regression is to find the optimal values of the coefficients that result in the best fit of the function to the observed data. To perform symbolic regression, some optimization algorithm is applied to the function and the coefficients are iteratively adjusted until the best fit is obtained. Metrics, such as mean squared error or correlation coefficient, are used to determine the goodness of fit of the function to the observed data.

2.4 Related work and contributions

In a nutshell, there are studies that have integrated PIDL into symbolic regression [30, 33, 40, 44, 44], PIDL into neural ODEs [28, 38], or GNN into neural ODEs [7, 18, 31, 45]. However, none has bridged all three methodologies into one unified framework, which is the focus of this paper. The closely related studies are [18, 43, 44]. [18] combined two neural ODEs, one for temporal processing and the other for spatial processing, for spatiotemporal traffic forecasting. However, the existing domain knowledge that uses ODEs/PDEs for traffic evolution is not accounted for. [44] applied PIDL, symbolic regression, and GNN to crowd modeling, and a student-teacher co-training algorithm was developed for multiple-step rollout, which helps generalize the original single-step prediction. The same issue exists in [43], where the one-step prediction is repeatedly conducted to achieve the long-term prediction. In contrast, our paper employs neural ODE for multiple-step prediction, which is more computationally efficient and scalable.

Our **main contributions** are: **(1)** we develop a unified framework of integrating PIDL and symbolic regression into graph neural ODEs, which can capture multi-agent, long-term prediction of social dynamics. **(2)** we develop an algorithm to train the data-driven component (i.e., GODE) and the physics-informed components (i.e., PIDL) simultaneously, rather than calibrate the physics models prior to the model training, and **(3)** the efficiency of the proposed algorithms is demonstrated on two types of dynamics, platooning and crowding, using both hypothetical and real-world datasets.

The rest of this paper is organized as follows: Sec. 3 introduces the integrated methodology framework and presents the algorithm. Secs. 4-5 demonstrate the performance of our algorithms on two scenarios, human driving and platooning (via a neighbouring interaction), and crowding (via a graph interaction). Sec. 6 concludes and points out future directions.

3 METHODOLOGY

3.1 Problem statement

Denote the system state as $X(t) = \{x_i(t)\}_{i=1}^N$, which contains the states of all agents. The movement of an agent at each time step t is assumed to be governed by some underlying control signal. Thus, the spatiotemporal trajectory is the result of the consecutive control signals or decision-making processes. Consider the multi-agent scenario, this process can be depicted as:

$$X(t) = X(t_0) + \int_{t_0}^{t} f(X(t), t, \theta) dt, \tag{6}$$

where $f(\cdot)$ is the underlying decision-making model that is inaccessible. The trajectory prediction problem is to find a surrogate decision-making model f_θ such that the predicted state error is minimized:

$$\begin{aligned} &\min_\theta \int_{t_0}^{t} (X(t) - \hat{X}(t))^2 \\ s.t. \quad &\hat{X}(t) = X(t_0) + \int_{t_0}^{t} f_\theta(X(t), t, \theta) dt. \end{aligned} \tag{7}$$

We focus on two problems of the spatiotemporal trajectory predictions, i.e., the initial value problem (IVP) and the sequence-to-sequence (seq2seq) learning problem. Each problem is detailed below.

(1) IVP involves making predictions based on only the initial states of the system, represented by $X(t_0)$, and the boundary conditions. This approach can be viewed as a trajectory prediction problem with limited data, as it only takes into account the initial states but not the historical information of the system.
(2) Seq2seq prediction utilizes historical observations of the system to make predictions about future states. In this approach, the observed trajectory is divided into segments, with the goal of using the previous segments to predict the next ones.

The IVP and seq2seq problems will be demonstrated using the platoon modeling and pedestrian trajectory prediction experiments, respectively. Both numerical and real-world datasets will be used in both experiments.

3.2 Model architecture

The proposed method, as depicted in Fig. 2, consists of two components. The first component is the data-driven component, which incorporates a neural ODE model, encoded with a message-passing type graph neural network. The second component is the physics-informed component, which includes a partially learned physics equation obtained through symbolic regression. The integration of these two components allows for a well-balanced solution between data-driven and physics-based approaches.

3.2.1 Graph neural ODEs (GODE). The GODE structure is illustrated in Fig. 2. To simplify the notation, we utilize the symbols

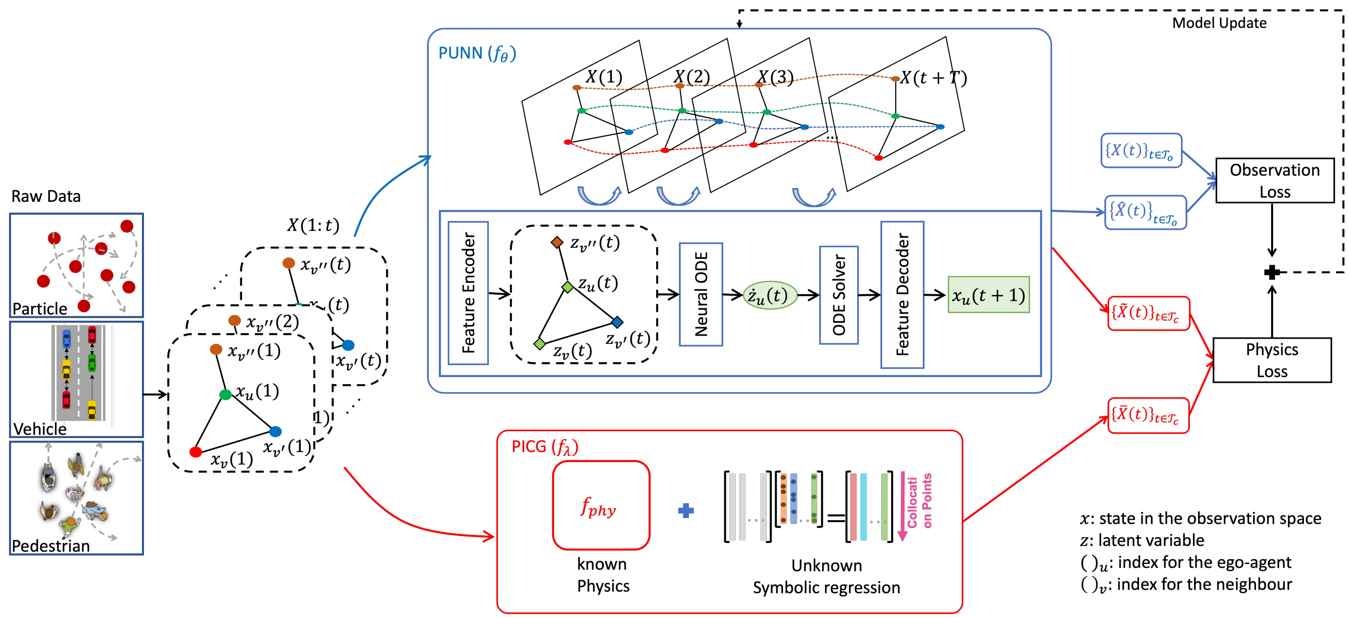


Figure 2: Architecture of our proposed method.

$1, \ldots, t$ to represent the time index $t_0, \ldots, t$. The GODE model consists of three main components: an encoder, a neural ODE model, and a decoder. In this framework, the encoder replaces the message passing function in Eq. 4 by transforming the historical states segment $x(1:t)$ into a latent variable $z_u(t)$ of the ego-agent, while taking into account the states of its neighbours. This latent variable is then used as the initial condition for the neural ODE model, which calculates the future latent solution $z_u(t+1:t+T)$. Finally, the decoder converts the latent solution back to the data space. The encoder and decoder can be implemented as either multi-layer perceptrons or recurrent networks like GRUs. The implementation details of these components for IVP and seq2seq problems are described in Secs. 4 and 5, respectively. The estimated trajectory $\hat{X}(t+1:t+T)$ is compared to the ground truth, and the observation loss is calculated accordingly:

$$L_o(\theta) = \frac{1}{T}\sum_{t=t+1}^{t+T}((\hat{X}(t)) - X(t))^2 dt. \tag{8}$$

3.2.2 *Physics-informed deep learning.*

Definition 3.1. Physics-informed deep learning (PIDL). Denote the (labeled) observation O and the (unlabeled) collocation points C below:

$$\begin{cases} O = \{X(t); t \in \mathcal{T}_o\} : \textit{within-domain observation}, \\ C = \{\bar{X}(t) = ODESolve(X(t_0), f_\lambda, t_0, t, \lambda); t \in \mathcal{T}_c\} : \textit{collocation points}, \end{cases} \tag{9}$$

where, $\mathcal{T}_o$ is a set of time steps at which the state $X(t)$ is observed; $\mathcal{T}_c$ is a set of time steps at which a physics-informed computational graph (PICG) $f_\lambda(\cdot)$ has a solution; λ is the parameter of $f_\lambda(\cdot)$.

We design two neural networks: (1) a physics-uninformed neural network (PUNN), denoted as $f_\theta(X(t), t|\theta)$, to predict the time derive of $X(t)$, denoted as $\dot{X}(t)$, and (2) a physics-informed computational graph (PICG), denoted as $f_\lambda(X(t), t|\lambda)$, for computing the time derive of $\tilde{X}(t)$, denoted as $\dot{\tilde{X}}(t)$.

In summary, a PIDL model, denoted as $f_{\theta,\lambda}(X(t), t|\theta, \lambda)$, is commonly represented by a hybrid of two graphs, namely, the PUNN and the PICG. It aims to train an optimal parameter set θ^* for PUNN and an optimal parameter set λ^* for PICG. For simplicity, we let observation and collocation time steps be the same, i.e. $\mathcal{T}_o = \mathcal{T}_c = [t+1, \ldots, t+T]$.

In our framework, the PUNN is the GODE model introduced above. The PICG is spanned by physics knowledge, in which nodes are mathematical quantities, edges are operators connecting two quantities, and a path represents a relation from a starting quantity to a target one [27]. PICG could be represented by a graph with known physics equations, or partial knowledge complemented by additional mathematical operators to be learned via symbolic regression [44].

In summary, the PUNN is replaced by GODE that represents a sequential decision-making process on graphs. The PICG is expanded with symbolic regression to learn unknown physics knowledge that could dominate the partial information from observational data. PICG can be generally represented by $f_\lambda(x) = f_{phy}(x) + w^T x$, where $f_{phy}(\cdot)$ is the known physics from prior knowledge; x is a vector of input variables and w is a vector of coefficients together with numerical operators such as "+", "-", "×", and"/" to be learned by symbolic regression. Mathematically, the PICG takes collocation points as input and produces augmented solutions, i.e., $\bar{X}(t) = X(t_0) + \int_{t_0}^{t} f_\lambda(X(t), t, \lambda)dt$. These collocation points are also fed into the PUNN for the solution $\tilde{X}(t) = X(t_0) + \int_{t_0}^{t} f_\theta(X(t), t, \theta)dt$. Based on these two solutions, the physics loss is defined as:

$$L_c(\theta, \lambda) = \frac{1}{T}\sum_{t=t+1}^{t+T}((\tilde{X}(t)) - \bar{X}(t))^2. \tag{10}$$

3.3 Training Algorithm

The final loss function is defined as

$$Loss_{\theta,\lambda} = \alpha \cdot L_o(\theta) + \beta \cdot L_c(\theta, \lambda), \quad (11)$$

where α, β are weights of the observation loss and the physics loss, respectively. The detailed training algorithm is shown in Alg. 1.

Algorithm 1 Training Algorithm for PI-NeuGODE.

Required: Training iterations $Iter$; Learning rate lr; Loss function weights α, β.

Input: The observation data $\{(X(t), t \in \mathcal{T}_o\}$ and collocation points $\{(X(t), t \in \mathcal{T}_c\}$.

```
for k ∈ {0, ..., Iter} do
    Sample the sequence states X(1 : t, t + 1 : t + T), from the batch.
    // generate PUNN solutions
    z(t) = Encoder(X(t))
    z(t + T) = z(t) + ∫_{t+T}^{t} f_θ(z(t), t, θ)dt
    X̂(t + T) = Decoder(z(t + T))
    Generate X̂(t + 1 : T) similarly using the collocation data
    // generate PICG solutions
    X̄(t + T) = X(t) + ∫_{t+T}^{t} f_λ(X(t), t, λ)dt
    // update the PUNN
    Calculate L_θ by Eq. 8, L_c by Eq. 10
    θ^{k+1} ← θ^k − lr · Adam(θ^k, ∇_θ(αL_o + βL_c))
    // update the PICG
    Conduct symbolic regression by minimizing L_c
end for
```

In the subsequent sections, we present a comprehensive evaluation of the proposed model through two experiments. Firstly, we assess its performance in solving the IVP problem in the context of vehicle platoon modeling, utilizing both numerical and real-world data. The numerical experiment is designed to verify the ability of the proposed model to accurately identify the ground-truth equation in a controlled environment. This is followed by a validation experiment, which uses real-world data to demonstrate the model's efficacy in capturing real-world dynamics. Finally, we demonstrate the proposed model's ability to solve the seq2seq problem by applying it to the pedestrian trajectory prediction problem, using both numerical and real-world data. We will also present the specific architectures employed for each problem.

4 EXPERIMENTS: PLATOON MODELING

We first briefly introduce the problem formulation as follows: In a Platoon Modeling (PM) problem, a platoon of vehicles forms a multi-agent system, where each vehicle decides to accelerate, brake or cruise depending on its relation with its leader. Denote the state of the ith vehicle at time step t as $x_i(t) = [p_i(t), v_i(t)]$, where $p_i(t)$ is the longitudinal position and $v_i(t)$ is the velocity. The dynamic of the followers can be depicted as :

$$\begin{cases} \dot{p}_i(t) &= \frac{dp_i(t)}{dt} = v_i(t) \\ \dot{v}_i(t) &= \frac{dv_i(t)}{dt} = f_{u,i}(x_{i-1}, x_i) \end{cases}, i \in \{1, ..., N\}. \quad (12)$$

where $f_{u,i}$ is the control strategy of the ith agent given the states of its leader and itself. Given the trajectory of the first leading vehicle $x_0(1 : t + T)$, the task of the PM is to predict the trajectory of all its followers. Thus, the state of the PM system at time t is $X(t) = \{x_i(t)\}_{i=1}^N \in \mathbb{R}^{N\times 2}$.

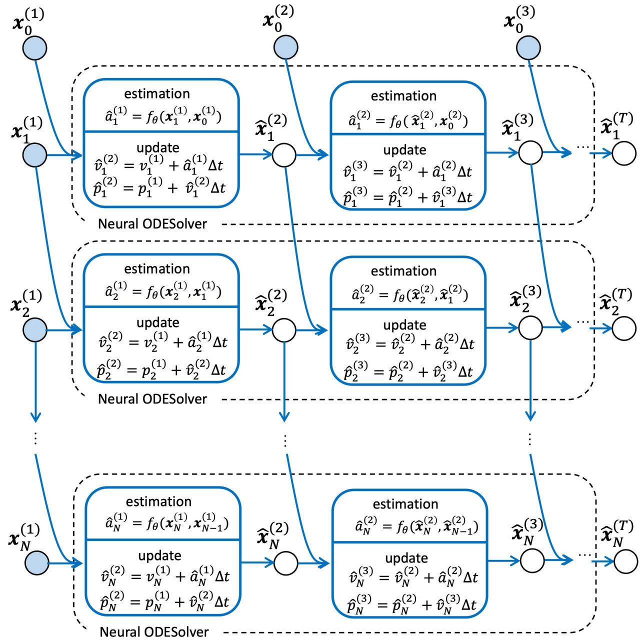


Figure 3: Structure of GODE for platoon modeling as an IVP.

The PM is an example of IVP, where only the initial state $X(1)$ and the platoon leader's trajectory $x_0(1 : t + T)$ are known. A platoon can be represented as a chain-like graph, where each vehicle only has one neighbouring vehicle, which is its immediate leader.

4.1 GODE Structure

Fig. 3 illustrates the GODE structure that is modified for the PM. The initial states of a vehicle and its leader are used as input to update the state of the follower. This GODE framework for IVP does not require an encoder or decoder as only one-time-step state information is used. The direct output of the network f_θ already contains meaningful information, as it represents the derivatives of the vehicle states. In the neural ODESolver, a neural network f_θ is trained to learn the mapping from the features of two consecutive vehicles to estimated acceleration $\hat{a}$ of the follower. The estimated acceleration $\hat{a}$ is used to update the state of the platoon by the Euler method.

4.2 Dataset

Numerical Dateset: In our numerical experiment, the platoon data is generated by the intelligent driving model (IDM) [39], a well-known model in the transportation domain. The IDM models the longitudinal motion of vehicles on a highway, considering their acceleration and deceleration behavior based on their speed and the gap to the leading vehicle. We assume each vehicle in the platoon follows the IDM equation. The complete form of the IDM equation is shown below:

$$\begin{cases} \dot{x}_i(t) = v_i(t) \\ \dot{v}_i(t) = a_{\max}\left[1 - \left(\frac{v_i(t)}{v_{\text{des}}}\right)^4 - \left(\frac{s_0 + v_i(t)T_0 + v_i(t)\Delta v_i(t)/2\sqrt{a_{\max}b}}{s_i(t)}\right)^2\right] \\ i \in \{1, ..., N\}, \end{cases} \tag{13}$$

where, $x_i(t)$ is the position of the ith vehicle at time step t, $v_i(t)$ is the velocity, $s_i = x_{i-1} - x_i$ is the gap to the leading vehicle, and $\Delta v_i = v_{i-1} - v_i$ is the velocity difference. This equation has 5 parameters: v_{des} is the desired velocity, T_0 is the desired time headway, s_0 is the minimum spacing in congested traffic, $a_{\max}$ is the maximum acceleration allowed, and b is the comfortable deceleration, respectively. All vehicles are assumed to share the same set of parameters, i.e., $[v_{\text{des}}, T_0, s_0, a_{\max}, b] = [30\ m/s, 0.75\ s, 2.15\ m, 2\ m/s^2, 4\ m/s^2]$. More numerical dataset details are included in the supplementary material.

Real-world Dataset: The real-world data is from the Next Generation SIMulation (NGSIM) dataset[1], which is an open dataset that collects vehicle trajectories every 0.1 seconds. We focus on the US Highway 101. More dataset details are shown in the supplementary material.

4.3 Setting

Baselines: In this study, we compare the performance of our proposed model with several baseline models including SocialGAN [15] and SocialLSTM [1]. The SocialGAN model is a generative adversarial network (GAN) based approach for modeling the interactions between pedestrians in a crowd. The SocialLSTM model is a long short-term memory (LSTM) network that is trained to predict the future trajectories of individuals in a crowd by incorporating the interactions between individuals. Furthermore, we also consider the simple baseline of independent decision-making, which is IDM, where each vehicle acts as an independent agent following its own trajectory computed by Eq. 13.

Evaluation Metrics: In this study, we evaluated the performance of our proposed model using mean squared error (MSE) and mean absolute error (MAE). Detailed settings are included in the supplementary material.

4.4 Results

Performance Comparison: In Tab. 1, we make a comparison of four models: PI-NeurODE, SocialGAN, SocialLSTM, and IDM by looking into MSE and MAE. We leverage noise-free and noisy data to evaluate all models. It is shown that our proposed PI-NeuGODE outperforms other models on both data sets. The results show that among all four models, PI-NeuGODE has the lowest MSE and MAE (marked in bold), while IDM has the highest MSE and MAE for the NGSIM data.

Effect of Varying Training Size: To demonstrate the generalizability of our model with limited labeled data, we visualize the performance of PI-NeuGODE and SocialGAN when varying training size of the car-following dataset in Fig. 4. We can see that the results of PI-NeuGODE do not deteriorate as drastically as those of socialGAN as we reduce the training size. This is because PI-NeuGODE uses both labeled and unlabeled instances in training, leading to the learning of more generalizable solutions compared to socialGAN that only uses labeled instances.

[1]www.fhwa.dot.gov/publications/research/operations/07030/index.cfm

Table 1: Evaluation of different models for the platoon modeling problem

Model	Noise-free		Noisy		NGSIM	
	MSE	MAE	MSE	MAE	MSE	MAE
SocialGAN	1.45	1.27	2.45	1.60	3.38	2.12
SocialLSTM	2.56	1.78	4.56	2.39	3.51	2.69
IDM	1.28	1.31	1.89	1.50	4.64	2.93
PI-NeuGODE	**1.23**	**1.16**	**1.34**	**1.18**	**2.62**	**1.46**

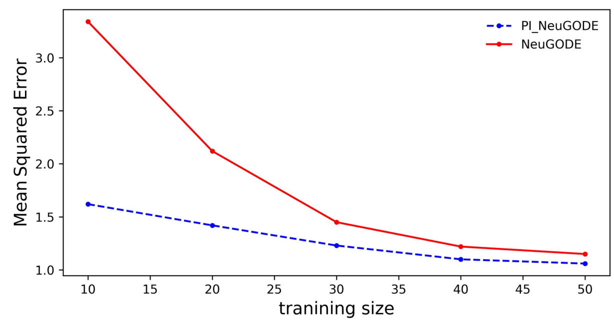


Figure 4: Effect of the training size.

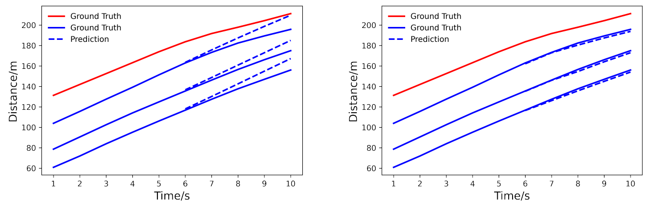
Recovered Physics: For the numerical data, the discovered physics is of the same form of Eq. 13, and the estimated coefficients are: $[v_{\text{des}}, T_0, s_0, a_{\max}, b] = [28.45\ m/s, 0.70\ s, 4.25\ m, 2.14\ m/s^2, 3.24\ m/s^2]$

For the symbolic regression of the NGSIM platooning trajectory, the operator set is $[+, -, *, /, \sqrt{()}, ()^2, ()^3, ()^4]$. To control the complexity of the learned equation, we add a constraint that the power operator, i.e., $()^2, ()^3, ()^4$, can only be used once and cannot be used with each other. The discovered equation is:

$$\begin{aligned} \dot{v}_i(t) = \quad & 0.063(0.042 s_i(t) - 0.072 v_i(t))^2 \\ & -0.041 v_i(t)^2 - 0.083 v_i(t)\Delta v_i(t) + 0.14, \end{aligned} \tag{14}$$

where the variable shares the same meaning as in Eq. 13. For comparison, we apply the symbolic regression to the NGSIM data directly, and the optimal learned equation is a constant $\dot{v}_i(t) = 0.025$. This is because the real-world data is noisy, and exploring high-order operators tends to cause high empirical risk. From this result, we can see that the neural network serves as a filter to smooth the real-world data, allowing subtle patterns to be learned through symbolic regression.

Visualization: Fig. 5a and 5b plot the trajectories for a platoon of vehicles predicted by SocialGAN and PI-NeuGODE, respectively. The red line is the trajectory of the leading car. The solid and dashed blue lines represent the actual and predicted trajectories of the following cars, respectively. It is shown that PI-NeuGODE outperforms SocialGAN when predicting trajectories for the next 5 seconds. The gap between solid blue lines keeps increasing, as the predicted time progresses, as seen in Fig. 5a. This trend is invisible in Fig. 5b. Thus, compared to SocialGAN, PI-NeuGODE has a higher



(a) SocialGAN **(b) PI-NeuGODE**

Figure 5: Predicted trajectories in vehicle platooning

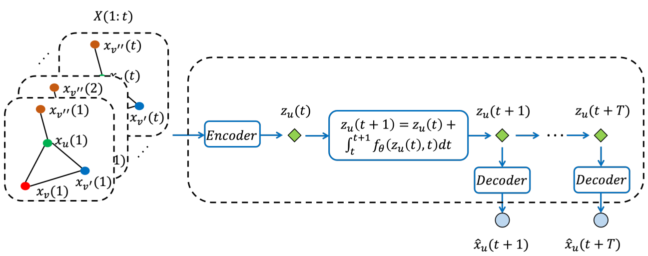


Figure 6: Structure of GODE for pedestrian modeling as a sequence-to-sequence prediction problem.

accuracy when predicting the trajectories of each vehicle in the platoon.

5 EXPERIMENT: PEDESTRIAN TRAJECTORY PREDICTION

In this section, we further apply PI-NeuGODE to the pedestrian trajectory prediction problem, which is an example of the seq2seq problem and is important for a wide range of real-world applications such as crowd management, autonomous navigation, and video surveillance.

5.1 GODE Structure

Different from IVP which only uses the initial state, seq2seq prediction observes a sequence of states, and predicts the future states sequentially. Solving the pedestrian trajectory prediction problem as a seq2seq prediction problem, the structure of neural ODESolver is illustrated in Fig. 6.

5.2 Dataset

Numerical Dataset: The trajectory of the pedestrian is governed by the social force model (SFM) [16] below

$$\frac{d^2 p_i}{dt^2} = F_{iD} + \sum_{\substack{j\in\{1,\dots,N\}\\(i,j)\in E}} F^S_{ji} + \sum_{j\in O} F^O_{ji}, \tag{15}$$

where E and O are the sets of edges and obstacles, respectively; F_{iD}, F^S_{ji} and F^O_{ji} represents the traction force of destination D, the repulsive force of pedestrian and obstacle j on pedestrian i, respectively.

Those three forces can be further depicted as:

$$\begin{aligned} F_{iD} &= m_i \frac{v_{id}\boldsymbol{n}_{iD} - v_i(t)}{\tau} \\ F_{ji} &= \lambda_1 e^{-d_{ji}(t)/\lambda_2} \cdot \boldsymbol{n}_{ji} \\ F_{oi} &= \lambda_3 e^{-d_{oi}(t)/\lambda_4} \cdot \boldsymbol{n}_{oi}, \end{aligned} \tag{16}$$

where $v_{id} = 0.5m/s$ is the desired walking speed. $\boldsymbol{n}_{iD}$ is the unit vector to the target direction. m_i is the mass of pedestrian i, and $\tau = 0.4s$ is the simulation time step. $\lambda_1 =$ and λ_2 are tunable parameters with a physical meaning of force intensity and force radius, respectively. More implementation details are shown in the supplementary material.

Real-world Dataset: We apply our method to the UCY [21] and ETH [29] datasets, which are classic benchmarks for pedestrian trajectory prediction in computer vision research.

5.3 Setting

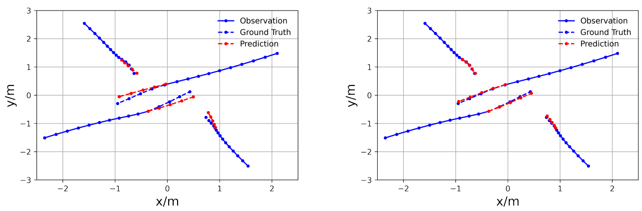
Baselines: In addition to SocialGAN and SocialLSTM, we include the social force model (SFM) as a baseline method in our evaluation. Two attention-based methods, the AgentFormer [42] and the Transformer TF [14], are included for comparison. Moreover, we investigate the effectiveness of Graph Neural ODE (GODE) [31] and the Coupled Graph Neural ODE (CGODE) [18]. To assess the generalizability of our physics-informed framework, we apply the physics-informed technique to CGODE and introduce the physics-informed CGODE (PI-CGODE) as another baseline.

5.4 Results

Table 2: Evaluation of different models for the pedestrian trajectory prediction problem

Model	Numerical Data		ETH (Mean)		UCY (Mean)	
	MSE	MAE	MSE	MAE	MSE	MAE
SFM	0.24	0.42	0.84	1.13	0.74	0.92
SocialGAN	0.38	0.56	0.64	0.86	0.42	0.66
SocialLSTM	0.51	0.73	0.71	0.93	0.51	0.73
Social-STGCNN	0.31	0.54	0.52	0.74	0.41	0. 69
AgentFormer	0.25	0.47	0.42	0.63	0.31	0.49
Transformer TF	0.28	0.51	0.44	0.67	0.35	0.55
GODE	0.26	0.45	0.75	0.85	0.43	0.68
CGODE	0.23	0.42	0.45	0.65	0.34	0.54
PI-CGODE	0.20	0.34	0.36	0.57	0.25	0.43
PI-NeuGODE	**0.16**	**0.32**	**0.31**	**0.46**	**0.22**	**0.40**

Performance Comparison: The experimental results for pedestrian trajectory prediction in Tab. 2 demonstrate the superior performance of our model compared to baselines. We note that the results of CGODE and PI-CGODE are not as good as our proposed model. Upon carefully analyzing the differences between CGODE and our proposed method, we believe that it is due to the different types of graph neural networks used. CGODE employs a GCN-based graph neural network with a binary adjacency matrix to represent the graph structure, which may not be sufficient to capture the



(a) SocialGAN (b) PI-NeuGODE

Figure 7: Visualization of the predicted pedestrian trajectory for the numerical data.

(a) UCY (b) ETH

Figure 8: Visualization of the pedestrian trajectory prediction for the UCY (a) and (b) ETH datasets.

complexity of pedestrian interaction. In contrast, message-passing graph neural networks are able to model the rapid change in the relative distance and direction between two pedestrians.

Visualization: Fig. 7a and 7b plot the trajectories for 4 pedestrians predicted by SocialGAN and PI-NeuGODE, respectively. In each figure, 10 seconds of trajectories are used as observation to predict the next 5 seconds. The ground truth is represented by blue lines while the predictions of both models are depicted by red dashed lines. From the figures, it can be seen that PI-NeuGODE outperforms S-GAN and is closer to the ground truth.

Fig. 8 presents the predicted pedestrian trajectory for the UCY (left) and ETH (right) datasets. The error is aggregated for all scenarios in each dataset. The solid yellow lines, solid green lines, and dashed red lines represent the input trajectories, ground-truth output trajectories, and predicted output trajectories, respectively. The figures show that there is a high level of agreement between the predicted and ground-truth trajectories in both datasets.

Computational Time: The computational time required for the proposed model, along with the baseline models, is presented in Tab 3. SocialGAN, SocialLSTM, and GODE are selected for their computational efficiency, making them suitable benchmarks for comparison. As can be observed, PI-NeuGODE has a similar computation cost as GODE and SocialGAN. SocialLSTM achieves a faster computation compared to other models, but this fast computation is at a cost of model accuracy as indicated in Tab. 2. The results indicate that the computational effort required for our model is negligible, thus confirming that it is not a limiting factor.

Table 3: The computation time for the trajectory prediction

Model	Computation Time	
	Training (s/epoch)	Inference (s)
PI-NeuGODE	28.5	0.26
GODE	23.6	0.23
SocialGAN	20.2	0.18
SocialLSTM	12.6	0.14

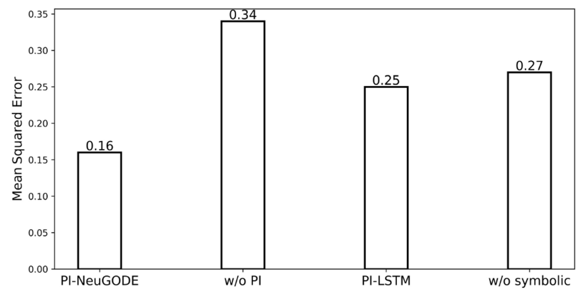


Figure 9: Ablation study.

Ablation study: In order to evaluate the contributions of each component in our proposed model, we conducted an ablation study as shown in Fig. 9. PI-LSTM" refers to applying PIDL to an LSTM model. "w/o PI" and "w/o symbolic regression" are the variants of our proposed PI-NeuGODE without using PIDL and symbolic regression, respectively. The experiment setting is the same as in the numerical trajectory prediction task. The results clearly show that the integration of the physics-informed technique significantly improves the performance of the model. However, when combining PIDL and GODE without incorporating symbolic regression, we observed no noticeable improvement in performance. Specifically, the outcomes indicate that the combined model achieves performance levels similar to that of the PI-LSTM, suggesting that a mere combination of these two techniques does not offer additional advantages. This finding underscores the indispensability of symbolic regression as an essential component within the model architecture.

6 CONCLUSION

We develop a first-of-its-kind methodological framework integrating PIDL, graph neural ODEs, and symbolic regression, for the prediction of interacting spatiotemporal trajectories. The integrated framework can not only make long-term prediction using historical data, but also leverage physics knowledge to learn individuals' dynamic decision-making processes with symbolic regression. The effectiveness of the proposed model has been validated on two tasks, human driving and platooning together with crowding. This work can be extended by modeling stochasticity and enhancing the model generalization using transfer learning.

ACKNOWLEDGMENTS

This work is sponsored by the National Science Foundation (NSF) under NSF SCC-2218809.

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] Mark Alber, Adrian Buganza Tepole, William R Cannon, Suvranu De, Salvador Dura-Bernal, Krishna Garikipati, George Karniadakis, William W Lytton, Paris Perdikaris, Linda Petzold, and Ellen Kuhl. 2019. Integrating machine learning and multiscale modeling—perspectives, challenges, and opportunities in the biological, biomedical, and behavioral sciences. *npj Digital Medicine* 2, 1 (2019), 1–11.

[3] Bahador Bahmani, Hyoung Suk Suh, and WaiChing Sun. 2023. Discovering interpretable elastoplasticity models via the neural polynomial method enabled symbolic regressions. *arXiv preprint arXiv:2307.13149* (2023).

[4] Bahador Bahmani and WaiChing Sun. 2023. Physics-constrained symbolic model discovery for polyconvex incompressible hyperelastic materials. *arXiv preprint arXiv:2310.04286* (2023).

[5] Peter Battaglia, Razvan Pascanu, Matthew Lai, Danilo Jimenez Rezende, et al. 2016. Interaction networks for learning about objects, relations and physics. *Advances in neural information processing systems* 29 (2016).

[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] Jeongwhan Choi, Hwangyong Choi, Jeehyun Hwang, and Noseong Park. 2022. Graph neural controlled differential equations for traffic forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 6367–6374.

[8] Miles Cranmer, Alvaro Sanchez Gonzalez, Peter Battaglia, Rui Xu, Kyle Cranmer, David Spergel, and Shirley Ho. 2020. Discovering symbolic models from deep learning with inductive biases. *Advances in Neural Information Processing Systems* 33 (2020), 17429–17442.

[9] Salvatore Cuomo, Vincenzo Schiano Di Cola, Fabio Giampaolo, Gianluigi Rozza, Maizar Raissi, and Francesco Piccialli. 2022. Scientific Machine Learning through Physics-Informed Neural Networks: Where we are and What's next. *arXiv preprint arXiv:2201.05624* (2022).

[10] Xuan Di and Rongye Shi. 2021. A survey on autonomous vehicle control in the era of mixed-autonomy: From physics-based to AI-guided driving policy learning. *Transportation Research Part C: Emerging Technologies* 125 (2021), 103008.

[11] Xuan Di, Rongye Shi, Zhaobin Mo, and Yongjie Fu. 2023. Physics-Informed Deep Learning For Traffic State Estimation: A Survey and the Outlook. *Algorithms* 16, 6 (2023), 305.

[12] Hao Du, Supeng Leng, Jianhua He, and Longyu Zhou. 2021. Digital twin based trajectory prediction for platoons of connected intelligent vehicles. In *2021 IEEE 29th International Conference on Network Protocols (ICNP)*. IEEE, 1–6.

[13] Longsen Gao, Giovanni Cordova, Claus Danielson, and Rafael Fierro. 2023. Autonomous Multi-Robot Servicing for Spacecraft Operation Extension. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 10729–10735.

[14] Francesco Giuliari, Irtiza Hasan, Marco Cristani, and Fabio Galasso. 2021. Transformer networks for trajectory forecasting. In *2020 25th international conference on pattern recognition (ICPR)*. IEEE, 10335–10342.

[15] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. 2018. Social gan: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2255–2264.

[16] Dirk Helbing and Peter Molnar. 1995. Social force model for pedestrian dynamics. *Physical review E* 51, 5 (1995), 4282.

[17] Archie J Huang and Shaurya Agarwal. 2020. Physics informed deep learning for traffic state estimation. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 1–6.

[18] Zijie Huang, Yizhou Sun, and Wei Wang. 2021. Coupled graph ode for learning interacting system dynamics. In *The 27th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)*.

[19] Varun Jain, Stephan Lapoehn, Tobias Frankiewicz, Tobias Hesse, Mohamed Gharba, Sandip Gangakhedkar, Karthikeyan Ganesan, Hanwen Cao, Josef Eichinger, Ali Ramadan Ali, et al. 2017. Prediction based framework for vehicle platooning using vehicular communications. In *2017 IEEE Vehicular Networking Conference (VNC)*. IEEE, 159–166.

[20] George Em Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. 2021. Physics-informed machine learning. *Nature Reviews Physics* 3, 6 (2021), 422–440.

[21] Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. 2007. Crowds by example. In *Computer graphics forum*, Vol. 26. Wiley Online Library, 655–664.

[22] Xiaoling Luo, Xiaobo Ma, Matthew Munden, Yao-Jan Wu, and Yangsheng Jiang. 2022. A multisource data approach for estimating vehicle queue length at metered on-ramps. *Journal of Transportation Engineering, Part A: Systems* 148, 2 (2022), 04021117.

[23] Xiaobo Ma, Abolfazl Karimpour, and Yao-Jan Wu. 2020. Statistical evaluation of data requirement for ramp metering performance assessment. *Transportation Research Part A: Policy and Practice* 141 (2020), 248–261.

[24] Zhaobin Mo and Xuan Di. 2022. Uncertainty quantification of car-following behaviors: physics-informed generative adversarial networks. In *the 28th ACM SIGKDD in conjunction with the 11th International Workshop on Urban Computing (UrbComp2022)*.

[25] Zhaobin Mo, Yongjie Fu, Daran Xu, and Xuan Di. 2022. Trafficflowgan: Physics-informed flow based generative adversarial network for uncertainty quantification. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 323–339.

[26] Zhaobin Mo, Wangzhi Li, Yongjie Fu, Kangrui Ruan, and Xuan Di. 2022. CVLight: Decentralized learning for adaptive traffic signal control with connected vehicles. *Transportation research part C: emerging technologies* 141 (2022), 103728.

[27] Zhaobin Mo, Rongye Shi, and Xuan Di. 2021. A physics-informed deep learning paradigm for car-following models. *Transportation research part C: emerging technologies* 130 (2021), 103240.

[28] Jared O'Leary, Joel A Paulson, and Ali Mesbah. 2022. Stochastic physics-informed neural ordinary differential equations. *J. Comput. Phys.* 468 (2022), 111466.

[29] Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc Van Gool. 2009. You'll never walk alone: Modeling social behavior for multi-target tracking. In *2009 IEEE 12th international conference on computer vision*. IEEE, 261–268.

[30] Brenden K Petersen. 2019. Deep symbolic regression: Recovering mathematical expressions from data via policy gradients. *CoRR* (2019).

[31] Michael Poli, Stefano Massaroli, Junyoung Park, Atsushi Yamashita, Hajime Asama, and Jinkyoo Park. 2019. Graph neural ordinary differential equations. *arXiv preprint arXiv:1911.07532* (2019).

[32] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. 2019. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* 378 (2019), 686–707.

[33] Chengping Rao, Pu Ren, Yang Liu, and Hao Sun. 2022. Discovering nonlinear PDEs from scarce data with physics-encoded learning. *arXiv preprint arXiv:2201.12354* (2022).

[34] Michael Schmidt and Hod Lipson. 2009. Distilling free-form natural laws from experimental data. *science* 324, 5923 (2009), 81–85.

[35] Rongye Shi, Zhaobin Mo, and Xuan Di. May, 2021. Physics informed deep learning for traffic state estimation: A hybrid paradigm informed by second-order traffic models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 540–547.

[36] Rongye Shi, Zhaobin Mo, Kuang Huang, Xuan Di, and Qiang Du. Aug., 2022. A physics-informed deep learning paradigm for traffic state and fundamental diagram estimation. *IEEE Transactions on Intelligent Transportation Systems* 23, 8 (Aug., 2022), 11688–11698.

[37] Rongye Shi, Zhaobin Mo, Kuang Huang, Xuan Di, and Qiang Du. Jan. 17, 2021. Physics-informed deep learning for traffic state estimation. *arXiv preprint arXiv:2101.06580* (Jan. 17, 2021).

[38] Naoya Takeishi and Alexandros Kalousis. 2021. Physics-Integrated Variational Autoencoders for Robust and Interpretable Generative Modeling. *Advances in Neural Information Processing Systems* 34 (2021), 14809–14821.

[39] Martin Treiber, Ansgar Hennecke, and Dirk Helbing. 2000. Congested traffic states in empirical observations and microscopic simulations. *Physical review E* 62, 2 (2000), 1805.

[40] Silviu-Marian Udrescu and Max Tegmark. 2020. AI Feynman: A physics-inspired method for symbolic regression. *Science Advances* 6, 16 (2020), eaay2631.

[41] Jur Van den Berg, Ming Lin, and Dinesh Manocha. 2008. Reciprocal velocity obstacles for real-time multi-agent navigation. In *2008 IEEE international conference on robotics and automation*. Ieee, 1928–1935.

[42] Ye Yuan, Xinshuo Weng, Yanglan Ou, and Kris M Kitani. 2021. Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 9813–9823.

[43] Jiangbei Yue, Dinesh Manocha, and He Wang. 2022. Human trajectory prediction via neural social physics. In *European Conference on Computer Vision*. Springer, 376–394.

[44] Guozhen Zhang, Zihan Yu, Depeng Jin, and Yong Li. 2022. Physics-infused Machine Learning for Crowd Simulation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2439–2449.

[45] Juntang Zhuang, Nicha Dvornek, Xiaoxiao Li, and James S Duncan. 2020. Ordinary differential equations on graph networks. (2020).