Physics-Informed Deep Learning for Traffic State Estimation: A Hybrid Paradigm Informed By Second-Order Traffic Models

Rongye Shi, Zhaobin Mo, Xuan Di
Columbia University, New York, NY, USA
rongyes@alumni.cmu.edu, zm2302@columbia.edu, sharon.di@columbia.edu

Abstract

Traffic state estimation (TSE) reconstructs the traffic variables (e.g., density or average velocity) on road segments using partially observed data, which is important for traffic managements. Traditional TSE approaches mainly bifurcate into two categories: model-driven and data-driven, and each of them has shortcomings. To mitigate these limitations, hybrid TSE methods, which combine both model-driven and data-driven, are becoming a promising solution. This paper introduces a hybrid framework, physics-informed deep learning (PIDL), to combine second-order traffic flow models and neural networks to solve the TSE problem. PIDL can encode traffic flow models into deep neural networks to regularize the learning process to achieve improved data efficiency and estimation accuracy. We focus on highway TSE with observed data from loop detectors and probe vehicles, using both density and average velocity as the traffic variables. With numerical examples, we show the use of PIDL to solve a popular second-order traffic flow model, i.e., a Greenshields-based Aw-Rascle-Zhang (ARZ) model, and discover the model parameters. We then evaluate the PIDL-based TSE method using the Next Generation SIMulation (NGSIM) dataset. Experimental results demonstrate the proposed PIDL-based approach to outperform advanced baseline methods in terms of data efficiency and estimation accuracy.

Introduction

Efficient traffic control, operation, ramp metering and many other transportation management strategies are important building blocks of Internet-of-Things (IoT), which tackle urban traffic challenges leveraging emerging sensing, communication, and computational technologies, as well as data science. A prerequisite step is to estimate traffic states (e.g., density, velocity, and flow) using observed data from traffic sensors (e.g., inductive loop detectors and probe vehicles). However, the estimation is nontrivial in practice because the information from traffic sensors is limited in most cases, making the traffic state estimation (TSE) problem a long-standing research topic over decades. Formally, TSE refers to the data mining problem of reconstructing traffic state variables, including but not limited to flow (veh/h), density (veh/km), and velocity (km/h), on road segments using partially observed data from traffic sensors (Seo et al. 2017).

Most traditional TSE methods are model-driven, i.e., to build up a traffic state estimator on top of a priori knowledge of traffic dynamics described by first-order principles. These models were proposed based on ideal assumptions and conditions that may fail to fully capture the phenomena of real-world traffic. An alternative is the data-driven method, which can conduct TSE without any explicit traffic models nor theoretical assumptions. Data-driven methods learn the estimation rule directly from observed data using statistical or machine learning (ML) methods. However, these methods require a large quantity of high-quality data to perform a satisfactory estimation, implying high demands on the constructions of data collection infrastructure and corresponding maintenance costs.

Human drivers exhibit highly unstable and nonlinear driving behaviors, leading to stop-and-go traffic waves and traffic congestions at an aggregate level. Accordingly, neither model-driven nor data-driven methods alone suffice to estimate such behavioral complexity with an allowable accuracy. Therefore, hybrid methods, which combine both model-driven and data-driven, are becoming a promising direction for IoT-related applications, and this paper introduces a hybrid framework, physics-informed deep learning (PIDL), to the TSE problem. The PIDL framework, which is originally proposed by Raissi for solving nonlinear partial differential equations (PDEs) (Raissi 2018; Raissi and Karniadakis 2018), consists of a model-driven component (a physics-informed neural network for regularization) and a data-driven component (a physics-uninformed neural network for estimation), making it a natural fit to the research goal of TSE.

Traditional methods of TSE highly rely on fixed-location sensors such as inductive loop detectors and traffic cameras. With the emergence of advanced sensing and communication technologies, such as connected vehicles equipped with GPS and vehicle-to-vehicle and vehicle-to-infrastructure connectivity modules, trajectories of probe vehicles have become important sources for tempo-spatial traffic information. This paper will not only take the advantage of loop detector data, but also trajectory data of probe vehicles for improved estimation accuracy.

Specifically, the main contributions of this paper are as follows:

- We will leverage tempo-spatial traffic information col-

lected from probe vehicles' trajectories, as well as aggregate traffic density and velocity collected from conventional inductive loop detectors, to improve TSE accuracy;

- We design a PIDL architecture for TSE with the traffic dynamics described by a second-order traffic flow model, i.e., a Greenshields-based Aw-Rascle-Zhang (ARZ) model. We leverage the proposed PIDL-based method to estimate the traffic density and average velocity at the same time using limited observed data from loop detectors or/and probe vehicles. We select the second-order traffic flow model because of its flexibility of accommodating both traffic density and velocity measurements, as well as its mathematical power in capturing the stop-and-go traffic phenomenon. Compared to first-order models, a second-order model contains more complicated non-linear differential equations, and whether it can successfully fit into PIDL for TSE will be comprehensively studied in the paper;
- We further show the ability of PIDL for conducting both model parameter discovery and TSE at the same time, to tackle the practical challenge, i.e., the parameters of a traffic flow model are usually unknown and offline calibration could be expensive and inaccurate;
- We demonstrate the advantages of PIDL using a real dataset of a highway scenario, i.e., the Next Generation SIMulation (NGSIM) dataset. Advanced baseline TSE methods, such as pure neural networks, extended Kalman filter, PIDL with a first-order traffic model, are compared with the ARZ-based PIDL method. In addition, we study and compare the estimation performance of different kinds of sensor configurations, which is of great interest in TSE practice.

Related Work

A critical step of model-driven TSE methods is to design and extend a macroscopic traffic flow model that describes the real-world dynamics as comprehensively as possible. These models can be briefly divided into two classes: first-order and second-order. The LWR model (Lighthill and Whitham 1955; Richards 1956) is the most classical first-order traffic flow model, which can reproduce simple aggregate traffic behaviors, such as the propagation and dissipation of traffic jams known as shockwaves. However, LWR assumes density-velocity equilibrium and fails to model more complicated phenomena, such as stop and go traffic (Flynn et al. 2009). To mitigate this issue, second-order models were developed, including the Payne-Whitham (PW) model (Payne 1971; Whitham 1974) and the ARZ model (Aw and Rascle 2000; Zhang 2002). The ARZ model has more accurate description of the information flow in the traffic dynamics compared to the PW model. This paper thus considers the ARZ model for the TSE research.

Given a traffic flow model, model-driven TSE methods apply data assimilation to conduct estimation by solving model equations while allowing observed data to correct the estimation. Examples include the Kalman filter and its variants (Wang, Papageorgiou, and Messmer 2008; Wang et al. 2009; Di, Liu, and Davis 2010). Most of them assume fixed-location sensors and may not apply to mobile sensors, such as probe vehicles. Data-driven TSE methods, in contrast, make an estimation from data directly using ML and statistical approaches, such as neural networks (Zheng et al. 2019). Most of them do not assume data types, but have high demands on data volumes (i.e., low data efficiency). The research on hybrid TSE methods is lacking, and this paper explores a hybrid TSE using the PIDL framework, which is originally proposed to leverage the deep learning paradigm for solving nonlinear PDEs (Raissi 2018; Raissi and Karniadakis 2018).

PIDL Framework for TSE Using Second-Order Traffic Flow Models

This paper focuses on solving the following TSE problem: *to estimate the spatio-temporal traffic density ρ and average velocity u of a highway segment over a period of time using observations from a limited number of loop detectors or probe vehicles*. To better describe the traffic dynamics, this paper considers second-order models using density and velocity as traffic state variables. This section defines the PIDL TSE framework at a high-level. We will introduce how to use PIDL and limited observed traffic sensor data to (1) estimate traffic dynamics which are described by traffic flow models and (2) discover unknown model parameters.

General PIDL Framework for TSE

Let $\mathcal{N}_1[\cdot]$ and $\mathcal{N}_2[\cdot]$ be two general nonlinear differential operators and Ω be a subset of $\mathbb{R}^d$. For one-dimentional TSE, d is one by default in this paper, i.e., $\Omega = [0, L], L \in \mathbb{R}^+$. Then, the problem is to find the traffic density $\rho(t, x)$ and average velocity $u(t, x)$ at each point (t, x) in a continuous domain, such that the following PDEs of a traffic flow model can be satisfied:

$$\begin{array}{l} \rho_t(t,x) + \mathcal{N}_1[\rho(t,x), u(t,x)] = 0 \\ u_t(t,x) + \mathcal{N}_2[\rho(t,x), u(t,x)] = 0 \end{array}, \tag{1}$$

where $x \in \Omega, t \in [0, T]$ and $T \in \mathbb{R}^+$. This setting implies that the spatio-temporal domain D of interest is a continuous set of points: $D = \{(t, x)|t \in [0, T], x \in [0, L]\}$. We use discrete grid points G evenly deployed over D to represent the domain: $G = \{(t^{(r)}, x^{(r)})|r = 1, .., N_g\}$. The total number N_g of grid points controls the fine-grained level of G as a representation of D.

PIDL approximates both $\rho(t, x)$ and $u(t, x)$ using a neural network with time t and location x as its inputs. This neural network is called *physics-uninformed neural network* (PUNN) (or estimation network in the TSE setting), which is parameterized by θ. We denote the approximation from PUNN as $\hat{\rho}(t, x; \theta)$ and $\hat{u}(t, x; \theta)$. To use physics to guide the training of PUNN, the following residuals are used:

$$\begin{array}{l} \hat{f}_1(t,x;\theta) := \hat{\rho}_t(t,x;\theta) + \mathcal{N}_1[\hat{\rho}(t,x;\theta), \hat{u}(t,x;\theta)] \\ \hat{f}_2(t,x;\theta) := \hat{u}_t(t,x;\theta) + \mathcal{N}_2[\hat{\rho}(t,x;\theta), \hat{u}(t,x;\theta)] \end{array}, \tag{2}$$

which is defined according to the traffic flow model in Eqs.(1). The calculation of both residuals $\hat{f}_1(t,x;\theta)$ and $\hat{f}_2(t,x;\theta)$ is done by a *physics-informed neural network* (PINN), directly using $\hat{\rho}(t,x;\theta)$ and $\hat{u}(t,x;\theta)$, the output of PUNN, as its inputs. When estimations are closer to the true $\rho(t,x)$ and $u(t,x)$, both residuals will be closer to zero. PINN introduces no new parameters, and thus, the mapping from (t,x) to $\hat{f}_1$ and $\hat{f}_2$ is parameterized by θ.

In implementation, PINN calculates the residuals using the function packages, such as `tf.gradient`, in Tensorflow. We can customize the activation functions and connecting structure of neurons in PINN to conduct the differential operation in Eqs.(2). It needs to be noted that once the traffic model is known, the connecting weights in PINN are fixed and no new learning parameters are introduced.

The training data for PIDL consist of (1) *observation points* $O = \{(t_o^{(i)}, x_o^{(i)}) | i = 1, ..., N_o\}$, (2) *target values* $P = \{(\rho^{(i)}, u^{(i)}) | i = 1, ..., N_o\}$ (i.e., the true traffic states at the observation points), and (3) *auxiliary points* $A = \{(t_a^{(j)}, x_a^{(j)}) | j = 1, ..., N_a\}$. We use i and j to index observation points and auxiliary points, respectively. One target value is associated with one observation point, and thus, O and P have the same indexing system i. This paper uses the term, observed data, to denote $\{O, P\}$. Both O and A are subsets of grid points G (i.e., $O \in G$ and $A \in G$). It is allowed to have some overlaps between O and A.

Observation points O are limited to the time and locations that traffic sensors can visit and record. In contrast, auxiliary points A have neither measurement requirements nor location limitations, and the number of A is controllable. A are used for regularization purposes, and this is why they are called "auxiliary". To train a PUNN for TSE, the following loss is used:

$$\begin{aligned} &Loss_\theta = MSE_o + MSE_a \\ &= \underbrace{\tfrac{1}{N_o}\sum_{i=1}^{N_o} \alpha_1 |\hat{\rho}(t_o^{(i)}, x_o^{(i)}; \theta) - \rho^{(i)}|^2 + \alpha_2 |\hat{u}(t_o^{(i)}, x_o^{(i)}; \theta) - u^{(i)}|^2}_{data\ discrepancy} \\ &\underbrace{+\tfrac{1}{N_a}\sum_{j=1}^{N_a} \beta_1 |\hat{f}_1(t_a^{(j)}, x_a^{(j)}; \theta)|^2 + \beta_2 |\hat{f}_2(t_a^{(j)}, x_a^{(j)}; \theta)|^2}_{physical\ discrepancy}, \end{aligned} \tag{3}$$

where α_1, α_2, β_1 and β_2 are hyperparameters for balancing how each component in data discrepancy and physical discrepancy influences the loss. The data discrepancy involves the mean square errors (MSEs) between approximations $(\hat{\rho}, \hat{u})$ on O and target values P. The physical discrepancy involves the MSEs between residual values $(\hat{f}_1, \hat{f}_2)$ and zero, which quantify how much an approximation is inconsistent to the traffic model, i.e., the loss is regularized by PINN via physical discrepancy. Given the training data, we apply neural network training algorithms, e.g., a backpropagation method, to solve $\theta^* = \text{argmin}_\theta \, Loss_\theta$. The PUNN parameterized by θ^* can then be used to estimate the density and average velocity at any point of G, as well as the whole domain D, with good physical consistency.

PIDL for TSE and Model Parameter Discovery

Sometimes, there exist unknown parameters in a given PDE traffic flow model, and model parameter discovery needs to be addressed in addition to estimation. The proposed PIDL is able to handle this case as well. Let $\mathcal{N}_1[\cdot;\lambda_1]$ and $\mathcal{N}_2[\cdot;\lambda_2]$ be two general nonlinear differential operators parameterized by unknown parameters λ_1 and λ_2, respectively. We merge both λ_1 and λ_2 into one vector $\boldsymbol{\lambda} = (\lambda_1, \lambda_2)$ for simplification. Then, the traffic flow model becomes:

$$\begin{aligned} &\rho_t(t,x) + \mathcal{N}_1[\rho(t,x), u(t,x); \boldsymbol{\lambda}] = 0 \\ &u_t(t,x) + \mathcal{N}_2[\rho(t,x), u(t,x); \boldsymbol{\lambda}] = 0 \end{aligned}, \tag{4}$$

and the goal is to find the parameters $\boldsymbol{\lambda}$ that best describe the observed data, and at the same time, approximate the traffic states ρ and u consistent to Eqs.(4). For this problem, the residuals are redefined as

$$\begin{aligned} \hat{f}_1(t,x;\theta,\boldsymbol{\lambda}) &:= \hat{\rho}_t(t,x;\theta) + \mathcal{N}_1[\hat{\rho}(t,x;\theta), \hat{u}(t,x;\theta); \boldsymbol{\lambda}] \\ \hat{f}_2(t,x;\theta,\boldsymbol{\lambda}) &:= \hat{u}_t(t,x;\theta) + \mathcal{N}_2[\hat{\rho}(t,x;\theta), \hat{u}(t,x;\theta); \boldsymbol{\lambda}] \end{aligned}. \tag{5}$$

The PINN, by which the residuals in Eqs.(5) are calculated, is related to both θ and $\boldsymbol{\lambda}$. Accordingly, the loss for both parameter discovering and TSE is defined as follows:

$$\begin{aligned} &Loss_{\theta,\boldsymbol{\lambda}} = MSE_o + MSE_a \\ &= \tfrac{1}{N_o}\sum_{i=1}^{N_o} \alpha_1 |\hat{\rho}(t_o^{(i)}, x_o^{(i)}; \theta) - \rho^{(i)}|^2 + \alpha_2 |\hat{u}(t_o^{(i)}, x_o^{(i)}; \theta) - u^{(i)}|^2 \\ &+ \tfrac{1}{N_a}\sum_{j=1}^{N_a} \beta_1 |\hat{f}_1(t_a^{(j)}, x_a^{(j)}; \theta, \boldsymbol{\lambda})|^2 + \beta_2 |\hat{f}_2(t_a^{(j)}, x_a^{(j)}; \theta, \boldsymbol{\lambda})|^2. \end{aligned} \tag{6}$$

Given the training data, we apply the backpropagation algorithm to solve $(\theta^*, \boldsymbol{\lambda}^*) = \text{argmin}_{\theta,\boldsymbol{\lambda}} \, Loss_{\theta,\boldsymbol{\lambda}}$, and the $\boldsymbol{\lambda}^*$-parameterized traffic flow model of Eqs.(4) is the most likely physics that generates the observed data and θ^*-parameterized PUNN can then be used to find physically consistent traffic states over the domain.

Numerical Example on PIDL for ARZ

We present a numerical example to show the ability of PIDL to estimate the traffic dynamics governed by a Greenshields-based ARZ model. Define flow rate Q to be the number of vehicles passing a specific position on the road per unit time, traffic density ρ to be the average number of vehicles per unit length, and u to be the average velocity of a specific position on the road, we can deduce $Q = \rho u$. Both ρ and u are the basic traffic state variables to estimate.

The ARZ model (Aw and Rascle 2000; Zhang 2002) involves both a conservation law of vehicles and a momentum equation on velocity. The former can be written as $\rho_t + (\rho u)_x = 0$, which defines that the vehicle volume change of a specific position on the road equates the difference between inflow and outflow vehicles. The latter analogizes the fluid dynamics by introducing a traffic "pressure" $h(\rho)$. Then, $(u + h(\rho))$ can be interpreted as the "desired" speed reached by vehicles on an empty road, which is *advected* with the velocity field. This advected property is formalized as $(u + h(\rho))_t + u(u + h(\rho))_x = 0$. To properly

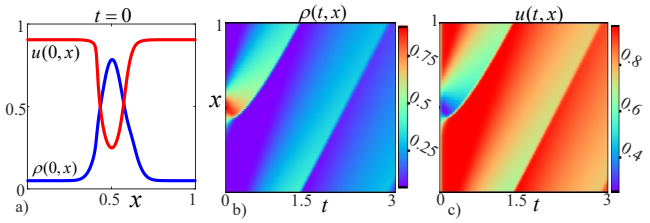

Figure 1: a) is the bell-shaped initial ρ and u over $x \in [0, 1]$; b) and c) are numerical solutions for ρ and u, respectively.

constrain the desired speed, a relaxation term of the form $(U_{eq}(\rho) - u)/\tau$ is added to the right-hand side, implying that drivers tend to adjust their actual velocity to the desired velocity $U_{eq}(\rho)$ with a relaxation time scale τ. $U_{eq}(\rho)$ can be defined separately, and Greenshields function (Greenshields, Channing, and Miller 1935) is a basic and popular choice, which is defined as $U_{eq}(\rho) = u_{max}(1 - \rho/\rho_{max})$, where u_{max} and ρ_{max} are maximum velocity and maximum (jam) density, respectively. This function has a linear form with two coefficients u_{max} and ρ_{max}, which are usually fitted with data.

In this section, we study the Greenshields-based ARZ traffic flow model of a "ring road" in $t \in [0, 3]$, $x \in [0, 1]$:

$$\begin{aligned} &\rho_t + (\rho u)_x = 0, \\ &(u + h(\rho))_t + u(u + h(\rho))_x = (U_{eq}(\rho) - u)/\tau, \\ &h(\rho) = U_{eq}(0) - U_{eq}(\rho) \quad (traffic\ pressure), \\ &U_{eq}(\rho) = u_{max}(1 - \rho/\rho_{max}) \quad (desired\ speed), \\ &\rho(t, 0) = \rho(t, 1), u(t, 0) = u(t, 1)\ (bdry.\ cond.), \end{aligned} \tag{7}$$

where we set the parameters irregularly as $\rho_{max} = 1.13$, $u_{max} = 1.02$, and $\tau = 0.02$. ρ_{max} and u_{max} are usually determined by physical restrictions of the road and vehicles. We select irregular parameters in this experiment to better justify the PIDL's capacity in model discovery.

Given the bell-shaped initial of ρ and u as shown in Fig.1.a, we apply the Lax–Friedrichs (LF) scheme (LeVeque 2002) to solve Eqs.(7) on grid G with 960 (time)×240(space) points evenly deployed over the $[0, 3] \times [0, 1]$ domain. Thus, the total number of points in G is N_g=960×240. The LF numerical solutions of both ρ and u over the domain are shown in Fig.1 as well.

From the figure, we can observe the phenomenon of traffic shockwave, i.e., the peak of density propagates along the direction of x. Because this is a ring road, a boundary condition (bdry. cond.) must impose that the dynamics reaching x=1 continues at x=0. We treat this numerical solution as the ground-truth to test our PIDL-based approach.

PIDL Architecture Design for ARZ

Following the concept of Eqs.(2), we define the residuals:

$$\begin{aligned} \hat{f}_1(t, x; \theta) &:= \hat{\rho}_t + (\hat{\rho}\hat{u})_x \\ \hat{f}_2(t, x; \theta) &:= (\hat{u} + h(\hat{\rho}))_t + \hat{u}(\hat{u} + h(\hat{\rho}))_x \\ &\quad -(U_{eq}(\hat{\rho}) - \hat{u})/\tau \end{aligned}, \tag{8}$$

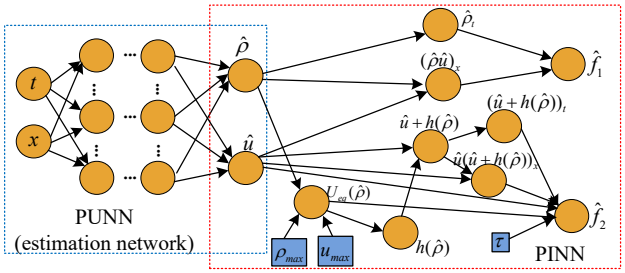

Figure 2: PIDL architecture for Greenshields-based ARZ in Eqs.(7). Model parameters are held by variable nodes (blue rectangular nodes).

where $\hat{\rho}$ and $\hat{u}$ are shorthands for $\hat{\rho}(t, x; \theta)$ and $\hat{u}(t, x; \theta)$, respectively outputted from a PUNN.

Given the definition of $(\hat{f}_1, \hat{f}_2)$, the PINN that encodes the Greenshields-based ARZ model is shown in Fig.2. This architecture contains a PUNN for traffic state estimation, followed by a PINN for calculating the residuals in Eqs.(8). The PUNN parameterized by θ is designed as a fully-connected feedforward neural network with 8 layers and the hidden nodes of each layer range between 20 and 60. The *tanh* is used as the activation function of each hidden neuron in PUNN. In contrast, the connecting weights in PINN are one by default and specific operations for intermediate values of Eqs.(8) are calculated by PINN nodes via customized activation functions. For experiments of estimation without parameter discovery, constant values ($\rho_{max} = 1.13$, $u_{max} = 1.02$, and $\tau = 0.02$) are fixed in variable nodes.

To customize the training of PIDL to Eqs. (7), in addition to the training data O, P and A defined in the previous section, we introduce *boundary auxiliary points* $B = \{(t_b^{(k)}, 0)|k = 1, ..., N_b\} \cup \{(t_b^{(k)}, 1)|k = 1, ..., N_b\}$, for learning the boundary condition in Eqs. (7). The time instances $t^{(k)}$ are randomly sampled from $[0, 3]$. For this experiment, we use the following learning loss:

$$\begin{aligned} &Loss_\theta = MSE_o + MSE_a + MSE_b \\ &= \tfrac{1}{N_o} \sum_{i=1}^{N_o} \alpha_1 |\hat{\rho}(t_o^{(i)}, x_o^{(i)}; \theta) - \rho^{(i)}|^2 + \alpha_2 |\hat{u}(t_o^{(i)}, x_o^{(i)}; \theta) - u^{(i)}|^2 \\ &\quad + \tfrac{1}{N_a} \sum_{j=1}^{N_a} \beta_1 |\hat{f}_1(t_a^{(j)}, x_a^{(j)}; \theta)|^2 + \beta_2 |\hat{f}_2(t_a^{(j)}, x_a^{(j)}; \theta)|^2 \\ &\quad + \tfrac{1}{N_b} \sum_{k=1}^{N_b} (\gamma_1 |\hat{\rho}(t_b^{(k)}, 0; \theta) - \hat{\rho}(t_b^{(k)}, 1; \theta)|^2 \\ &\quad + \gamma_2 |\hat{u}(t_b^{(k)}, 0; \theta) - \hat{u}(t_b^{(k)}, 1; \theta)|^2) \end{aligned}, \tag{9}$$

where MSE_b consists of two error components weighted by hyperparameters (γ_1, γ_2), quantifying the discrepancy of $\hat{\rho}$ and $\hat{u}$ from the boundary condition.

TSE Using Data from Loop Detectors and Probe Vehicles We apply PIDL to the TSE problem using observations from two kinds of traffic sensors: loop detectors and probe vehicles. A loop detector on a road is able to count the vehicles passing a location, and a pair of loop detectors at a regular interval can be used to measure traffic density. However, this

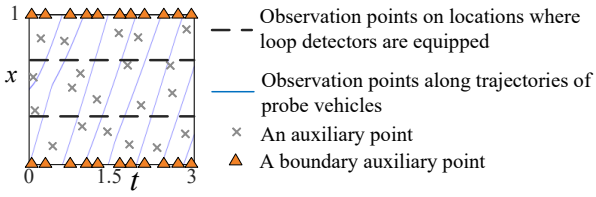

Figure 3: A sparse presentation of how observations points, and auxiliary points are deployed.

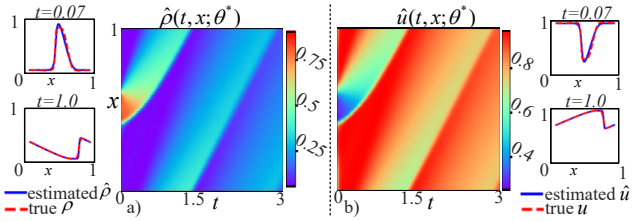

Figure 4: a) Estimation of the density and comparisons of estimated and true values at certain time points. (b) Results for average velocity.

lp	$Err(\hat{\rho},\rho)$	$Err(\hat{u},u)$	(lp,tj)	$Err(\hat{\rho},\rho)$	$Err(\hat{u},u)$
3	0.0682	0.0284	(1,1)	0.0829	0.0256
4	0.0437	0.0201	(2,1)	0.0595	0.0282
5	0.0382	0.0110	(1,2)	0.0542	0.0217

Table 1: Results for detector-only and mixed configs. " lp" and "tj" are the number of loops and trajectories, respectively.

sensor is not good at measuring average speed because only point speed measurements are given. A probe vehicle frequently records the velocity on its trajectory along the flow, providing a better velocity measurement. However, this sensor cannot provide density due to vehicles' limitations to comprehend surroundings. Because our PIDL requires observations on both ρ and u, this section considers two types of sensing configurations: 1) a loop detector-only configuration and each detector can record both ρ and u, and 2) a mixed sensing configuration, where ρ and u are separately measured by loop detectors and probe vehicles, respectively.

As shown in Fig.3, observation points O are supported by sensors. Loop detectors are equipped evenly on the road and traffic states at these certain locations can be observed at all time (the figure illustrates an example where two detectors are used). Probe vehicles collect traffic velocities along their trajectories, and they are assumed to initialize evenly on the road at $t = 0$ (the figure illustrates two probe vehicles). Each observation point is associated with target values ρ or u, constituting the target set P. Over the N_g=960×240 grid points, we randomly sampled N_a=150,000 auxiliary points for A and N_b=750 out of the 960 grid time instances for boundary auxiliary points B.

We train the proposed PIDL on an NVIDIA Titan RTX GPU with 24 GB memory in Ubuntu 18.04.3. By default, we use the L^2 relative error to quantify the estimation error:

$$Err(\hat{\rho},\rho)=\frac{\sqrt{\sum_{r=1}^{N_g}\left|\hat{\rho}(t^{(r)},x^{(r)};\theta)-\rho(t^{(r)},x^{(r)})\right|^2}}{\sqrt{\sum_{r=1}^{N_g}\left|\rho(t^{(r)},x^{(r)})\right|^2}}. \quad (10)$$

The same error applies to velocity as $Err(\hat{u},u)$. The reason for using the relative error is to normalize the estimation inaccuracy while mitigating the influence from value scales of different data types.

We use the Xavier uniform initializer to initialize θ of PUNN, and train the PUNN through the PIDL architecture using Adam optimizer for 5,000 steps as a rough training. A followup fine-grained training is done by L-BFGS optimizer. The process terminates until the loss change of two consecutive steps is no larger than 10^{-6}, and θ^* is obtained. L-BFGS is used here because it is beneficial for a stable learning when the training dataset is relatively small. Our TSE settings use small observed data (up to a few thousand points) and meet the small-data situation.

The results of applying PIDL TSE for Greenshields-based ARZ dynamics using four loop detectors are presented in Fig.4, where PUNN is parameterized by the optimal θ^*. The estimation $\hat{\rho}$ and $\hat{u}$ is visually the same as the true dynamics in Fig.1, and the performance is justified by the snapshots, comparing the estimation curves over x at a certain time with the true curves. The L^2 relative errors of estimated density and average velocity are 4.3745 $\times 10^{-2}$ and 2.0075 $\times 10^{-2}$, respectively. Empirically, the difference between true and estimated values is visually indistinguishable when the errors are smaller than 6.70 $\times 10^{-2}$ and 2.90 $\times 10^{-2}$ for density and velocity, respectively. Performances with accuracy below these values are considered as "acceptable".

We conduct experiments on various numbers of traffic sensors of both sensing configurations. For each configuration, we conduct grid search of each hyperparameter between 0 and 150 with step 5, and the minimal-achievable estimation errors are presented in Tab.1. From the tables, we observe that more observed data lead to improved TSE performance and PIDL can efficiently achieve acceptable estimation accuracy in both sensing configurations. For detector-only setting, satisfactory performance is achieved when more than 3 loops are used (below the dashed line). For mixed setting, the performance can achieve satisfactory level when more than 1 loop or probe vehicle are used, implying the benefit of using trajectory information as it contains more frequent and diverse samples.

TSE and Parameter Discovery This subsection shows the ability of the PIDL for addressing both TSE and model parameter discovery. For this experiment, three model parameters $\boldsymbol{\lambda} = (\rho_{max}, u_{max}, \tau)$ are encoded as learning variables in PINN in Fig.2, and the residuals in Eq.(9) become $\hat{f}_1(t_a^{(j)}, x_a^{(j)}; \theta, \boldsymbol{\lambda})$ and $\hat{f}_2(t_a^{(j)}, x_a^{(j)}; \theta, \boldsymbol{\lambda})$, leading to a new objective function $Loss_{\theta,\boldsymbol{\lambda}}$. We use the same learning strategy to obtain $(\theta^*, \boldsymbol{\lambda}^*) = \text{argmin}_{\theta,\boldsymbol{\lambda}}\ Loss_{\theta,\boldsymbol{\lambda}}$, and compare the estimation and $\boldsymbol{\lambda}^*$ to the true values. The evaluations of model parameters are based on the L^2 relative errors in per-

lp	$Err(\hat{\rho}, \rho)$	$Err(\hat{u}, u)$	$\rho^*_{max}(\%)$	$u^*_{max}(\%)$	$\tau^*(\%)$
2	0.668	0.192	14.59	2.42	100.57
3	0.183	0.0488	0.43	0.11	3.32
4	0.0481	0.0153	1.24	0.23	7.79
5	0.0424	0.0103	0.56	0.07	7.11
(lp,tj)	$Err(\hat{\rho}, \rho)$	$Err(\hat{u}, u)$	$\rho^*_{max}(\%)$	$u^*_{max}(\%)$	$\tau^*(\%)$
(1,1)	0.150	0.0424	0.34	0.02	11.05
(1,2)	0.0667	0.0249	0.98	0.23	4.23
(2,1)	0.0651	0.0244	0.94	0.10	0.86

Table 2: Results for TSE and parameter discovery. $\boldsymbol{\lambda}^* = (\rho^*_{max}, u^*_{max}, \tau^*)$ are estimated parameters, compared to the true $\rho_{max} = 1.13$, $u_{max} = 1.02$, and $\tau = 0.02$.

centage. The results are shown in Tab.2.

In Tab.2, the errors below the dashed line are acceptable. In addition to accurate traffic state estimation, the PIDL is able to converge to the true parameters $\boldsymbol{\lambda}$. Specifically, for the case of (loops, trajs)=(1,2), acceptable TSE errors for both density and velocity are obtained, and the model parameters converge to ρ^*_{max}=1.14113, u^*_{max}=1.0177, τ^*=0.019155, which are very close to the true ones, demonstrating the advantages of the proposed PIDL.

ARZ-based PIDL for TSE on NGSIM Data

We evaluate the ARZ-based PIDL on the TSE problem using real traffic data, the Next Generation SIMulation (NGSIM) dataset, and compare the performance to baselines.

NGSIM Dataset

The public NGSIM dataset[1] provided by the US Federal Highway Administration records detailed information about vehicle trajectories on several road segments. We focus on a segment of the US Highway 101 (US 101), monitored by a camera mounted on top of a high building on June 15, 2005.

The locations/actions of vehicles in the monitored region for a total of around 680 meters and 2,770 seconds were converted from camera videos. This dataset has gained a great attention in recent traffic studies (Laval and Leclercq 2010; Montanino and Punzo 2013; Fan and Seibold 2013). We use the data averaged from all the involved five lanes to calculate the density and average velocity for every 30 meters over a 1.5-second period. After preprocessing (to remove the non-monitored vehicles at the beginning and end of the video), the density and velocity are visualized in Fig.5, where we end up with 21 and 1770 cells on the spatial and temporal dimensions, respectively. The center of each cell is treated as a grid point. The shockwave phenomenon can be clearly observed in the figure.

We assume the above-mentioned sensing configurations, where loop detectors are able to record the density and/or average velocity of cells on certain locations, and probe vehicles can only record the average velocity of cells on the trajectories. By default, detectors are evenly installed along the road. Out of all the vehicles in the dataset (around 4750), we randomly select a certain number (per a given penetration rate) as the probe vehicles to collect trajectory observations.

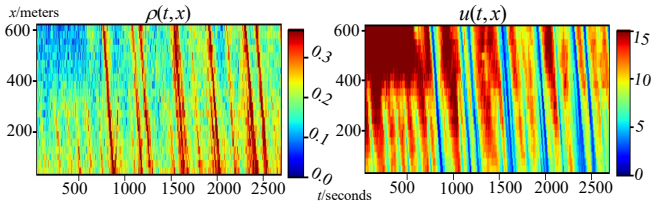

Figure 5: Visualization of NGSIM dataset.

TSE Methods and Baselines for NGSIM

We compare the proposed TSE method with selected baselines that are suitable for both sensing configurations.

Pure Neural Network (NN) The first baseline is a pure feedforward neural network, which directly uses the PUNN component in Fig.2 for learning the estimation rule from observed data. The loss function contains $Loss_\theta = MSE_o$ only, i.e., the physical discrepancy term is ignored. Of all the TSE methods in this section, the PUNN part has the same structure.

Extended Kalman Filter (EKF) The second baseline is a representative model-based TSE method, the ARZ-based extended Kalman filter (EKF) (Seo and Bayen 2017). This method applies the nonlinear version of the Kalman filter that linearizes the Greenshields-based ARZ model in a discrete manner to make state estimations.

LWR (First-Order)-based PIDL (LWR-PIDL) The third baseline assumes that we know a first-order traffic flow model, i.e., the LWR model with a three-parameter flux function, introduced by (Fan and Seibold 2013). Their model combines a three-parameter flux $Q(\rho)$ and the conservative law, $\rho_t + (Q(\rho))_x = 0$ to describe the density dynamics. The average velocity is calculated using $u = Q(\rho)/\rho$.

In implementation, we set model parameters in PINN as learning variables, and the initial values are determined from the observed data using least-squares fitting (Fan and Seibold 2013). Making model parameters as variables instead of fixed values may give more flexibility to PIDL to achieve better estimation. However, because no ground-truth model parameters are available for the real data, we skip the parameter discovery experiments and only present the TSE results.

We select 80% of grid points G as auxiliary points A (for regularization purpose, to have A dominate the domain is beneficial). The loss function has the same structure with Eq.(6), except for that the contents of $(\hat{f}_1, \hat{f}_2)$ and model parameters $\boldsymbol{\lambda}$ are different (see supplementary[2] for details). The parameter tuning uses grid search and the minimal-achievable errors are presented and compared. We use this first-order model baseline to form a PIDL and show how the choice of traffic model affects the estimation performance.

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

[2] https://rongyeshi.github.io/suppl3617.pdf

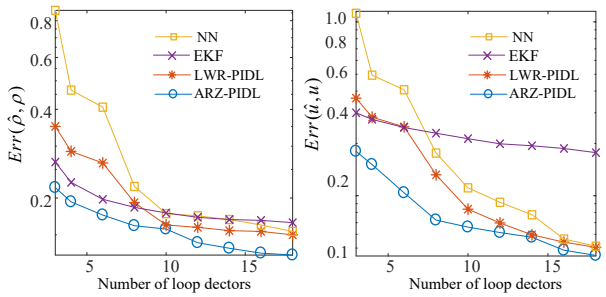

Figure 6: Results for loop detector-only config.

ARZ (Second-Order)-based PIDL (ARZ-PIDL) As mentioned in previous sections, this method encodes the Greenshields-based ARZ into the PINN, and the Eq.(6) is applied as the loss function. Other experimental setups are the same with those of the LWR-based PIDL baseline.

For fair comparison, NN, LWR-PIDL and ARZ-PIDL all use the Adam plus L-BFGS learning strategy.

Evaluations

We first conduct experiments using the loop detector-only configuration. The ARZ-based PIDL and baseline methods are applied to TSE on the NGSIM dataset with different numbers of loop detectors ranging between 3 and 18. In this sensing configuration, detectors are able to measure both density and velocity. The results are presented in Fig.6.

From Fig.6, we observe that the performances of the four TSE methods tend to improve to low estimation errors when the number of loop detectors increases, i.e., more observed data are collected and used for learning. The NN method makes an ample use of data to learn the dynamics. However, it cannot perform well when the number of loop detectors is small, and the data efficiency is low. By taking advantage of a first-order LWR traffic flow model, the LWR-PIDL method presents improvements in data efficiency. The EKF method performs better than the NN and LWR-PIDL when the number of loop detectors are small. And both NN and LWR-PIDL are better when sufficient data are available from more sensors. This is because EKF is a model-driven method, which makes sufficient use of the second-order ARZ traffic flow model to make proper estimations. However, the model cannot fully capture the complicated traffic dynamics in the real world, and the EKF's performance tends to flatten out. NN/LWR-PIDL can make ample use of data to estimate the dynamics and catch up with the EKF's performance when more data are collected. The ARZ-PIDL can make a full use of both the second-order model's advantages and the data to achieve the best estimation accuracy and data efficiency.

We conduct another experiment using the mixed sensing configuration, where loop detectors collect traffic density and probe vehicles record velocity along the trajectories. Probe vehicles are usually volunteer and constrained to a small scale. Thus, low penetration rates around or below 10% is considered in this paper. Some research in cities like Tokyo also reports the penetration rates around this level (Seo and Kusakabe 2015). Tab.3 shows the experimental results, where the penetration rate of 1% and 8% are used.

	$Err(\hat{\rho}, \rho)$				$Err(\hat{u}, u)$			
loops	NN	EKF	LWR-PIDL	ARZ-PIDL	NN	EKF	LWR-PIDL	ARZ-PIDL
8	0.272	0.189	0.192	**0.179**	0.527	0.465	0.161	**0.148**
10	0.238	0.177	0.180	**0.164**	0.505	0.260	0.143	**0.125**
14	0.227	0.168	0.159	**0.155**	0.484	0.253	0.129	**0.122**
8	0.169	0.166	0.165	**0.156**	0.098	0.275	0.0740	**0.0541**
10	0.162	0.156	0.157	**0.150**	0.060	0.301	0.0628	**0.0537**
14	0.156	0.143	0.153	**0.132**	0.058	0.269	0.0568	**0.0520**

Table 3: Results for mixed sensing config. The penetration rates of probe vehicles in the traffic flow are 1% and 8% for the upper half and lower half of the table, respectively. Probe vehicles are randomly selected from all the vehicles in the NGSIM data.

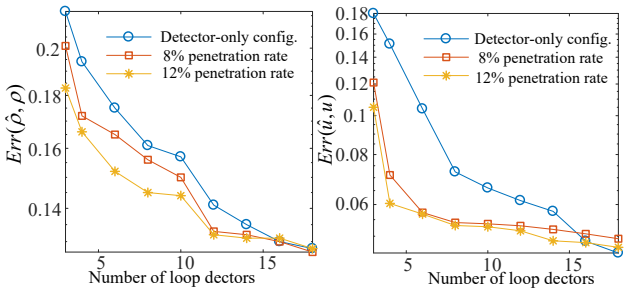

Figure 7: Comparison between the loop detector-only and mixed configurations with different penetration rates.

With a larger penetration rate, all the methods improve significantly in terms of velocity errors, because the amount of velocity observations increases. Similar to the previous experimental results, both the PIDL-based methods outperform the NN method because the integration of the physics-informed component leads to a more efficient use of the data. The ARZ-PIDL achieves the best overall performance. However, the NN method presents a faster improvement compared to the PIDL-based methods when a larger data amount is available. This is because the information gain provided by traffic flow models becomes limited when the data are sufficient, implying that the PIDL framework is a reasonable choice when the availability of data is low.

We further study the properties of the two sensing configurations when using the ARZ-PIDL. The comparison between the two sensing configurations are shown in Fig.7.

When a penetration rate is fixed, the total number of velocity observations is fixed when using the mixed sensing configuration, and we can observe a significantly better estimation on the velocity with small detector numbers compared to that of the detector-only configuration in Fig.7(right). As the number of loop detectors increases, though the velocity errors of ARZ-PIDL decrease consistently on both configurations, the performances of the mixed ones tend to flatten out. In terms of density estimation, the

curves of the mixed configuration are below the detector-only curve, implying that the ARZ-PIDL is able to "transfer" the information gain obtained from the velocity data to help improve the density estimation. This transferring property is made possible by the PINN component in the PIDL framework: the data of a physical quantity can affect the learning of another physical quantity via the physics-informed regularization term in the loss, where both quantities are involved and coupled.

Conclusion

This paper presents a physics-informed deep learning (PIDL) framework for the traffic state estimation (TSE) problem using second-order traffic flow model and detector-trajectory data. The PIDL consists of a data-driven component (PUNN) and a model-driven component (PINN). A second-order traffic flow model, the Greenshields-based ARZ model, has been encoded into the framework in the form of PINN to regularize the learning of the PUNN for TSE. Numerical examples and experiments on the real NGSIM traffic flow dataset demonstrate the advantages of the PIDL TSE method in terms of estimation accuracy and data efficiency.

There are two limitations of this paper. First, for the same traffic model, there could be various designs of PINN structures, and this paper only proposes one version of the design. It is worthy of researching how the changes in PINN structure affect the TSE performance and learning stability. Second, it is possible that the model parameters $\boldsymbol{\lambda}$ converge to different values with different initialization. Thus, the identifiability issue is worthy of investigating in the future work.

Acknowledgments

This work is financially supported by the seed grant from the Data Science Institute (DSI) at Columbia University. The authors of this paper would like to thank Kuang Huang and Professor Qiang Du for their helpful comments.

References

Aw, A.; and Rascle, M. 2000. Resurrection of "Second Order" Models of Traffic Flow. *SIAM Journal on Applied Mathematics* 60(3): 916–938.

Di, X.; Liu, H. X.; and Davis, G. A. 2010. Hybrid extended Kalman filtering approach for traffic density estimation along signalized arterials: Use of global positioning system data. *Transportation Research Record* 2188(1): 165–173.

Fan, S.; and Seibold, B. 2013. Data-fitted first-order traffic models and their second-order generalizations: Comparison by trajectory and sensor data. *Transportation Research Record* 2391(1): 32–43.

Flynn, M. R.; Kasimov, A. R.; Nave, J.-C.; Rosales, R. R.; and Seibold, B. 2009. Self-sustained nonlinear waves in traffic flow. *Physical Review E* 79(5): 056113.

Greenshields, B.; Channing, W.; and Miller, H. 1935. A study of traffic capacity. *Highway Research Board Proceedings* 14: 448–477.

Laval, J. A.; and Leclercq, L. 2010. A mechanism to describe the formation and propagation of stop-and-go waves in congested freeway traffic. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 368(1928): 4519–4541.

LeVeque, R. J. 2002. *Finite Volume Methods for Hyperbolic Problems*, volume 31. Cambridge Univ. Press.

Lighthill, M. J.; and Whitham, G. B. 1955. On kinematic waves II. A theory of traffic flow on long crowded roads. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences* 229(1178): 317–345.

Montanino, M.; and Punzo, V. 2013. Making NGSIM data usable for studies on traffic flow theory: Multistep method for vehicle trajectory reconstruction. *Transportation Research Record* 2390(1): 99–111.

Payne, H. J. 1971. Model of freeway traffic and control. *Mathematical Model of Public System* 51–61.

Raissi, M. 2018. Deep hidden physics models: Deep learning of nonlinear partial differential equations. *Journal of Machine Learning Research* 19(1): 932–955.

Raissi, M.; and Karniadakis, G. E. 2018. Hidden physics models: Machine learning of nonlinear partial differential equations. *Journal of Computational Physics* 357: 125–141.

Richards, P. I. 1956. Shock waves on the highway. *Operations Research* 4(1): 42–51.

Seo, T.; and Bayen, A. M. 2017. Traffic state estimation method with efficient data fusion based on the Aw-Rascle-Zhang model. In *IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 1–6.

Seo, T.; Bayen, A. M.; Kusakabe, T.; and Asakura, Y. 2017. Traffic state estimation on highway: A comprehensive survey. *Annual Reviews in Control* 43: 128–151.

Seo, T.; and Kusakabe, T. 2015. Probe vehicle-based traffic flow estimation method without fundamental diagram. *Transportation Research Procedia* 9: 149–163.

Wang, Y.; Papageorgiou, M.; and Messmer, A. 2008. Real-time freeway traffic state estimation based on extended Kalman filter: Adaptive capabilities and real data testing. *Transportation Research Part A: Policy and Practice* 42(10): 1340–1358.

Wang, Y.; Papageorgiou, M.; Messmer, A.; Coppola, P.; Tzimitsi, A.; and Nuzzolo, A. 2009. An adaptive freeway traffic state estimator. *Automatica* 45(1): 10–24.

Whitham, G. B. 1974. *Linear and Nonlinear Waves*. New York: John Wiley & Sons Inc.

Zhang, H. M. 2002. A non-equilibrium traffic model devoid of gas-like behavior. *Transportation Research Part B: Methodological* 36(3): 275–290.

Zheng, Z.; Yang, Y.; Liu, J.; Dai, H.-N.; and Zhang, Y. 2019. Deep and embedded learning approach for traffic flow prediction in urban informatics. *IEEE Transactions on Intelligent Transportation Systems* 20(10): 3927–3939.