A Physics-Informed Deep Learning Paradigm for Traffic State and Fundamental Diagram Estimation

Rongye Shi[ID], *Member, IEEE*, Zhaobin Mo, Kuang Huang, Xuan Di[ID], *Member, IEEE*, and Qiang Du

Abstract—Traffic state estimation (TSE) bifurcates into two main categories, model-driven and data-driven (e.g., machine learning, ML) approaches, while each suffers from either deficient physics or small data. To mitigate these limitations, recent studies introduced hybrid methods, such as physics-informed deep learning (PIDL), which contains both model-driven and data-driven components. This paper contributes an improved paradigm, called physics-informed deep learning with a fundamental diagram learner (PIDL + FDL), which integrates ML terms into the model-driven component to learn a functional form of a fundamental diagram (FD), i.e., a mapping from traffic density to flow or velocity. The proposed PIDL + FDL has the advantages of performing the TSE learning, model parameter identification, and FD estimation simultaneously. This paper focuses on highway TSE with observed data from loop detectors, using traffic density or velocity as traffic variables. We demonstrate the use of PIDL + FDL to solve popular first-order and second-order traffic flow models and reconstruct the FD relation as well as model parameters that are outside the FD term. We then evaluate the PIDL + FDL-based TSE using the Next Generation SIMulation (NGSIM) dataset. The experimental results show the superiority of the PIDL + FDL in terms of improved estimation accuracy and data efficiency over advanced baseline TSE methods, and additionally, the capacity to properly learn the unknown underlying FD relation.

Index Terms—Traffic state estimation, traffic flow models, fundamental diagram learner, physics-informed deep learning.

I. INTRODUCTION

TRAFFIC state estimation (TSE) refers to the data mining problem of reconstructing traffic state variables, including but not limited to flow, density, and speed, on road segments using partially observed data from traffic sensors [1]. TSE approaches can be briefly divided into two main categories: model-driven and data-driven [2]. A model-driven approach is based on a priori knowledge of traffic dynamics, usually described by a physical model, e.g., the Lighthill-Whitham-Richards (LWR) model [3], [4] and Aw-Rascle-Zhang (ARZ) model [5], [6], to estimate the traffic state using partial observation. It assumes the model to be representative of the real-world traffic dynamics such that the unobserved values can be properly added using the model with small data. The disadvantage is that existing models, which are provided by different modelers, may only capture limited dynamics of the real-world traffic, resulting in low-quality estimation in the case of inappropriately-chosen models and poor model calibrations. Paradoxically, sometimes, verifying or calibrating a model requires a large amount of observed data, undermining the data efficiency of model-driven approaches.

A data-driven approach is to infer traffic states based on the dependence learned from historical data using statistical or machine learning (ML) methods. Approaches of this type do not use any explicit traffic models or other theoretical assumptions, and can be treated as a "black box" with no interpretable and deductive insights. The disadvantage is that in order to maintain a good generalizable inference to long-term unobserved values, massive and representative historical data are a prerequisite, leading to high demands on data collection infrastructure and enormous installation-maintenance costs.

To mitigate the limitations of the above-mentioned TSE approaches, hybrid TSE methods are introduced, which integrate the traffic knowledge in the form of traffic flow models to ML models for TSE. The hybrid methods based on the learning paradigm of physics-informed deep learning (PIDL) are gaining increasing attentions in recently years, and is the focus of this paper. PIDL contains both a model-driven component (a physics-informed neural network for regularization) and a data-driven component (a physics-uninformed neural network for estimation), making possible the integration of the strengths of both model-driven and data-driven approaches while overcoming the weaknesses of either.

Despite that the addition of physics could guide the training of PIDL efficiently, complicated mathematical formulas could instead make the PIDL difficult to train. There are many theoretical attempts made to add sophistication (usually in the form of complicated terms) to the FD relation for an improved description of the dynamics. To balance the sophistication

and trainability of encoding physics, this paper explores a promising direction by approximating the FD relation with an ML surrogate, instead of hard-encoding an FD equation. Following this direction, we introduce an improved PIDL paradigm, called physics-informed deep learning with a fundamental diagram learner (PIDL + FDL), which integrates an ML surrogate (e.g., an NN) into the model-driven component to represent the fundamental diagram (FD) in the macroscopic traffic flow models and estimate the FD relation. We focus on highway TSE with observed data from loop detectors, using traffic density or velocity as the traffic variables. This paper makes the following contributions:

- We propose the PIDL + FDL-based TSE method that possesses advantages to

  - Perform the TSE with improved estimation accuracy: A proper integration of ML surrogates may avoid directly encoding the complicated terms in PIDL and trade off between the sophistication of the model-driven aspect of PIDL and the training flexibility, making the framework a better fit to the TSE problem;

  - Perform the FD estimation: The PIDL + FDL uses an ML surrogate to directly learn the underlying FD relation without any FD output measurements, i.e., the ML surrogate is purely trained under the physical regularization from PIDL, making it more likely to learn a suitable relation along with the TSE training. It can also get around the calibration of parameters inside the FD equation;

  - Perform the model parameter identification: For a complete traffic model reconstruction, in addition to the FD estimation, there may exist model parameters outside the FD term that need to be learned and the proposed PIDL + FDL can conduct the model parameter identification jointly.

- We validate the PIDL + FDL performance with both numerical experiments and real-world data: To demonstrate the strengths of the PIDL + FDL, we design the PIDL + FDL architectures for the traffic dynamics governed by the Greenshields-based LWR model and Greenshields-based ARZ model, respectively. Additionally, experiments using the real-world data, the Next Generation SIMulation (NGSIM) dataset, are conducted. The experimental results show the advantages of PIDL in terms of estimation accuracy and data efficiency over baselines and the capacity to properly estimate the FD relation and model parameters.

The rest of this paper is organized as follows. Section II briefs related work on TSE and PIDL. Section III formalizes the PIDL + FDL framework for TSE. Sections IV and V detail the designs and experiments of PIDL + FDL for Greenshields-based LWR and Greenshields-based ARZ, respectively. Section VI evaluates the PIDL + FDL on NGSIM data over baselines. Section VII concludes our work.

II. RELATED WORK OF TRAFFIC STATE ESTIMATION

Model-driven approaches accomplish their estimation process relying on traffic flow models, such as the LWR and Aw-Rascle-Zhang (ARZ) for one-dimensional traffic flow (link models) and junction models [7] for network traffic flow (node models). Most estimation approaches in this category are data assimilation (DA) based, which attempt to find "the most likely state", allowing observation to correct the model's prediction. Popular examples include the Kalman filter (KF) and its variants (e.g., extended KF [8]–[10]), which find the state that maximizes the conditional probability of the next state given current estimate. Other than KF-like methods, particle filter (PF) [11] with improved nonlinear representation, adaptive smoothing filter (ASF) [12] for combining multiple sensing data, were proposed to improve and extend different aspects of the TSE process. In addition to DA-based methods, there have been many studies utilizing microscopic trajectory models to simulate the state or vehicle trajectories given some boundary conditions from data [13], [14].

Data-driven approaches estimate traffic states using historical data or streaming data without explicit traffic flow models. Early studies considered relatively simple statistical methods, such as heuristic imputation techniques using empirical statistics of historical data [15] and regression methods [16], [17]. To handle more complicated traffic data, ML approaches were involved, including principal component analysis (PCA) and its variants [18], [19], k-nearest neighbors (kNN) [20], and probabilistic graphical models (i.e., Bayesian networks) [21]. Deep learning models, such as long short term memory (LSTM) and deep embedded models, have recently been applied for traffic flow prediction [22]–[24].

Each of the two approaches has disadvantages, and it is promising to develop a framework to integrate physics to ML. Such a hybrid paradigm has gained increasing interests recently. Yuan *et al.* [25] proposed to leverage a hybrid framework, physics regularized Gaussian process [26] for macroscopic traffic flow modeling and TSE. The hybrid methods using the PIDL framework [27], [28] recently becomes an active field. Huang and Agarwal [29] studied the use of PIDL to encode the Greenshields-based LWR and validated it in the loop detector scenarios using SUMO simulated data. Barreau *et al.* [30]–[32] studied the probe vehicle sensors and developed coupled micro-macro models for PIDL to perform TSE. Shi *et al.* [33] extended the PIDL-based TSE to the second-order ARZ with observed data from both loop detectors and probe vehicles. Thodi *et al.* [34] proposed a kinematic wave based Deep Convolutional Neural Network (Deep CNN) to estimate high resolution traffic speed dynamics from sparse probe vehicle trajectories.

As to model reconstruction, which is another feature of PIDL-based TSE, this paper only assumes a traffic flow conservation equation and optionally, a momentum equation for the velocity field, without specifying any mathematical relation between traffic quantities. The aforementioned related studies in [30], [32] directly fit a velocity function using measured density and velocity from probe vehicles before or during the PIDL training. In contrast, we focus on a more general case, where the output of the FD function is unobserved from sensors, and the end-to-end FD relation is to be learned directly using ML surrogates under the PIDL framework. In summary, this paper contributes to the trend of developing hybrid methods for TSE and model reconstruction, especially

with both FD estimation and model parameter identification involved. We believe that using PIDL for TSE and other transportation-related problems will become a popular topic in the transportation community.

III. MATHEMATICAL SETTING FOR PIDL + FDL

This section introduces the PIDL + FDL framework in the context of TSE at a high level.

A. PIDL for TSE

Consider a traffic flow dynamics of a road segment that is governed by a set of non-linear equations (e.g., partial differential equations, PDEs):

$$\mathcal{N}(\boldsymbol{M}(t, x), \boldsymbol{Q}; \boldsymbol{\lambda}) = \boldsymbol{0}, \quad x \in [0, L], \ t \in [0, T], \quad (1)$$

where $L \in \mathbb{R}^+$, $T \in \mathbb{R}^+$. We use bold symbols to denote vectors by default. The operator $\mathcal{N}$ contains the governing non-linear equations of the traffic flow dynamics, while $\boldsymbol{M}(t, x)$ contains the traffic state variables, such as the traffic density $\rho(t, x)$ and velocity $u(t, x)$. $\boldsymbol{\lambda}$ contains the model parameters. The model includes intermediate unobserved traffic variables $\boldsymbol{Q}$ that have some hidden relationship with $\boldsymbol{M}(t, x)$. Thus, the dynamics can be represented by

$$\mathcal{N}(\boldsymbol{M}, \boldsymbol{Q}(\boldsymbol{M}); \boldsymbol{\lambda}) = \boldsymbol{0}, \quad (2)$$

and $\boldsymbol{M}$ stands for $\boldsymbol{M}(t, x)$. For general discussion, the values of $\boldsymbol{Q}$ are not assumed to be directly observable, and the relation is either unknown or deduced based on assumptions which may be deficient. The TSE problem is to reconstruct the traffic states $\boldsymbol{M}$ at each point (t, x) in a continuous domain from partial observation of $\boldsymbol{M}$. Accordingly, the continuous spatio-temporal domain D is a set of points: $D = \{(t, x) | \forall t \in [0, T], x \in [0, L]\}$. We represent this continuous domain in a discrete manner using grid points $G \in D$ that are evenly deployed throughout the domain. We define the set of grid points as $G = \{(t^{(r)}, x^{(r)}) | r = 1, \ldots, N_g\}$. The total number of grid points, N_g, controls the fine-grained level of G as a representation of the continuous domain.

PIDL approximates $\boldsymbol{M}(t, x)$ 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 our TSE study), which is parameterized by θ. We denote the approximation of $\boldsymbol{M}(t, x)$ from PUNN as $\hat{\boldsymbol{M}}(t, x; \theta)$. When $\mathcal{N}$, $\boldsymbol{Q}$ and $\boldsymbol{\lambda}$ are known, during the learning phase of PUNN (i.e., to find the optimal θ for PUNN), the following equation defines the residual values of the approximation $\hat{\boldsymbol{M}}(t, x; \theta)$:

$$\hat{\boldsymbol{f}}(t, x; \theta) := \mathcal{N}(\hat{\boldsymbol{M}}(t, x; \theta), \boldsymbol{Q}(\hat{\boldsymbol{M}}(t, x; \theta)); \boldsymbol{\lambda}), \quad (3)$$

which is designed according to the traffic flow model in Eq. (2). The calculation of residual $\hat{\boldsymbol{f}}(t, x; \theta)$ is done by a *physics-informed neural network* (PINN). This network can compute $\hat{\boldsymbol{f}}(t, x; \theta)$ directly using $\hat{\boldsymbol{M}}(t, x; \theta)$, the output of PUNN, as its input. When $\hat{\boldsymbol{M}}(t, x; \theta)$ is closer to the true value $\boldsymbol{M}(t, x)$, the residual $\hat{\boldsymbol{f}}$ will be closer to $\boldsymbol{0}$. PINN introduces no new parameters, and thus, shares the same θ of PUNN.

In most cases, even the model is given, the model parameters $\boldsymbol{\lambda}$ are unknown and can be made as learning variables in PINN for model parameter identification. The residual $\hat{\boldsymbol{f}}$ is then redefined as

$$\hat{\boldsymbol{f}}(t, x; \theta, \boldsymbol{\lambda}) := \mathcal{N}(\hat{\boldsymbol{M}}(t, x; \theta), \boldsymbol{Q}(\hat{\boldsymbol{M}}(t, x; \theta)); \boldsymbol{\lambda}). \quad (4)$$

This paper assumes unknown model parameters by default. In PINN, $\hat{\boldsymbol{f}}(t, x; \theta, \boldsymbol{\lambda})$ is calculated by automatic differentiation technique [35], which can be done by the function `tf.gradient` in Tensorflow. The activation functions and the connecting structure of neurons in PINN are designed to conduct the differential operation in Eq. (4). We would like to emphasize that, the connecting weights in PINN have fixed values which are determined by the traffic flow model and $\boldsymbol{\lambda}$ are encoded as learning variables. Thus, the residual $\hat{\boldsymbol{f}}$ is parameterized by both θ and $\boldsymbol{\lambda}$.

The training data for PIDL consist of (1) *observation points* $O = \{(t_o^{(i)}, x_o^{(i)}) | i = 1, \ldots, N_o\}$, (2) *target values* $P = \{\boldsymbol{M}^{(i)} | i = 1, \ldots, N_o\}$ (i.e., the true traffic states at the observation points), and (3) *auxiliary points* $A = \{(t_a^{(j)}, x_a^{(j)}) | j = 1, \ldots, N_a\}$. i and j are the indexes of 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 (indexed by i). This paper uses the term, observed data, to denote $\{O, P\}$. Both O and A are subsets of the grid points G (i.e., $O \in G$ and $A \in G$).

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:

$$
\begin{aligned}
Loss_{\theta, \lambda} &= \alpha \cdot MSE_o + \beta \cdot MSE_a \\
&= \alpha \cdot \underbrace{\frac{1}{N_o} \sum_{i=1}^{N_o} |\hat{\boldsymbol{M}}(t_o^{(i)}, x_o^{(i)}; \theta) - \boldsymbol{M}^{(i)}|^2}_{data\ discrepancy} \\
&\quad + \beta \cdot \underbrace{\frac{1}{N_a} \sum_{j=1}^{N_a} |\hat{\boldsymbol{f}}(t_a^{(j)}, x_a^{(j)}; \theta, \boldsymbol{\lambda})|^2}_{physical\ discrepancy}. \quad (5)
\end{aligned}
$$

where α and β are hyperparameters for balancing the contribution to the loss made by data discrepancy and physical discrepancy, respectively. The data discrepancy is defined as the mean square error (MSE) between approximation $\hat{\boldsymbol{M}}$ on O and target values P. The physical discrepancy is the MSE between residual values on A and $\boldsymbol{0}$, quantifying the extent to which the approximation deviates from the traffic model.

Given the training data, we apply neural network training algorithms to solve $(\theta^*, \boldsymbol{\lambda}^*) = \text{argmin}_{\theta, \lambda} Loss_{\theta, \lambda}$. Then, the $\boldsymbol{\lambda}^*$-parameterized traffic flow model of Eq. (4) is the most likely physics that generates the observed data, and the θ^*-parameterized PUNN can then be used to approximate the traffic states on G, which are consistent with the reconstructed traffic flow model in Eq. (2).

B. PIDL + FDL for TSE

As has been discussed previously, the PIDL-based TSE methods may perform poorly when informed by a highly sophisticated traffic flow model. This is because the models may contain complicated terms that are unfriendly to differentiation-based learning (e.g., square root operators of learning variables in the denominator, etc.), making the training and performance very sensitive to the structural design of PINN. Many efforts such as variable conversion, decomposition and factorization need to be made to have the PINN trainable and the loss to converge. In our framework of Eq.(2), these "unfriendly" terms can be contained as part of the hidden relation Q. To address the issues of PIDL-based TSE, we propose to use an ML surrogate $\hat{Q}$ to directly represent the Q and learn the relation under the PIDL framework, instead of hard-encoding a complicated term in PINN.

The advantages of properly introducing an ML surrogate of Q are two-fold: (1) An ML term is usually differentiation-friendly, giving the PIDL more flexibility to achieve an improved TSE accuracy. (2) No assumptions are made to the hidden relation, and it is possible to learn a more suitable $\hat{Q}$ when trained under the physical regularization from PINN.

This paper focuses on one kind of hidden relationships, the fundamental diagram (FD), and the corresponding learning paradigm is called *PIDL with an FD Learner* (PIDL + FDL). Specifically, the FD Learner, formalized as $\hat{Q}(\hat{M}; \omega)$, can be designed as a neural network parameterized by ω to represent the unknown FD relation, which takes the estimated traffic variables $\hat{M}(t, x; \theta)$ as its input. The residual of Eq.(4) is redefined as the following

$$\hat{f}(t, x; \theta, \omega, \lambda) := \mathcal{N}(\hat{M}(t, x; \theta), \hat{Q}(\hat{M}(t, x; \theta); \omega); \lambda). \quad (6)$$

The loss function becomes

$$
\begin{aligned}
Loss_{\theta,\omega,\lambda} &= \alpha \cdot MSE_o + \beta \cdot MSE_a \\
&= \alpha \cdot \frac{1}{N_o} \sum_{i=1}^{N_o} |\hat{M}(t_o^{(i)}, x_o^{(i)}; \theta) - M^{(i)}|^2 \\
&\quad + \beta \cdot \frac{1}{N_a} \sum_{j=1}^{N_a} |\hat{f}(t_a^{(j)}, x_a^{(j)}; \theta, \omega, \lambda)|^2. \quad (7)
\end{aligned}
$$

Using the training data, we apply neural network training algorithms to solve $(\theta^*, \omega^*, \lambda^*) = \text{argmin}_{\theta,\omega,\lambda} Loss_{\theta,\omega,\lambda}$. Then, in addition to TSE learning and model parameter identification, the FD estimation is conducted automatically. The ω^*-parameterized $\hat{Q}$ can be used to represent the unknown hidden fundamental diagram relation. Note that the values of Q are not assumed to be observable, and thus, are not part of the data (i.e., to directly learn the $\hat{Q}$ from data cannot apply).

In some cases, the curve of the learned $\hat{Q}$ may present abnormal shapes on edge conditions. To mitigate this, one can encode prior knowledge into the loss as an additional regularization term $Reg(\hat{Q})$ to reshape the FD. As an example, we can use Q to represent the density-flow relation, i.e., the flux function (one typical kind of FD), mapping the density ρ to the flow value, which is denoted as a *scalar* $Q(\rho)$. Existing theoretical works usually assume Q to be concave with respect

to the traffic density ρ. To impose the concavity property, we design the following regularization term:

$$Reg(Q) = \int_a^b \max(0, \frac{\partial^2 Q(\rho)}{\partial \rho^2}) d\rho, \quad (8)$$

where the hyperparameters a and b determine the interval of ρ on which the reshaping takes effects without interfering the learning on other regions. We propose this design because most abnormal shapes only occur on edge region and it is not necessary to regularize over the whole traffic density domain. We apply $Loss_{\theta,\omega,\lambda} = \alpha \cdot MSE_o + \beta \cdot MSE_a + \xi \cdot Reg(\hat{Q})$ in the learning phase and properly reshape the learned FD curves.

IV. PIDL + FDL FOR GREENSHIELDS-BASED LWR

The first numerical example aims to show the capability of our method to estimate the traffic dynamics governed by the LWR model with a Greenshields flux function.

Define flow rate Q (a scalar) to be the number of vehicles passing a specific position on the road per unit time, and traffic density ρ to be the average number of vehicles per unit length of the road. The traffic flux $Q(\rho)$ describes Q as a function of ρ, which is the FD relation of interest in this numerical example. We treat ρ as the basic traffic state variable to estimate. Greenshields flux [36] is a basic and popular choice of $Q(\rho)$, which is defined as $Q(\rho) = \rho u_{max}(1 - \rho/\rho_{max})$, where u_{max} and ρ_{max} are maximum velocity and maximum (jam) density, respectively. This flux function has a quadratic form with two coefficients u_{max} and ρ_{max}.

The LWR model [3], [4] describes the macroscopic traffic flow dynamics as $\rho_t + (Q(\rho))_x = 0$, which is derived from a conservation law of vehicles. In order to reproduce more phenomena in observed traffic data, such as delayed driving behaviors due to drivers' reaction time, diffusively corrected LWRs were introduced, by adding a diffusion term, containing a second-order derivative ρ_{xx}. We focus on one version of the diffusively corrected LWRs: $\rho_t + (Q(\rho))_x = \epsilon \rho_{xx}$, where ϵ is the diffusion coefficient.

In this section, we study the Greenshields-based LWR traffic flow model of a "ring road":

$$
\begin{aligned}
\rho_t + (Q(\rho))_x &= \epsilon \rho_{xx}, \quad t \in [0, 3], \ x \in [0, 1], \\
Q(\rho) &= \rho \cdot u_{max} \left(1 - \frac{\rho}{\rho_{max}}\right) \quad (FD \ relation), \\
\rho(t, 0) &= \rho(t, 1) \quad (boundary \ condition \ 1), \\
\rho_x(t, 0) &= \rho_x(t, 1) \quad (boundary \ condition \ 2), \quad (9)
\end{aligned}
$$

where $\rho_{max} = 1.0$, $u_{max} = 1.0$, and $\epsilon = 0.005$. ρ_{max} and u_{max} are usually determined by physical restrictions of the road and vehicles.

Given the bell-shaped initial $0.1 + 0.8e^{-(\frac{x-0.5}{0.2})^2}$, $x \in [0, 1]$, we apply the Godunov scheme [37] to solve Eqs. (9) on 960 (time) × 240 (space) grid points G evenly deployed throughout the $[0, 3] \times [0, 1]$ domain. In this case, the total number of grid points G is $N_g = 960 \times 240$. The numerical solution is shown in Fig. 2 (see the heat map background). From the figure, we can visualize the dynamics as follows: At $t = 0$, there is a peak density at the center of the road segment, and this peak evolves to propagate along the
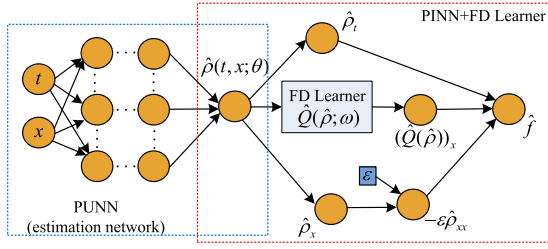
Fig. 1. PIDL + FDL architecture for first-order traffic dynamics, consisting of a PUNN for traffic density estimation and a PINN + FD Learner for calculating the residual Eq. (10). The model parameter ϵ is held by a variable node (blue rectangular nodes). All connecting weights are predefined and fixed in PINN.

Fig. 2. A sparse presentation of the deployment of observation points O at loop detectors, auxiliary points A randomly selected from G, and boundary auxiliary points B deployed at the boundaries $x = 0$ and $x = 1$ for certain time points. The heatmap is the numerical solution of Eqs. (9) using the Godunov scheme. We treat the numerical solution as the ground truth.

direction of x, which is known as the phenomenon of traffic shockwave. Since this is a ring road, the shockwave reaching $x = 1$ continues at $x = 0$. This numerical solution of the Greenshields-based LWR model is treated as the ground-truth traffic density. We will apply a PIDL + FDL-based approach method to estimate the entire traffic density field using observed data as well as to estimate the FD relation and model parameters.

A. PIDL + FDL Architecture Design

The authors' previous work [33] has shown the capacity of PIDL to perform both TSE and model parameter identification when the closed traffic flow model is given. Here we are only given the knowledge of conservation law and boundary conditions, i.e., the FD relation is unknown and no direct observation of Q is available.

We employ a neural network $\hat{Q}(\cdot; \omega)$ to estimate the traffic flow from the traffic density and to represent the FD relation of interest. Based on Eqs. (9), we define the residual value of PUNN's traffic density estimation $\hat{\rho}(t, x; \theta)$ as

$$\hat{f}(t, x; \theta, \omega, \epsilon) := \hat{\rho}_t(t, x; \theta) + (\hat{Q}(\hat{\rho}(t, x; \theta); \omega))_x$$
$$- \epsilon \hat{\rho}_{xx}(t, x; \theta). \quad (10)$$

Note that the model parameter λ contains the coefficient ϵ only.

Given the definition of $\hat{f}$, the corresponding PIDL + FDL architecture is shown in Fig. 1. This architecture consists of a PUNN for traffic density estimation, followed by a PINN + FD Learner for calculating the residual Eq. (10). The PUNN parameterized by θ is designed as a fully-connected feedforward neural network with 8 hidden layers and 20 hidden nodes in each hidden layer. Hyperbolic tangent function (tanh) is used as the activation function for each hidden neuron in PUNN. In contrast, in PINN, connecting weights are fixed and the activation function of each node is designed to conduct specific nonlinear operation for calculating an intermediate (hidden) value of $\hat{f}$. The flow value is calculated by a separate neural network $\hat{Q}(\hat{\rho}; \omega)$ with two hidden layers and 20 hidden nodes for each. The model parameter ϵ is held by a variable node (blue rectangular nodes).

To customize the training of PIDL + FDL to Eqs. (9), in addition to the training data O, P and A defined in Section III-A, we need to introduce *boundary auxiliary points*
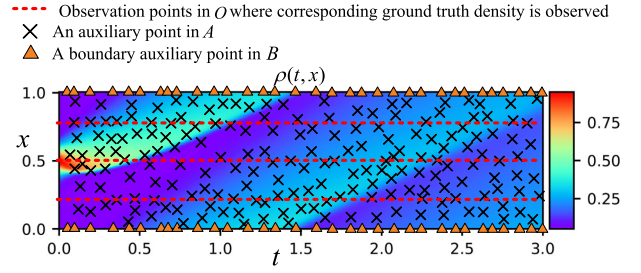
$B = \{(t_b^{(k)}, 0)|k = 1, \ldots, N_b\} \cup \{(t_b^{(k)}, 1)|k = 1, \ldots, N_b\}$, for learning the two boundary conditions in Eqs. (9).

For experiments of state estimation with both parameter identification and FD estimation, we design the following loss

$$Loss_{\theta, \omega, \epsilon} = \alpha \cdot MSE_o + \beta \cdot MSE_a + \gamma \cdot B1 + \eta \cdot B2$$
$$= \frac{\alpha}{N_o} \sum_{i=1}^{N_o} |\hat{\rho}(t_o^{(i)}, x_o^{(i)}; \theta) - \rho^{(i)}|^2$$
$$+ \frac{\beta}{N_a} \sum_{j=1}^{N_a} |\hat{f}(t_a^{(j)}, x_a^{(j)}; \theta, \omega, \epsilon)|^2$$
$$+ \frac{\gamma}{N_b} \sum_{k=1}^{N_b} |\hat{\rho}(t_b^{(k)}, 0; \theta) - \hat{\rho}(t_b^{(k)}, 1; \theta)|^2$$
$$+ \frac{\eta}{N_b} \sum_{k=1}^{N_b} |\hat{\rho}_x(t_b^{(k)}, 0; \theta) - \hat{\rho}_x(t_b^{(k)}, 1; \theta)|^2. \quad (11)$$

$B1$, scaled by γ, is the MSE between estimations at the two boundaries $x = 0$ and $x = 1$. $B2$, scaled by η, quantifies the difference of first order derivatives at the two boundaries.

B. TSE + FDL Using Observation From Loop Detectors

We justify the capacity of PIDL + FDL in Fig. 1 for estimating the traffic density field using observation from loop detectors, i.e., only the traffic density at certain locations where loop detectors are installed can be observed. By default, loop detectors are evenly located along the road. To be specific, the grid points at certain locations are used as the observation points O, and their corresponding densities constitute the target values P for training. There are $N_a = 100,000$ auxiliary points in A randomly selected from grid points G. $N_b = 650$ out of 960 *grid time points* (i.e., the time points on the temporal dimension of G) are randomly selected to create boundary auxiliary points B. A sparse version of the deployment of O, A and B in the spatio-temporal domain is shown in Fig. 2. Each observation point is associated with a target value in P. Note O, A and B are all subsets of G.

We train the proposed PIDL + FDL on an NVIDIA Titan RTX GPU with 24 GB memory. By default, we use the $\mathbb{L}^2$ relative error on G to quantify the estimation error of the
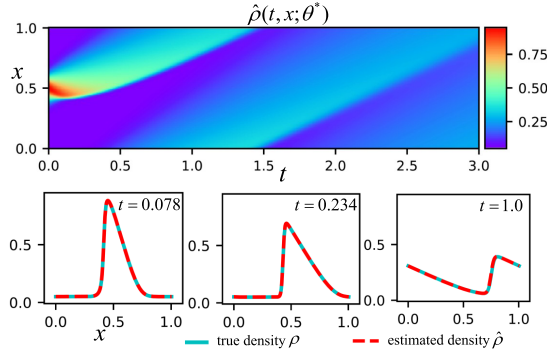
Fig. 3. Top: Estimation of the traffic density dynamics $\hat{\rho}(t, x; \theta^*)$ on grid points G in the domain using the trained PUNN. Bottom: Snapshots of estimated and true traffic density at certain time points.

entire domain:

$$Err(\hat{\rho}, \rho) = \frac{\sqrt{\sum_{r=1}^{N_g} \left| \hat{\rho}(t^{(r)}, x^{(r)}; \theta) - \rho(t^{(r)}, x^{(r)}) \right|^2}}{\sqrt{\sum_{r=1}^{N_g} \left| \rho(t^{(r)}, x^{(r)}) \right|^2}}. \quad (12)$$

The reason for choosing the $\mathbb{L}^2$ relative error is to normalize the estimation inaccuracy, mitigating the influence from the scale of true density values. One remark is that there are some TSE methods (e.g., non parametric ones) that do not perform any estimation on the observation points and directly use the target values there. For these cases, the observation points will be removed from G before calculating Eq.(12).

We use the Xavier uniform initializer [38] to initialize θ of PUNN and ω of FD Learner (FDL). This neural network initialization method takes the number of a layer's incoming and outgoing network connections into account when initializing the weights of that layer, which may lead to a good convergence. The ϵ is initialized at 0. Then, we train the PUNN, FDL and ϵ through the PIDL + FDL architecture using a popular stochastic gradient descent algorithm, the Adam optimizer [39], for a rough training. A follow-up fine-grained training is done by the L-BFGS optimizer [40] for stabilizing the convergence, and the process terminates until the loss change of two consecutive steps is no larger than 10^{-16}. This training process converges to a local optimum θ^*, ω^* and ϵ^* that minimize the loss in Eq. (11).

We would like to clarify that in this paper, the training data are the observed data from detectors, i.e., the traffic states on the points at certain locations where loops are equipped.

The results of applying the PIDL + FDL with 4 loops to the Greenshields-based LWR dynamics is presented in Fig. 3, where PUNN is parameterized by the optimal θ^*. As shown in Fig. 3, the estimation $\hat{\rho}(t, x; \theta^*)$ is visually the same as the true dynamics $\rho(t, x)$ in Fig. 2. By looking into the estimated and true traffic density over x at certain time points, there is a good agreement between two traffic density curves. The $\mathbb{L}^2$ estimation error $Err(\hat{\rho}, \rho)$ is 1.287×10^{-2}. Empirically, the difference cannot be visually distinguished when the estimation error is smaller than 6×10^{-2}.

We change the number of loop detectors. For a fixed number of loop detectors, we use grid search for hyperparameter tuning by default. Specifically, since Adam optimizer is scale

loops	2	3	4	5
$Err(\hat{\rho}, \rho)$	0.6021	0.03327	0.01287	0.004646
ϵ^*	3.12573	0.00495	0.00506	0.00509

loops stands for the number of loop detectors. ϵ^* is the estimated diffusion coefficient. Note the true $\epsilon = 0.005$.
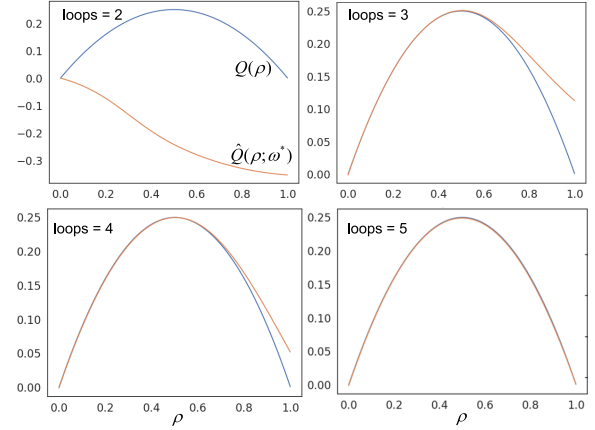

Fig. 4. The learned FD relation $\hat{Q}(\rho; \omega^*)$ based on PIDL + FDL architecture compared with the true $Q(\rho)$.

invariant, we fix the hyperparameter α to 100 and tune the other hyperparameters from [1, 10, 50, 100, 150, 200] with some follow-up fine tuning. The minimal-achievable estimation errors of PIDL + FDL over the numbers of loop detectors are presented in Table I. From the table, we can see that the traffic density estimation errors improve as the number of loop detectors increases. When more than two loop detectors are used, the model parameters to be learned are able to converge to the true parameters ϵ. Specifically, with three loop detectors, in addition to a good traffic density estimation error of 3.327×10^{-2}, the model parameter converges to $\epsilon^* = 0.00495$, which is very close to the true value 0.005. The results demonstrate that PIDL + FDL method can handle both TSE and model parameter identification with three loop detectors for the traffic dynamics of the Greenshields-based LWR.

The performance of FD estimation based on PIDL + FDL is presented is Fig. 4. Compared with the true $Q(\rho)$, the proposed method can reconstruct the exact Greenshields FD when 5 loop detectors are used. The results are meaningful because neither any assumptions on the FD relation are made nor the flow values are observed directly. In addition to traffic density estimation and model parameter identification, the PIDL + FDL is able to make full use of the conservation law and boundary conditions to retrieve the density-flow relation automatically.

V. PIDL + FDL FOR GREENSHIELDS-BASED ARZ

The second numerical example aims to show the capacity of the proposed method to handle the traffic dynamics governed by the Greenshields-based ARZ, a second-order traffic flow model with both traffic density ρ and velocity u as the traffic state variables.
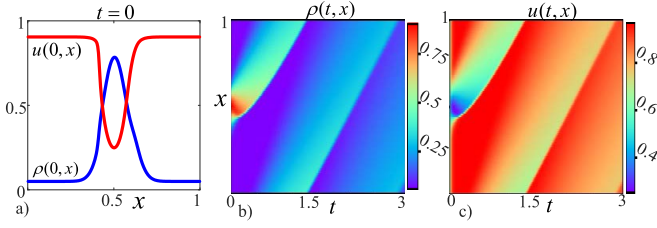
Fig. 5. a) is the bell-shaped initial ρ and u over $x \in [0, 1]$; b) and c) are numerical solutions for ρ and u, respectively.

An ARZ model involves both a conservation law of vehicles and a momentum equation on velocity. Specifically, we study the following traffic flow dynamics of a "ring road" in $t \in [0, 3]$, $x \in [0, 1]$:

$$\begin{aligned}
\rho_t + (\rho u)_x &= 0, \\
(u + h(\rho))_t + u(u + h(\rho))_x &= (U_{eq}(\rho) - u)/\tau, \\
h(\rho) &= U_{eq}(0) - U_{eq}(\rho) \quad (traffic\ pressure), \\
U_{eq}(\rho) &= u_{max}(1 - \rho/\rho_{max}) \quad (equilibrium\ speed), \\
\rho(t, 0) &= \rho(t, 1), \quad u(t, 0) = u(t, 1) \ (boundary\ cond.),
\end{aligned}$$
$$(13)$$

where we set the parameters irregularly as $\rho_{max} = 1.13$, $u_{max} = 1.02$, and $\tau = 0.02$. U_{eq} is the equilibrium velocity, $h(\rho)$ defines the traffic pressure and τ denotes the relaxation time scale. For more explanations of this ARZ setting, we refer readers to our previous work in [33].

Given the bell-shaped initial of ρ and u as shown in Fig. 5.a, we apply the Lax–Friedrichs (LF) scheme [41] to solve Eqs.(13) on grid G with 960 (time) $\times$ 240 (space) points evenly deployed over the $[0, 3] \times [0, 1]$ domain. The LF numerical solutions of both ρ and u over the domain are shown in Fig. 5 as well. We treat this numerical solution as the ground-truth to test our PIDL + FDL-based approach for the ARZ dynamics.

A. PIDL + FDL Architecture Design for ARZ

We employ a neural network $\hat{U}_{eq}(\cdot; \omega)$ to estimate the equilibrium velocity and $U_{eq}(\rho)$ is the target FD relation in this numerical example. Based on Eqs. (13), we define the following residuals

$$\begin{aligned}
\hat{f}_1(t, x; \theta) &:= \hat{\rho}_t + (\hat{\rho}\hat{u})_x, \\
\hat{f}_2(t, x; \theta, \omega, \tau) &:= (\hat{u} + h(\hat{\rho}))_t + \hat{u}(\hat{u} + h(\hat{\rho}))_x \\
&\quad - (\hat{U}_{eq}(\hat{\rho}; \omega) - \hat{u})/\tau,
\end{aligned}$$
$$(14)$$

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. By using the FD surrogate $\hat{U}_{eq}$, the relaxation τ is the only model parameter to be learned.

Given the definition of $(\hat{f}_1, \hat{f}_2)$, the design of PINN + FDL architecture is shown in Fig. 6. The structures of hidden layers of the PUNN and FDL are the same with those of Section IV. For this experiment, we adjust the learning loss as the following:

$$\begin{aligned}
Loss_{\theta, \omega, \tau} &= MSE_o + MSE_a + B1 \\
&= \frac{1}{N_o} \sum_{i=1}^{N_o} \alpha_1 |\hat{\rho}(t_o^{(i)}, x_o^{(i)}; \theta) - \rho^{(i)}|^2
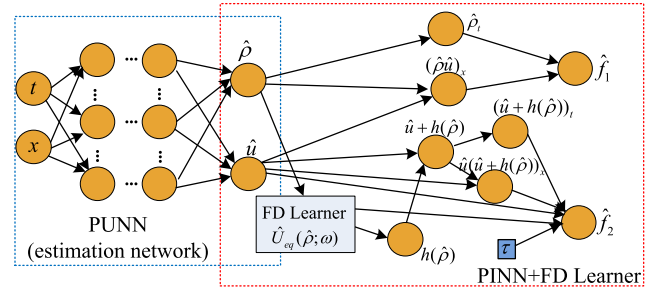\end{aligned}$$

Fig. 6. PIDL + FDL architecture for second-order traffic dynamics. The model parameter τ is held by a variable node (blue rectangular node).

TABLE II
PERFORMANCE OF PIDL + FDL ON THE ARZ DYNAMICS

loops	2	3	4	5
$Err(\hat{\rho}, \rho)$	0.5111	0.2249	0.04871	0.05243
$Err(\hat{u}, u)$	0.1586	0.05914	0.01402	0.01389
τ^*	0.02276	0.018994	0.019654	0.021619

loops stands for the number of loop detectors. τ^* is the estimated relaxation time. Note the true $\tau = 0.02$.

$$\begin{aligned}
&\quad + \alpha_2 |\hat{u}(t_o^{(i)}, x_o^{(i)}; \theta) - u^{(i)}|^2 \\
&\quad + \frac{1}{N_a} \sum_{j=1}^{N_a} \beta_1 |\hat{f}_1(t_a^{(j)}, x_a^{(j)}; \theta)|^2 \\
&\quad + \beta_2 |\hat{f}_2(t_a^{(j)}, x_a^{(j)}; \theta, \omega, \tau)|^2 \\
&\quad + \frac{1}{N_b} \sum_{k=1}^{N_b} (\gamma_1 |\hat{\rho}(t_b^{(k)}, 0; \theta) - \hat{\rho}(t_b^{(k)}, 1; \theta)|^2 \\
&\quad + \gamma_2 |\hat{u}(t_b^{(k)}, 0; \theta) - \hat{u}(t_b^{(k)}, 1; \theta)|^2).
\end{aligned}$$
$$(15)$$

We solve $(\theta^*, \omega^*, \tau^*) = \text{argmin}_{\theta, \omega, \tau} Loss_{\theta, \omega, \tau}$, and the results of traffic state estimation and model parameter identification are presented in Table II. Empirically, the difference between true and estimated values is visually indistinguishable when the errors are smaller than 6.00×10^{-2} and 2.90×10^{-2} for density and velocity, respectively. Performances with accuracy below these values are considered as "acceptable". From Table II, we can observe that the TSE performance of PIDL + FDL with more than three loops is acceptable.

The FDL results are shown in Fig. 7. As can be observed, the FDL performance improves as the number of loop detectors increases, and the proposed method with 4 loops and above is able to correctly learn the Greenshields FD relation.

VI. PIDL + FDL-BASED TSE ON NGSIM DATA

This section evaluates the PIDL + FDL-based TSE method using real-world traffic data, the Next Generation SIMulation (NGSIM) dataset,[1] and compares its performance to baselines.

A. NGSIM Dataset

NGSIM dataset contains real-world vehicle trajectories on several road scenarios. We focus on a segment of the US

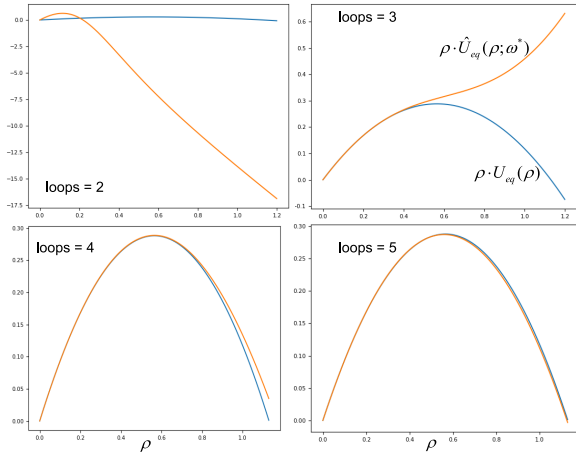[1] www.fhwa.dot.gov/publications/research/operations/07030/index.cfm

Fig. 7. The learned FD relation $\hat{U}_{eq}(\rho; \omega^*)$ based on PIDL + FDL architecture, compared with the ground-truth. To make the presentation consistent to Fig. 4, $\rho \cdot \hat{U}_{eq}(\rho; \omega^*)$ and $\rho \cdot U_{eq}$ are drawn.
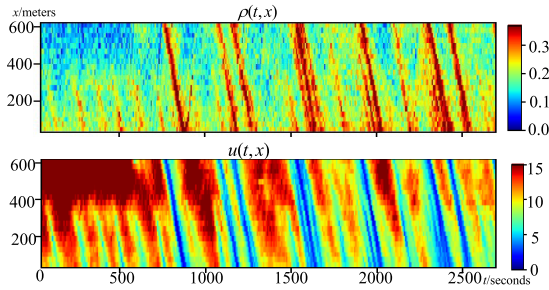

Fig. 8. Visualization of the traffic states on US 101 highway.

Highway 101 (US 101), monitored by a camera mounted on top of a high building on June 15, 2005. The locations and actions of each vehicle 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 many traffic flow studies [42].

We select the data from all the mainline lanes of the US 101 highway segment to calculate the average traffic density for approximately every 30 meters over a 1.5 seconds period. After preprocessing to remove the time when there are non-monitored vehicles running on the road (at the beginning and end of the video), there are 21 and 1770 valid cells on the spatial and temporal dimensions, respectively. We treat the center of each cell as a grid point. Fig. 8 shows the spatio-temporal field of traffic density $\rho(t, x)$ and velocity $u(t, x)$ in the dataset. From the figure, we can observe that shockwaves backpropagate along the highway.

For TSE experiments in this section, loop detectors are used to provide observed data with a recording frequency of 1.5 seconds. By default, they are evenly installed on the highway segment. We assume that the loop detectors are able to record the density and average velocity of cells on certain locations

B. TSE Methods for Real Data

We first introduce the PIDL + FDL-based methods for the real-world TSE problem, and then, describe advanced baselines to compare with.
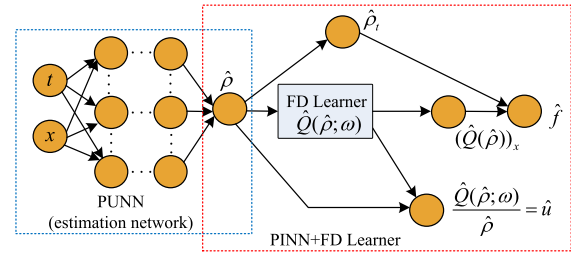

Fig. 9. The structure design of LWR-PIDL + FDL method for NGSIM data.

1) LWR-PIDL + FDL: This method is based on the PIDL + FDL encoded with the first-order LWR, using the structure in Fig. 1 except for two modifications: 1) no assumptions on diffusion effects are made and the traffic flow becomes $\rho_t + (Q(\rho))_x = 0$, which is commonly-used in literature [42]; and 2) the estimation of velocity is calculated using an additional calculation node $\hat{u} = \hat{Q}(\hat{\rho}; \omega)/\hat{\rho}$. Specifically, the modified structure of this TSE method is presented in Fig. 9. We select 80% of the grid G as the auxiliary points A. The loss in Eq. (11) (with the MSE on velocity estimation added and the boundary conditions removed) is used for training the PUNN and FDL using the observed data from loop detectors (i.e., both observation points O and corresponding target state values P). After tuning the hyperparameters with grid search, we present the minimal-achievable estimation errors. The same for other baselines by default. Because the real data could be noisy, leading to abnormal learned FD curves, the reshaping regularization term in Eq. (8) is applied.

2) ARZ-PIDL + FDL: This method is based on the PIDL + FDL encoded with the second-order ARZ, using the structure in Fig. 6. The Eq. (15) (with boundary conditions removed) is applied as the loss function. Other experimental setups are the same with those of the LWR-PIDL + FDL method.

3) Two-Dimensional Data Interpolation (Interp2): The two-dimensional linear interpolation method is used as a baseline, which interpolates the traffic states using the neighboring observed data in a linear manner.

4) Adaptive Smoothing (AS) Method: This method estimates the traffic state of a cell using the sum of all the observed data weighted by some smoothing kernel filters. We implement a generalized AS method proposed in [12] with the parameters suggested in [43] for the AS model.

5) Long Short Term Memory (LSTM) Based Method: This baseline method employs the LSTM architecture, which is customized from the LSTM-based TSE proposed by [23]. This model can be applied to our problem by leveraging the spatial dependency, i.e., to use the information of previous cells to estimate the traffic density and velocity of the next cell along the spatial dimension.

Other baselines include the **Pure Neural Network (NN)** and the **Extended Kalman Filter (EKF)** as well as the advanced PIDL-based TSE methods: **LWR-based PIDL (LWR-PIDL)** and **ARZ-based PIDL (ARZ-PIDL)**. For more descriptions regarding these four baselines, we refer the readers to the authors' previous work [33].
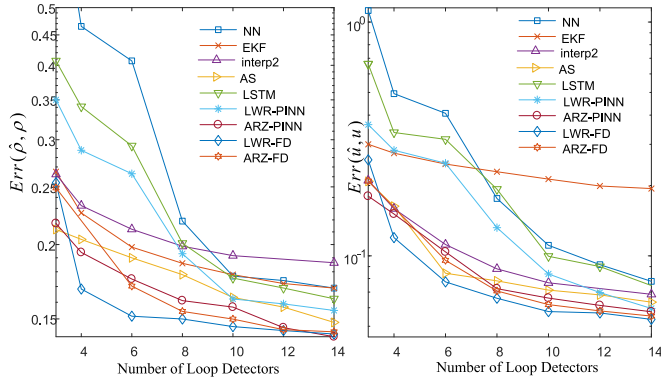
Fig. 10.   TSE performance of PIDL + FDL-based methods and baselines.

C. Results and Discussion

We apply PIDL + FDL-based, and baseline methods to TSE on the NGSIM dataset with different numbers of loop detectors. The results are presented in Fig. 10.

From Fig. 10, we can observe that the PIDL-related methods generally perform better than the model-driven and data-driven TSE baselines. The EKF/Interp2/AS methods achieve better errors than the NN/LSTM methods when the number of loop detectors are small, while the NN/LSTM methods' performance catches up when more loops are available.

The results are reasonable. The EKF is a model-driven approach, making sufficient use of the traffic flow model to appropriately estimate unobserved values when limited data are available. However, the model cannot fully capture the complicated traffic dynamics in the real world, and as a result, the EKF's performance flattens out. The Interp2 is a non-parametric data-driven method interpolating the unobserved fields using neighboring observation in a linear manner. The AS method incorporates the characteristic velocities of information propagation in free and congested traffic, by skewing the principal axes of the smoothing kernel. The Interp2 and AS methods have a relatively low complexity which can prevent over-fitting when the data is small. However, they may not effectively handle subtle state changes in the unobserved area due to the linearly-extrapolating nature or filtering nature, respectively. The PIDL-based method's errors are generally below the baselines, because it can make efficient use of both the traffic flow model and observed data. The ARZ-PIDL method is informed by a more advanced second-order traffic model and its performance is superior to that of LWR-PIDL.

The PIDL + FDL-based methods can generally achieve the best estimation accuracy and data efficiency over the above TSE baselines. The results demonstrate that the proper integration of the NN-based FD surrogate to the PIDL can give the learning framework more flexibility to achieve an improved TSE accuracy. One interesting phenomenon is that the PIDL + FDL with the first-order LWR (LWR-PIDL + FDL) can beat the one with a more sophisticated second-order ARZ model (ARZ-PIDL + FDL). This observation supports our discussion that sophisticated traffic models may not always lead to a better TSE performance, because the model may contain complicated terms that makes the TSE performance sensitive
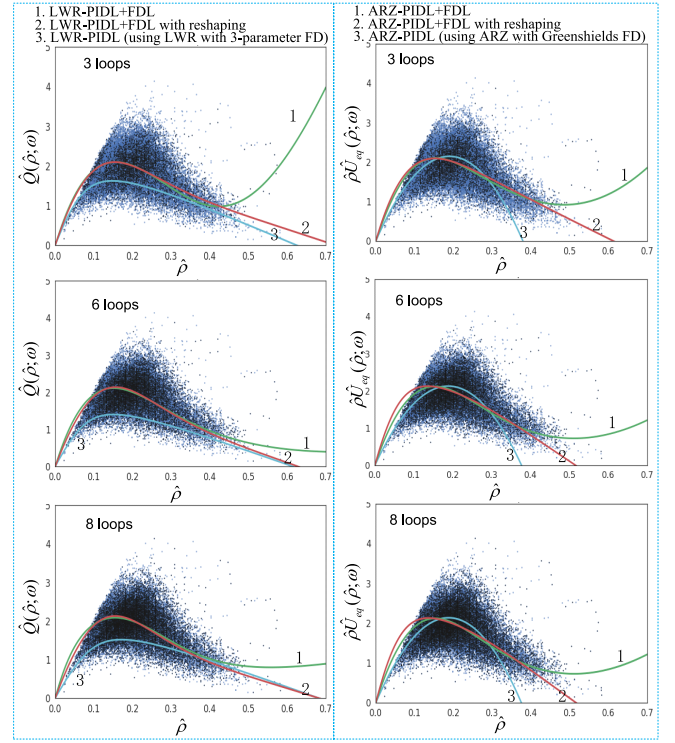

Fig. 11.   Comparison of the FD curves estimated by the PIDL + FDL-based TSE methods and PIDL-based TSE methods. For consistent visualization, the learned $\hat{U}_{eq}(\hat{\rho}; \omega)$ is converted to $\hat{\rho}\hat{U}_{eq}(\hat{\rho}; \omega)$ to represent the estimated flow.

to the PINN structural design, and thus, the model becomes difficult to train. Compared to the ARZ-PIDL + FDL, the LWR-PIDL + FDL can balance the trade-off between the sophisticated level of PINN and the training flexibility more properly, making it a better fit to the NGSIM scenario.

D. Discussions on Fundamental Diagram Estimation

The PIDL + FDL-based methods can further learn the hidden fundamental diagram (FD) relation. We compare the FD curves learned via the PIDL + FDL-based methods and PIDL-based methods in the density-flow space when small number of loops are available. The results are presented in Fig. 11 where each dark blue dot is a density-flow data point in the NGSIM dataset. Note, the flow values are not part of the observed data during the training phase.

For the PIDL-based methods, the closed form of the flux function and velocity function are given and the parameters in the PINN component are learned along with the TSE training, and thus, the shape of the FD curves are predefined. The corresponding FD curves with the learned model parameters are indexed as "3" in Fig. 11. For consistent visualization, the learned $\hat{U}_{eq}(\hat{\rho}; \omega)$ is converted to $\hat{\rho}\hat{U}_{eq}(\hat{\rho}; \omega)$ to represent the estimated flow. The LWR-PIDL method is encoded with the 3-parameter-based flux, and the ARZ-PIDL is with the Greenshields function for the equilibrium velocity. The former has a proper shape defined by the given mathematical formula, but due to the complicated nature of the PINN for the 3-parameter flux and the noisy quality of the data, the learned FD curves do not fit the density-flow points to

a satisfactory extent. The latter has a predefined quadratic shape and can capture the density-flow characteristics to a limited level.

For the PIDL + FDL-based methods, using the standard training loss in Eq.(7), the learned FD shapes (indexed as "1" in Fig. 11) fit the NGSIM density-flow points well over the domain where the observed traffic state data are distributed, i.e., around $\rho \in [0, 0.5]$. However, the FD curves tend to curl up in the large-density domain, where the data are sparse. To address this abnormal shape, we apply the regularization term in Eq. (8) for reshaping and impose the prior knowledge of concavity over a narrowed interval of $\rho \in [0.6, 0.7]$. To this end, we set the hyperparameters in Eq. (8) to $a = 0.6$ and $b = 0.7$. The corresponding learned FD curves using the reshaping regularization term are indexed as "2" in the figures, and they can properly capture the density-flow characteristics to a satisfactory level. Because of using the FD Learner, the LWR-PIDL + FDL contains no model parameters, and the conservation law plus the $\hat{Q}(\rho; \omega^*)$ constitutes the LWR model reconstructed by the LWR-PIDL + FDL. The ARZ-PIDL + FDL contains one model parameter, i.e., the relaxation time τ. The learning with data from 3, 6 and 8 loops converges to $\tau^* = [23.36, 25.99, 27.66]$, which is reasonably close to $\tau = [27.6, 28.8, 30.5]$ directly fitted from data. The conservation law, the momentum of velocity with model parameter τ^*, and the learned $\hat{U}_{eq}(\rho; \omega^*)$ constitute the ARZ model reconstructed by the ARZ-PIDL + FDL.

The experimental results demonstrate that the proposed PIDL + FDL-based TSE method (with the regularization for reshaping) is able to efficiently conduct high-quality TSE, model parameter identification and fundamental diagram estimation at the same time with relatively small amounts of observed data.

VII. CONCLUSION

We introduced the PIDL + FDL framework to the TSE problem on highways using loop detector data and demonstrate the significant benefits of the integration of an ML surrogate into the model-driven component in PIDL. This framework can be used to handle traffic state estimation, model parameter identification, and fundamental diagram estimation simultaneously. The experiments on real highway data show that PIDL + FDL-based approaches can outperform baselines in terms of estimation accuracy and data efficiency as well as the estimation of FD.

The limitations and potential future works of this paper are as follows: (1) Similar to most deep learning methods, hyperparameter tuning is an issue of PIDL-based TSE, and tuning such a large number of hyperparameters based on approaches like cross-validation is too complicated for real-world application, and the model basically has to be tuned for each scenario (spatial resolution, temporal resolution, spatial grid size, temporal grid size, observation error, etc), which limits its applicability to real-world problems; (2) PIDL is known to have issues with noisy data, and how the high noise and corruption in the real traffic data undermine the performance of PIDL-based TSE needs further investigations; (3) It is worthy of considering more ML surrogate components

to represent other unobserved traffic quantities in the traffic flow model, such as $h(\rho)$ and $U_{eq}(\rho - u)/\tau$, and study to what extent the addition of surrogates affects the performance.

REFERENCES

[1] T. Seo, A. M. Bayen, T. Kusakabe, and Y. Asakura, "Traffic state estimation on highway: A comprehensive survey," *Annu. Rev. Control*, vol. 43, pp. 128–151, Jan. 2017.

[2] X. Di and R. Shi, "A survey on autonomous vehicle control in the era of mixed-autonomy: From physics-based to AI-guided driving policy learning," *Transp. Res. C, Emerg. Technol.*, vol. 125, Apr. 2021, Art. no. 103008.

[3] M. J. Lighthill and G. B. Whitham, "On kinematic waves II. A theory of traffic flow on long crowded roads," *Proc. Roy. Soc. A, Math., Phys. Eng. Sci.*, vol. 229, no. 1178, pp. 317–345, 1955.

[4] P. I. Richards, "Shock waves on the highway," *Oper. Res.*, vol. 4, no. 1, pp. 42–51, 1956.

[5] A. Aw and M. Rascle, "Resurrection of 'second order' models of traffic flow," *SIAM J. Appl. Math.*, vol. 60, no. 3, pp. 916–938, 2000.

[6] H. M. Zhang, "A non-equilibrium traffic model devoid of gas-like behavior," *Transp. Res. B, Methodol.*, vol. 36, no. 3, pp. 275–290, 2002.

[7] M. Garavello, K. Han, and B. Piccoli, *Models for Vehicular Traffic on Networks*, vol. 9. Springfield, MO, USA: American Institute of Mathematical Sciences, 2016.

[8] Y. Wang, M. Papageorgiou, and A. Messmer, "Real-time freeway traffic state estimation based on extended Kalman filter: Adaptive capabilities and real data testing," *Transp. Res. A, Policy Pract.*, vol. 42, no. 10, pp. 1340–1358, Dec. 2008.

[9] Y. Wang, M. Papageorgiou, A. Messmer, P. Coppola, A. Tzimitsi, and A. Nuzzolo, "An adaptive freeway traffic state estimator," *Automatica*, vol. 45, no. 1, pp. 10–24, 2009.

[10] X. Di, H. X. Liu, and G. A. Davis, "Hybrid extended Kalman filtering approach for traffic density estimation along signalized arterials: Use of global positioning system data," *Transp. Res. Rec.*, vol. 2188, no. 1, pp. 165–173, Jan. 2010.

[11] L. Mihaylova and R. Boel, "A particle filter for freeway traffic estimation," in *Proc. 43rd IEEE Conf. Decis. Control (CDC)*, Nassau, Bahamas, Dec. 2004, pp. 2106–2111.

[12] M. Treiber, A. Kesting, and R. E. Wilson, "Reconstructing the traffic state by fusion of heterogeneous data," *Comput. Aided Civil Infrastruct. Eng.*, vol. 26, no. 6, pp. 408–419, Aug. 2011.

[13] S. Blandin, J. Argote, A. M. Bayen, and D. B. Work, "Phase transition model of non-stationary traffic flow: Definition, properties and solution method," *Transp. Res. B, Methodol.*, vol. 52, pp. 31–55, Jun. 2013.

[14] S. Fan *et al.*, "Comparative model accuracy of a data-fitted generalized Aw-Rascle-Zhang model," *Netw. Heterogeneous Media*, vol. 9, no. 2, pp. 239–268, 2014.

[15] B. Smith, W. Scherer, and J. Conklin, "Exploring imputation techniques for missing data in transportation management systems," *Transp. Res. Rec.*, vol. 1836, pp. 132–142, Jan. 2003.

[16] C. Chen, J. Kwon, J. Rice, A. Skabardonis, and P. Varaiya, "Detecting errors and imputing missing data for single-loop surveillance systems," *Transp. Res. Rec.*, vol. 1855, no. 1, pp. 160–167, 2003.

[17] M. Zhong, P. Lingras, and S. Sharma, "Estimation of missing traffic counts using factor, genetic, neural, and regression techniques," *Transp. Res. C, Emerg. Technol.*, vol. 12, no. 2, pp. 139–166, 2004.

[18] L. Li, Y. Li, and Z. Li, "Efficient missing data imputing for traffic flow by considering temporal and spatial dependence," *Transp. Res. C, Emerg. Technol.*, vol. 34, pp. 108–120, Sep. 2013.

[19] H. Tan, Y. Wu, B. Cheng, W. Wang, and B. Ran, "Robust missing traffic flow imputation considering nonnegativity and road capacity," *Math. Problems Eng.*, vol. 2014, pp. 1–8, Jan. 2014.

[20] S. Tak, S. Woo, and H. Yeo, "Data-driven imputation method for traffic data in sectional units of road links," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 6, pp. 1762–1771, Jun. 2016.

[21] D. Ni and J. Leonard, II, "Markov chain Monte Carlo multiple imputation using Bayesian networks for incomplete intelligent transportation systems data," *Transp. Res. Rec.*, vol. 1935, no. 1, pp. 57–67, 2005.

[22] N. G. Polson and V. O. Sokolov, "Deep learning for short-term traffic flow prediction," *Transp. Res. C, Emerg. Technol.*, vol. 79, pp. 1–17, Jun. 2017.

[23] W. Li, J. Hu, Z. Zhang, and Y. Zhang, "A novel traffic flow data imputation method for traffic state identification and prediction based on spatio-temporal transportation big data," in *Proc. Int. Conf. Transp. Professionals (CICTP)*, Shanghai, China, 2017, pp. 79–88.

[24] Z. Zheng, Y. Yang, J. Liu, H.-N. Dai, and Y. Zhang, "Deep and embedded learning approach for traffic flow prediction in urban informatics," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 10, pp. 3927–3939, Oct. 2019.

[25] Y. Yuan, Z. Zhang, X. T. Yang, and S. Zhe, "Macroscopic traffic flow modeling with physics regularized Gaussian process: A new insight into machine learning applications in transportation," *Transp. Res. B, Methodol.*, vol. 146, pp. 88–110, Apr. 2021.

[26] Z. Wang, W. Xing, R. Kirby, and S. Zhe, "Physics regularized Gaussian processes," 2020, *arXiv:2006.04976*. [Online]. Available: http://arxiv.org/abs/2006.04976

[27] M. Raissi and G. E. Karniadakis, "Hidden physics models: Machine learning of nonlinear partial differential equations," *J. Comput. Phys.*, vol. 357, pp. 125–141, Mar. 2018.

[28] Z. Mo, R. Shi, and X. Di, "A physics-informed deep learning paradigm for car-following models," *Transp. Res. C, Emerg. Technol.*, vol. 130, Sep. 2021, Art. no. 103240.

[29] J. Huang and S. Agarwal, "Physics informed deep learning for traffic state estimation," in *Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst. (ITSC)*, Sep. 2020, pp. 3487–3492.

[30] M. Barreau, J. Liu, and K. H. Johansson, "Learning-based state reconstruction for a scalar hyperbolic PDE under noisy Lagrangian sensing," Nov. 2020, *arXiv:2011.09871*. [Online]. Available: http://arxiv.org/abs/2011.09871

[31] J. Liu, M. Barreau, M. Cicic, and K. H. Johansson, "Learning-based traffic state reconstruction using probe vehicles," Nov. 2020, *arXiv:2011.05031*. [Online]. Available: http://arxiv.org/abs/2011.05031

[32] M. Barreau, M. Aguiar, J. Liu, and K. H. Johansson, "Physics-informed learning for identification and state reconstruction of traffic density," Mar. 2021, *arXiv:2103.13852*. [Online]. Available: http://arxiv.org/abs/2103.13852

[33] R. Shi, Z. Mo, and X. Di, "Physics-informed deep learning for traffic state estimation: A hybrid paradigm informed by second-order traffic models," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, May 2021, vol. 35, no. 1, pp. 540–547.

[34] B. T. Thodi, Z. S. Khan, S. E. Jabari, and M. Menendez, "Incorporating kinematic wave theory into a deep learning method for high-resolution traffic speed estimation," Feb. 2021, *arXiv:2102.02906*. [Online]. Available: http://arxiv.org/abs/2102.02906

[35] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, "Automatic differentiation in machine learning: A survey," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 5595–5637, 2017.

[36] B. Greenshields, W. Channing, and H. Miller, "A study of traffic capacity," *Proc. Highway Res. Board*, vol. 14, pp. 448–477, 1935.

[37] S. K. Godunov, "A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics," *Matematicheskii Sbornik*, vol. 89, no. 3, pp. 271–306, 1959.

[38] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 30th Int. Conf. Artif. Intell. Statist. (AISTATS)*, Chia Laguna, Sardinia, Italy, 2010, pp. 249–256.

[39] O. Konur, D. P. Kingma, and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, San Diego, CA, USA, 2015, pp. 1–15.

[40] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, "A limited memory algorithm for bound constrained optimization," *SIAM J. Sci. Comput.*, vol. 16, no. 5, pp. 1190–1208, 1995.

[41] R. J. LeVeque, *Finite Volume Methods for Hyperbolic Problems*. Cambridge, U.K.: Cambridge Univ. Press, 2002, vol. 31.

[42] S. Fan and B. Seibold, "Data-fitted first-order traffic models and their second-order generalizations: Comparison by trajectory and sensor data," *Transp. Res. Rec.*, vol. 2391, no. 1, pp. 32–43, Jan. 2013.

[43] X. Wang, Y. Wu, D. Zhuang, and L. Sun, "Low-rank Hankel tensor completion for traffic speed estimation," 2021, *arXiv:2105.11335*. [Online]. Available: http://arxiv.org/abs/2105.11335
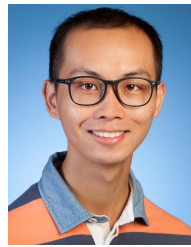
Rongye Shi (Member, IEEE) received the Ph.D. degree in electrical and computer engineering from Carnegie Mellon University in 2019.

He was a Research Scientist with the Department of Civil Engineering and Engineering Mechanics, Columbia University. His research is in the area of data science, machine learning, multi-agent planning, and their applications to intelligent transportation systems. He is a member of ACM.

Zhaobin Mo received the B.S. degree in automotive engineering from Tsinghua University, Beijing, China, in 2017. He is currently pursuing the Ph.D. degree with the Department of Civil Engineering and Engineering Mechanics, Columbia University.

His research interests include deep learning, reinforcement learning, and connected and automated vehicle.

Kuang Huang received the B.S. degree in mathematics from Wuhan University, Wuhan, China, in 2017, and the M.S. degree in applied mathematics from Columbia University in 2019, where he is currently pursuing the Ph.D. degree with the Department of Applied Physics and Applied Mathematics.

His research areas include traffic flow modeling, numerical methods for conservation laws, mean field games, and nonlocal partial differential equations.

Xuan Di (Member, IEEE) received the Ph.D. degree from the University of Minnesota at Twin Cities in 2014.

She is currently an Associate Professor with the Department of Civil Engineering and Engineering Mechanics and a Committee Member of the Smart Cities Center, Data Science Institute, Columbia University. Her research interests include data-driven urban mobility modeling and shared mobility, leveraging optimization, game theory, and data analytics. She is a member of ACM.

Qiang Du received the Ph.D. degree from Carnegie Mellon University in 1988.

He is currently the Fu Foundation Professor of applied mathematics with the Department of Applied Physics and Applied Mathematics, Columbia University. He is also a co-chair of the Center for Computing Systems for Data-Driven Science, Data Science Institute, Columbia University. His research interests are in numerical analysis, mathematical modeling, and scientific computation with applications in physical and data sciences. He is a fellow of SIAM, AMS, and AAAS.