A PINN approach for traffic state estimation and model calibration based on loop detector flow data

Paola Goatin, Daniel Inzunza

A PINN approach for traffic state estimation and model calibration based on loop detector flow data

Daniel Inzunza
Université Côte d'Azur
Inria, CNRS, LJAD
Sophia Antipolis, France
daniel-eduardo.inzunza-herrera@inria.fr

Paola Goatin
Université Côte d'Azur
Inria, CNRS, LJAD
Sophia Antipolis, France
paola.goatin@inria.fr

***Abstract*—We analyze the performances of a Physics Informed Neural Network (PINN) strategy applied to traffic state estimation and model parameter identification in realistic situations. The traffic dynamics is modeled by a first order macroscopic traffic flow model involving two physical parameters and an auxiliary one. Besides, observations consist of (averaged) density and flow synthetic data computed at fixed space locations, simulating real loop detector measurements. We show that the proposed approach is able to give a good approximation of the underlying dynamics even with poorer information. Moreover, the precision generally improves as the number of measurement locations increases.**

***Index Terms*—macroscopic traffic flow models, state estimation, model parameter identification, physics-informed deep learning**

I. Introduction

Physics-informed neural networks (PINNs) have recently been introduced as an alternative method of solving nonlinear differential equations [1], [2]. Compared to purely physics- or purely data-driven models, this approach allows to combine data and physical constraints in the same computing framework. They can also be seen as surrogate models for solving differential equations including additional data information, or inverse problems for parameter identification. This methodology has been applied to a variety of fields, ranging from fluid-dynamics [3] and quantum mechanics [1] to epidemiology [4].

In this paper, inspired by the results obtained in [5] for macroscopic traffic model discovery and state estimation, we propose to use PINNs for calibrating the celebrated Lighthill-Whitham-Richards (LWR) model [6], [7] against synthetic loop-detector flow measurements, mimicking a real-world setting. Indeed, previous analysis conducted in [5], [8], [9], is based on (sparse) traffic density data, which are usually not directly provided by common traffic measurements, such as loop detector average flow, velocity or occupancy observed at fixed locations, or floating car speed data provided by GPS devices. In our study, we integrate more realistic flux measurements at fixed locations along the road, and we evaluate the performances of PINN approach in reconstructing the numerical solution that originated the synthetic data and in calibrating the corresponding model parameters, depending on the number of loop detectors and the averaging intervals. Of course, flux data are less informative than density one, since they are unable to distinguish between free flow and congested situations (the same flux value can correspond to two different densities). This results often in a poorer reconstruction of initial conditions, which can be corrected by increasing the number of observation points. Nevertheless, the overall performances are satisfying, and encourage a larger study, including real data and the comparison with other calibration techniques, see e.g. [10]–[15].

The article is organized as follows. Section II recalls the basis of the PINN approach for traffic state estimation and model parameter calibration. Section III specifies the procedure we follow for the LWR model. Experimental results are presented in Section IV and a conclusion is drawn in Section V.

II. PINNs for macroscopic traffic models

Below, we recall the main principles of the PINN strategy for traffic state reconstruction.

A. State estimation

We consider a physical model in the form of a nonlinear differential equation for the evolution of the traffic state $u = u(t,x)$ on a road segment

$$\partial_t u(t,x) + \mathcal{N}[u(t,x)] = 0, \quad x \in [0,L],\ t \in [0,T], \quad (1)$$

where $L, T \in \mathbb{R}^+$ and $\mathcal{N}$ is a general nonlinear space differential operator. We aim at reconstructing the solution u, while matching at best some available measurements.

This work has been supported by the French government, through the 3IA Côte d'Azur Investments in the Future project managed by the National Research Agency (ANR) with the reference number ANR-19-P3IA-0002.

978-1-6654-5530-5/23/$31.00 ©2023 IEEE

We discretize the time-space continuous domain $D = [0,T] \times [0,L]$ defining a grid of points evenly distributed, which will be denoted by

$$G = \{(t^{(r)}, x^{(r)}) \,|\, r = 1, \dots, N_g\}.$$

The PINN algorithm approximates $u(t,x)$ using neural networks with time t, location x and a set of data related to the state u as inputs. We denote this approximation of $u(t,x)$ as $\hat{u}(t,x)$. During the learning phase, the algorithm includes the minimization of the modeling error, given by the following residual value of the approximation:

$$\hat{f}(t,x) := \partial_t \hat{u} + \mathcal{N}\left[\hat{u}(t,x)\right], \tag{2}$$

which expresses the discrepancy with the solution of (1): the closer $\hat{u}$ is to u, the closer the residual will be to zero. In our code, following [1], $\hat{f}$ is calculated by automatic differentiation technique, using the function `tf.gradient` of Tensorflow[1]. Note that this approach is very sensitive to the presence of discontinuities in the solution, as in the case of traffic models. This limitation needs to be handled with specific corrections, see e.g. [3], [16], [17]. For simplicity, here we make the choice of regularizing the modeling equation by artificial viscosity.

The training data for the PINN algorithm consist of observation points O, target values P and auxiliary points A given by

$$\begin{aligned} O &= \{(t_o^{(i)}, x_o^{(i)}) \,|\, i = 1, \dots, N_o\} \subset G, \\ P &= \{U^{(i)} \,|\, i = 1, \dots, N_o\}, \\ A &= \{(t_a^{(j)}, x_a^{(j)}) \,|\, j = 1, \dots, N_a\} \subset G. \end{aligned} \tag{3}$$

Note that $\{O, P\}$ defines the observed data, where P are the measured traffic states at the observations points, which do not necessarily coincide with the conserved traffic quantities u, as in the case of speed or flow measurements.

To train the PINN for the considered traffic model (1), we define the loss function as follows:

$$\begin{aligned} \text{Loss} &= \omega_o \cdot \text{MSE}_o + \omega_a \cdot \text{MSE}_a \\ &= \frac{\omega_o}{N_o} \sum_{i=1}^{N_o} |\varphi(\hat{u}(t_o^{(i)}, x_o^{(i)})) - U^{(i)}|^2 + \frac{\omega_a}{N_a} \sum_{j=1}^{N_a} |\hat{f}(t_a^{(j)}, x_a^{(j)})|^2, \end{aligned}$$

where ω_o and ω_a are weights to be defined for balancing the contribution to the loss made by the data and physical discrepancies, respectively. In MSE_o, the function φ maps the traffic conserved quantities in the measured ones.

B. Model parameter calibration

In addition to estimating the traffic state with the known PDE traffic model, PINN approach can handle traffic models with unknown parameters, allowing to identify the parameter values that best describe the observed data.

In the case of unknown parameters λ, (1) can be rewritten as

$$\partial_t u(t,x) + \mathcal{N}\left[u(t,x); \lambda\right] = 0, \quad x \in [0,L],\ t \in [0,T], \tag{4}$$

where $\mathcal{N}\left[\cdot; \lambda\right]$ denotes the parameterized nonlinear differential operator.

The goal is now to find λ^* and $\hat{u} = \hat{u}(\cdot, \cdot; \lambda^*)$ that best describes the observed data while approximating the model solution $u(t,x;\lambda^*)$ at points in G. For this, the residual value of traffic state approximation $\hat{u}(t,x;\lambda)$ from PINN takes the form

$$\hat{f}(t,x;\lambda) := \partial_t \hat{u}(t,x) + \mathcal{N}\left[\hat{u}(t,x); \lambda\right]. \tag{5}$$

Notice that, by the above residual, the PINN reconstruction $\hat{u}$ itself depends on λ. The way in which training data are obtained and distributed remains the same as in the previous section. Then the loss function for parameter calibration and state reconstruction is defined as:

$$\begin{aligned} \text{Loss}(\lambda) &= \omega_o \cdot \text{MSE}_o(\lambda) + \omega_a \cdot \text{MSE}_a(\lambda) \\ &= \frac{\omega_o}{N_o} \sum_{i=1}^{N_o} |\varphi(\hat{u}(t_o^{(i)}, x_o^{(i)}); \lambda) - U^{(i)}|^2 \\ &\quad + \frac{\omega_a}{N_a} \sum_{j=1}^{N_a} |\hat{f}(t_a^{(j)}, x_a^{(j)}; \lambda)|^2. \end{aligned}$$

Notice that, in general, also the mapping φ, and therefore MSE_o, may depend on the model parameter λ, as will be the case in Sections IV-C and IV-D.

Given the training data, we apply neural network training algorithms to solve

$$\lambda^* = \text{argmin}_\lambda \text{Loss}(\lambda).$$

Then, the λ^*-parameterized traffic flow model (4) is the most likely physics that generates the observed data.

III. PINN FOR THE LWR MODEL

The LWR model represents the paradigm of first order macroscopic traffic flow models. It consists in the scalar conservation law

$$\partial_t \rho + \partial_x Q(\rho; \theta) = 0, \tag{6}$$

which expresses the conservation of the number of vehicles. In (6), $\rho = \rho(t,x)$ denotes the traffic density (number of vehicles per unit length) and the flow rate $Q = Q(\rho;\theta)$ (number of vehicles per unit time) prescribes a functional relationship between the density and the flux, depending on some model parameter $\theta \in \mathbb{R}^m$, $m \geq 1$, specific of the road section under consideration.

Since solutions of (6) are generally discontinuous, resulting in poor performance of classical PINN algorithms employing automatic differentiation, we will consider the viscous approximation of LWR model including a diffusive correction

$$\partial_t \rho + \partial_x Q(\rho; \theta) = \epsilon\, \partial_{xx} \rho, \tag{7}$$

where $\epsilon > 0$ is the (small) diffusion coefficient.

[1] https://www.tensorflow.org/

For our study, we consider synthetic data generated by numerical approximation of the following Cauchy problem on an interval with periodic boundary conditions

$$\begin{cases} \partial_t \rho + \partial_x Q(\rho;\theta) = \epsilon \partial_{xx}\rho, \\ \rho(0,x) = \rho_0(x), \\ \rho(t,0) = \rho(t,1), \end{cases} \qquad x \in [0,1]\,,\ t \in [0,T]\,, \quad (8)$$

where $\theta = (V, R)$ and

$$Q(\rho;\theta) = V\rho\left(1 - \frac{\rho}{R}\right).$$

Above, V denotes the maximal speed and R the maximal density corresponding to a bump-to-bump situation, for which the average traffic speed is zero.

Given an initial condition $\rho_0 = \rho_0(x)$, we apply the Godunov scheme [18] to solve (8) on $M \times N$ (space $\times$ time) grid points G evenly deployed throughout the $[0,1]\times[0,T]$ domain. Thus, the total number of grid points G is $N_g = M \times N$.

Based on (7), we define the residual value of PINN's traffic density estimation $\hat{\rho}(t,x)$ as

$$\hat{f}(t,x;\lambda) := \partial_t \hat{\rho}(t,x) + \partial_x Q(\hat{\rho}(t,x);\theta) - \epsilon \partial_{xx}\hat{\rho}(t,x), \quad (9)$$

where we have set $\lambda := (\theta, \epsilon)$.

For the tests, we define the following loss functions:

$$Case\,A: \quad \text{Loss}_1 = \omega_o\,\text{MSE}_o + \omega_a\,\text{MSE}_a\,, \quad (10a)$$

$$Case\,B: \quad \text{Loss}_2 = \omega_q\,\text{MSE}_q + \omega_a\,\text{MSE}_a\,, \quad (10b)$$

where

$$\text{MSE}_o = \frac{1}{N_o}\sum_{i=1}^{N_o} |\hat{\rho}(t_o^{(i)}, x_o^{(i)}) - \rho^{(i)}|^2, \quad (11a)$$

$$\text{MSE}_a(\lambda) = \frac{1}{N_a}\sum_{j=1}^{N_a} |\hat{f}(t_a^{(j)}, x_a^{(j)};\lambda)|^2, \quad (11b)$$

$$\text{MSE}_q(\lambda) = \frac{1}{N_o}\sum_{i=1}^{N_o} |Q(\hat{\rho}(t_o^{(i)}, x_o^{(i)});\theta) - Q^{(i)}|^2, \quad (11c)$$

and the weight values ω_o, ω_a and ω_q are to be specified. In (11a) and (11c), $\rho^{(i)}$ and $Q^{(i)}$ denote the density and flow observations, respectively.

IV. Experiments

We apply the PINN method to compute the traffic state estimation, using as input data the information obtained from loop detectors at fixed locations. In particular, we will analyze how the performance evolves as the number of loops increases.

Our neural network is fully connected feed-forward consisting of 8 hidden layers with 20 neurons in each layer, with $\tanh$ as the activation function. The activation functions and the connecting structure of neurons in the PINN are designed to conduct the differential operations in (2), see [5, Fig. 1] for a visual representation of the employed PINN architecture. The optimizing procedure consists of the Adam optimizer with an initial learning rate 0.001 for 20 000 steps, followed by the L-BFGS-B optimizer.

TABLE I
Parameters used in our experiments

Layers and neurons	[2,20,20,20,20,20,20,20,20,1]
Activation function	$\tanh$
Learning rate	0.001
Optimizer	Adam + L-BFGS-B
Number of iterations (Adam)	20.000
Maximum number of iterations (L-BFGS-B)	50.000
Number of observation points	$N_0 = 2880 \times m_{\text{loop}}$
Number of observation points (averaged)	$N_0 = 40 \times m_{\text{loop}}$
Number of auxiliary points	$N_a = 20.000$
Maximal speed (label value)	$V = 1$
Maximal density (label value)	$R = 1$
Diffusive correction (label value)	$\epsilon = 0.005$

As in [5], [9], we construct the reference solution ρ and the data-sets by Godunov numerical approximation of (8) on the time interval $[0,3]$ with initial datum $\rho_0(x) = 0.1 + 0.8\exp(-25(x-0.5)^2)$, shown in Fig. 1, on a grid of $N_g = 2880 \times 240$ points, setting

$$\lambda = (V, R, \epsilon) = (1, 1, 0.005),$$

see Fig 2. The data-sets are constructed by extracting the computed density and flow values at given space coordinates $\bar{x}_\ell$, $\ell = 1, \ldots, m_{\text{loop}}$, corresponding to the fictive loop positions, for all time steps $t^n = n\Delta t$, $n = 1, \ldots, 2880$, eventually averaged on time sub-intervals of length 0.075. In all the experiments, the residual is evaluated at $N_a = 20\,000$ auxiliary points randomly selected. Moreover, we set the weights in (10) to $\omega_o = \omega_a = \omega_q = 1$ in all tests. The above information is summarized in Table I.

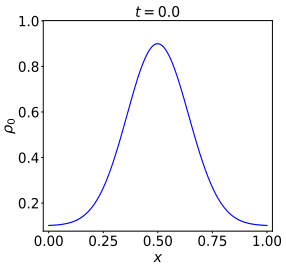

Fig. 1. Initial traffic density $\rho_0(x) = 0.1 + 0.8\exp(-25(x-0.5)^2)$.

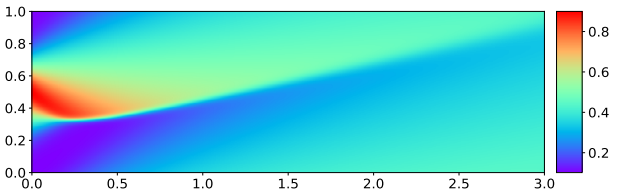

Fig. 2. Reference solution of (8) corresponding to $\lambda = (V, R, \epsilon) = (1, 1, 0.005)$ computed by Godunov's scheme.

As a measure of performance, we compute the $\mathbf{L}^2$ relative

error on G

$$\mathrm{Err}(\hat{\rho},\rho)=\sqrt{\frac{\sum_{r=1}^{N_g}|\hat{\rho}(t^{(r)},x^{(r)})-\rho(t^{(r)},x^{(r)})|^2}{\sum_{r=1}^{N_g}|\rho(t^{(r)},x^{(r)})|^2}}$$

to quantify the estimation error on the entire domain.

We detail below the different tests we carried out. In all the situations, we compare the performances of the PINN approach using density and flow data, respectively. Indeed, the analysis carried out in [5], [8], [9] relied on (synthetic) density data measured at fixed positions. Nevertheless, loop detectors primarily measure flow rates, giving accurate information on the number of vehicles passing at their locations. Moreover, in many cases, raw data are not available, and only aggregated quantities are recorded, resulting in less precise information.

A. State estimation using full loop detector observation

We consider $N_0 = 2880 \times m_{\text{loop}}$ observation points, where $m_{\text{loop}} = 3,\dots,9$ denotes the number of equally spaced loop detectors considered, and their corresponding target density/flow values P (setting $\varphi(u)=u$ and $\varphi(u)=Q(u;(1,1))$ respectively). The results are presented in Table II and Figures 3 and 4. As expected, flow data give worse results, but the gap reduces strongly increasing the number of loop detectors.

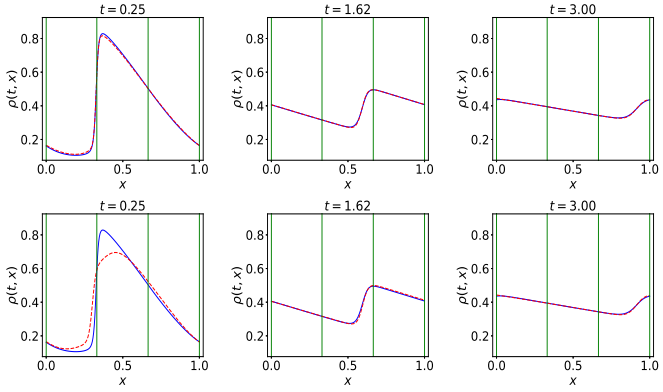

Fig. 3. Comparison of the predicted and exact solutions corresponding to three temporal snapshots for $m_{\text{loop}} = 4$, obtained with state reconstruction using full loop detector observation for Case A (top) and B (bottom).

TABLE II
ERROR ON ESTIMATED TRAFFIC DENSITY FOR CASES A AND B USING FULL DATA INFORMATION

m_{loop}	Error Case A	Error Case B	Relative Error
3	1.172e-02	3.090e-01	25.356
4	9.271e-03	6.979e-02	6.528
5	1.887e-02	2.971e-02	0.575
6	6.767e-03	1.610e-02	1.379
7	9.302e-03	1.120e-02	0.204
8	6.454e-03	7.850e-02	11.164
9	8.378e-03	8.632e-03	0.030

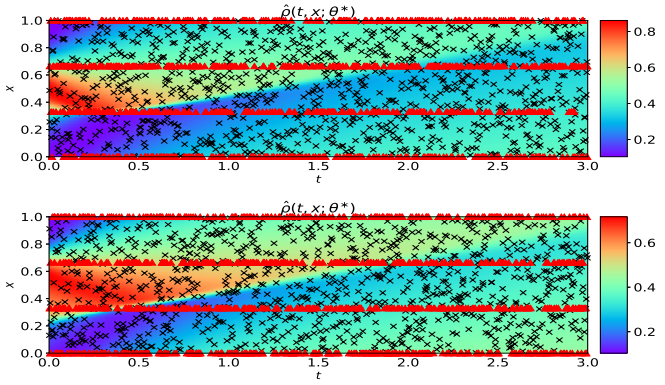

Fig. 4. Sparse presentation of the deployment of observation points O (red triangles) and auxiliary points A (black crosses) for $m_{\text{loop}} = 4$, and comparison of state estimation for Case A (top) and Case B (bottom).

B. State estimation using averaged loop detector observation

We now apply the framework of Section IV-A to averaged data information, which are those usually available in real world situations. For this, we divide the time interval $[0,3]$ into 40 sub-intervals of 72 time grid points each, and we consider the corresponding density and flow averages on each interval as the new target points. We then replace $\hat{\rho}(t_o^{(i)},x_o^{(i)})$, $\rho^{(i)}$, $Q(\hat{\rho}(t_o^{(i)},x_o^{(i)});\theta)$ and $Q^{(i)}$ in (11b) and (11c) by

$$\overline{\rho}(\overline{t}_o^{(i)},x_o^{(i)}) := \frac{1}{R}\sum_{k=1}^{R}\hat{\rho}(t_o^{(R(i-1)+k},x_o^{(i)}), \tag{12a}$$

$$\overline{\rho}^{(i)} := \frac{1}{R}\sum_{k=1}^{R}\rho^{(R(i-1)+k)}, \tag{12b}$$

and

$$\overline{Q}(\overline{\rho}(\overline{t}_o^{(i)},x_o^{(i)});\theta) := \sum_{k=1}^{R}\frac{1}{R}Q(\hat{\rho}(t_o^{(R(i-1)+k},x_o^{(i)});\theta), \tag{12c}$$

$$\overline{Q}^{(i)} := \frac{1}{R}\sum_{k=1}^{R}Q^{(R(i-1)+k)}, \tag{12d}$$

for $i=1,\dots,N_o$, with $N_0 = 40\times m_{\text{loop}}$ and $R=72$, where $\overline{t}_o^{(i)} := \frac{1}{R}\sum_{k=1}^{R}t_o^{(R(i-1)+k}$.

The results obtained are shown in Table III and Fig.6. We can notice that the estimation error oscillates for both cases as the number of loops increases, reaching satisfactory performances for higher loop numbers in both cases A and B, but in general with better results for case A, as shown in Fig. 5.

C. Parameter identification using full loop detector observation

This subsection investigates the ability of PINN method to reconstruct the solution of (8) and at the same time to identify the parameter λ associated with the model. Here we solve

TABLE III
ERROR ON ESTIMATED TRAFFIC DENSITY FOR CASE A AND B FOR AVERAGED DATA

m_{loop}	Error Case A	Error Case B	Relative Error
3	2.818e-02	3.087e-01	9.953
4	1.274e-02	7.002e-02	4.497
5	1.955e-02	2.903e-02	0.485
6	1.666e-02	8.817e-03	0.471
7	1.834e-02	1.005e-02	0.452
8	6.823e-03	7.790e-02	10.418
9	6.569e-03	1.688e-02	1.569

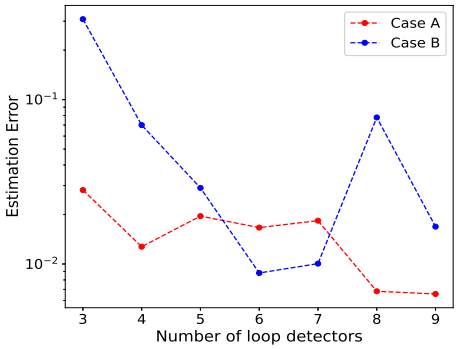

Fig. 5. Estimation error vs number of loops detectors for Cases A and B with averaged loop detector observations.

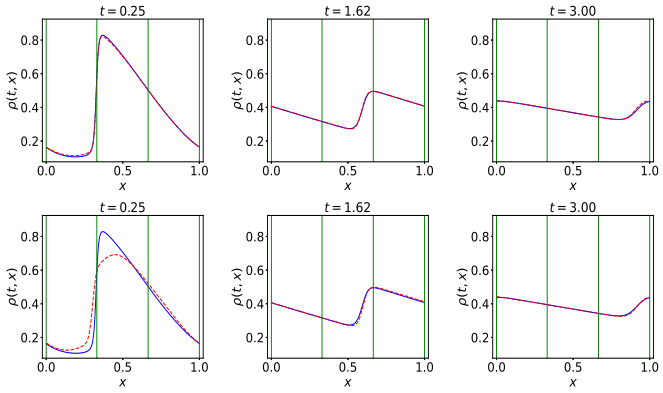

Fig. 6. Comparison of the predicted and exact solutions corresponding to three temporal snapshots for $m_{\text{loop}} = 4$, obtained with state reconstruction using averaged loop detector observation for Case A (top) and B (bottom).

$\lambda^* = \text{argmin}_\lambda \text{Loss}(\lambda)$ for m_{loop} loop detectors, where $\lambda^* = (V^*, R^*, \epsilon^*)$ are the estimated model parameters (which are constraint by setting them equal to zero if they are negative during the training process). The relative error for the density and the estimated parameters for different numbers of loop detectors are shows in Tables IV and V, and the comparison of the density profiles are shown in Fig. 7, showing again better performances with density observations.

TABLE IV
ERROR ON ESTIMATED TRAFFIC DENSITY AND MODEL PARAMETERS FOR CASE A

m_{loop}	Error Case A	$V^*_{\max}(\%)$	$\rho^*_{\max}(\%)$	$\epsilon^*(\%)$
3	2.133e-02	1.346	0.078	13.299
4	1.212e-02	1.988	0.273	16.114
5	9.051e-03	1.151	0.349	14.298
6	4.296e-03	0.957	0.141	15.983
7	8.483e-03	1.080	0.401	17.000
8	4.034e-03	0.604	0.159	14.235
9	5.291e-03	0.322	0.041	12.001

TABLE V
ERROR ON ESTIMATED TRAFFIC DENSITY AND MODEL PARAMETERS FOR CASE B

m_{loop}	Error Case B	$V^*_{\max}(\%)$	$\rho^*_{\max}(\%)$	$\epsilon^*(\%)$
3	2.960e-01	16.801	18.892	58.169
4	9.109e-02	7.611	7.786	54.087
5	3.436e-01	21.815	22.005	8.448
6	2.484e-02	2.215	2.213	22.031
7	1.969e-02	1.293	1.364	15.478
8	9.378e-02	8.327	9.048	7.882
9	1.694e-02	0.510	0.573	44.487

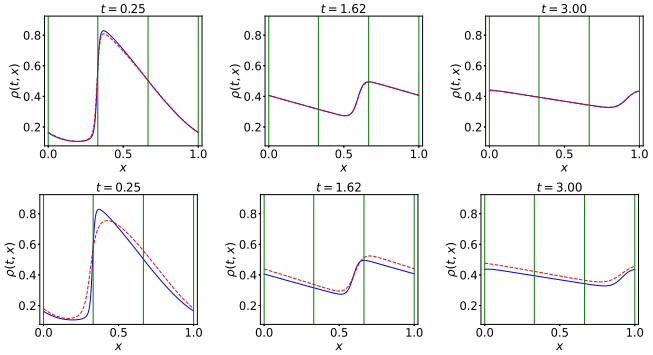

Fig. 7. Comparison of the predicted and exact solutions corresponding to three temporal snapshots for $m_{\text{loop}} = 4$, obtained with parameter identification using full loop detector observation for Case A (top) and B (bottom).

D. Parameter identification using averaged loop detector observation

We apply the same architecture of Section IV-C using averaged data for cases A and B respectively. Compared to those of the previous section, the results in Tables VI and VII are generally worse, especially concerning parameter identification, but become comparable, when not better, for high loop numbers. See also Fig. 8 for an example of reconstructed density profiles.

V. CONCLUSION

We have shown that PINN strategy successfully applies to realistic traffic flow situations using possibly aggregated density and flow measurements at fixed locations. This paves the way to the application of this methodology to real loop

TABLE VI
ERROR ON ESTIMATED TRAFFIC DENSITY AND MODEL PARAMETERS FOR CASE A WITH AVERAGED DATA

m_{loop}	Error Case A	$V^*_{\max}(\%)$	$\rho^*_{\max}(\%)$	$\epsilon^*(\%)$
3	2.336e-02	2.364	0.345	17.035
4	2.442e-02	2.607	0.184	13.646
5	4.157e-03	0.504	0.127	16.450
6	5.464e-03	0.451	0.039	15.835
7	3.420e-03	0.764	0.257	15.672
8	5.781e-03	0.596	0.317	19.678
9	4.147e-03	0.253	0.085	15.397

TABLE VII
ERROR ON ESTIMATED TRAFFIC DENSITY AND MODEL PARAMETERS FOR CASE B WITH AVERAGED DATA

m_{loop}	Error Case B	$V^*_{\max}(\%)$	$\rho^*_{\max}(\%)$	$\epsilon^*(\%)$
3	3.438e-01	26.492	30.134	64.919
4	2.404e-01	22.530	23.851	23.715
5	3.226e-01	18.661	18.762	27.376
6	2.174e-02	0.797	0.740	34.759
7	3.556e-02	3.054	3.094	53.834
8	9.068e-02	7.468	8.230	17.141
9	1.184e-02	0.282	0.253	25.536

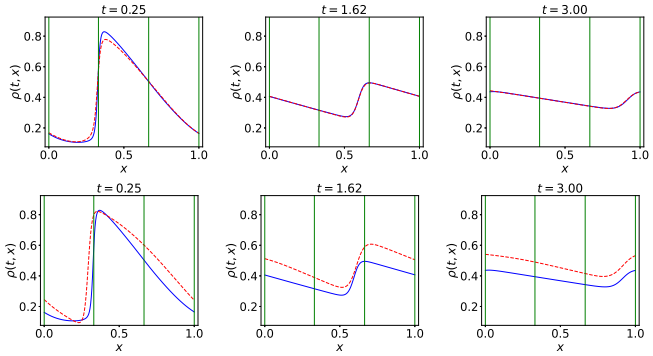

Fig. 8. Comparison of the predicted and exact solutions corresponding to three temporal snapshots for $m_{\text{loop}} = 4$, obtained with parameter identification using averaged loop detector observation for Case A (top) and B (bottom).

detector data. Future work includes the use of speed measurements, possibly along vehicle trajectories or non-evenly distributed loop detectors, and the extension to second order models [15]. To complete the analysis, performances in the case of noisy data need to be evaluated.

Concerning the PINN architecture, improved results could be obtained by a careful calibration of loss weights in (10), see e.g. [4].

The comparison with more established calibration techniques, see e.g. [10]–[15], will complete the assessment of the methodology.

REFERENCES

[1] M. Raissi, P. Perdikaris, and G. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0021999118307125

[2] J. G. Hoffer, A. B. Ofner, F. M. Rohrhofer, M. Lovrić, R. Kern, S. Lindstaedt, and B. C. Geiger, "Theory-inspired machine learning—towards a synergy between knowledge and data," *Welding in the World*, vol. 66, pp. 1291–1304, 2022.

[3] C. Michoski, M. Milosavljević, T. Oliver, and D. R. Hatch, "Solving differential equations using deep neural networks," *Neurocomputing*, vol. 399, pp. 193–212, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231220301909

[4] F. Heldmann, S. Treibert, M. Ehrhardt, and K. Klamroth, "PINN training using biobjective optimization: The trade-off between data loss and residual loss," 2022.

[5] R. Shi, Z. Mo, K. Huang, X. Di, and Q. Du, "A physics-informed deep learning paradigm for traffic state and fundamental diagram estimation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 11 688–11 698, 2022.

[6] M. J. Lighthill and G. B. Whitham, "On kinematic waves. II. A theory of traffic flow on long crowded roads," *Proc. Roy. Soc. London Ser. A*, vol. 229, pp. 317–345, 1955. [Online]. Available: https://doi.org/10.1098/rspa.1955.0089

[7] P. I. Richards, "Shock waves on the highway," *Operations Res.*, vol. 4, pp. 42–51, 1956. [Online]. Available: https://doi.org/10.1287/opre.4.1.42

[8] A. J. Huang and S. Agarwal, "Physics informed deep learning for traffic state estimation," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 2020, pp. 1–6.

[9] R. Shi, Z. Mo, and X. Di, "Physics-informed deep learning for traffic state estimation: A hybrid paradigm informed by second-order traffic models," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 1, pp. 540–547, May 2021. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/16132

[10] S. Fan, M. Herty, and B. Seibold, "Comparative model accuracy of a data-fitted generalized Aw-Rascle-Zhang model," *Netw. Heterog. Media*, vol. 9, no. 2, pp. 239–268, 2014. [Online]. Available: https://doi.org/10.3934/nhm.2014.9.239

[11] G. Dervisoglu, G. Gomes, J. Kwon, R. Horowitz, and P. Varaiya, "Automatic calibration of the fundamental diagram and empirical observations on capacity," in *Transportation Research Board 88th Annual Meeting*, vol. 15. Citeseer, 2009, pp. 31–59.

[12] D. Ngoduy and S. Hoogendoorn, "An automated calibration procedure for macroscopic traffic flow models," *IFAC Proceedings Volumes*, vol. 36, no. 14, pp. 263–268, 2003, 10th IFAC Symposium on Control in Transportation Systems 2003, Tokyo, Japan, 4-6 August 2003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474667017324308

[13] G. Strofylas, K. Porfyri, I. Nikolos, A. Delis, and M. Papageorgiou, "Using synchronous and asynchronous parallel differential evolution for calibrating a second-order traffic flow model," *Advances in Engineering Software*, vol. 125, pp. 1–18, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0965997817308645

[14] P. Wagner, "Fluid-dynamical and microscopic description of traffic flow: a data-driven comparison," *Philos. Trans. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci.*, vol. 368, no. 1928, pp. 4481–4495, 2010. [Online]. Available: https://doi.org/10.1098/rsta.2010.0122

[15] A. Würth, M. Binois, P. Goatin, and S. Göttlich, "Data driven uncertainty quantification in macroscopic traffic flow models," adv. Comput. Math., to appear.

[16] Z. Mao, A. D. Jagtap, and G. E. Karniadakis, "Physics-informed neural networks for high-speed flows," *Computer Methods in Applied Mechanics and Engineering*, vol. 360, p. 112789, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0045782519306814

[17] H. Huang, Y. Liu, and V. Yang, "Neural networks with inputs based on domain of dependence and a converging sequence for solving conservation laws, part i: 1d riemann problems," 2021. [Online]. Available: https://arxiv.org/abs/2109.09316

[18] S. K. Godunov, "A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics," *Mat. Sb. (N.S.)*, vol. 47 (89), pp. 271–306, 1959.