Developing A Physics-informed Deep Learning Paradigm for Traffic State Estimation

Jiheng Huang
University of Central Florida

Part of the Civil Engineering Commons, and the Transportation Engineering Commons

Find similar works at: https://stars.library.ucf.edu/etd2020

This Doctoral Dissertation (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Electronic Theses and Dissertations, 2020- by an authorized administrator of STARS. For more information, please contact STARS@ucf.edu.

STARS Citation
https://stars.library.ucf.edu/etd2020/1579
DEVELOPING A PHYSICS-INFORMED DEEP LEARNING PARADIGM FOR TRAFFIC STATE ESTIMATION

By

ARCHIE J. HUANG
B.Eng. Tsinghua University, 2012
M.S. New York University, 2016

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Civil, Environmental, and Construction Engineering in the College of Engineering and Computer Science at the University of Central Florida Orlando, Florida

Spring Term
2023

Major Professor: Shaurya Agarwal
ABSTRACT

The traffic delay due to congestion cost the U.S. economy $81 billion in 2022, and on average, each worker lost 97 hours each year during commute due to longer wait time. Traffic management and control strategies that serve as a potent solution to the congestion problem require accurate information on prevailing traffic conditions. However, due to the cost of sensor installation and maintenance, associated sensor noise, and outages, the key traffic metrics are often observed partially, making the task of estimating traffic states (TSE) critical. The challenge of TSE lies in the sparsity of observed traffic data and the noise present in the measurements. The central research premise of this dissertation is whether and how the fundamental principles of traffic flow theory could be harnessed to augment machine learning in estimating traffic conditions. This dissertation develops a physics-informed deep learning (PIDL) paradigm for traffic state estimation. The developed PIDL framework equips a deep learning neural network with the strength of the governing physical laws of the traffic flow to better estimate traffic conditions based on partial and limited sensing measurements. First, this research develops a PIDL framework for TSE with the continuity equation Lighthill-Whitham-Richards (LWR) conservation law - a partial differential equation (PDE). The developed PIDL framework is illustrated with multiple fundamental diagrams capturing the relationship between traffic state variables. The framework is expanded to incorporate a more practical, discretized traffic flow model - the cell transmission model (CTM). Case studies are performed to validate the proposed PIDL paradigm by reconstructing the velocity and density fields using both synthetic and realistic traffic datasets, such as the next-generation simulation (NGSIM). The case studies mimic a multitude of application scenarios with pragmatic considerations such as sensor placement, coverage area, data loss, and the penetration rate of connected autonomous vehicles (CAVs). The study results indicate that the proposed PIDL approach brings exceedingly superior performance in state estimation tasks with a lower training data requirement compared to the benchmark deep learning (DL) method. Next, the dissertation continues with an investiga-
tion of the empirical evidence which points to the limitation of PIDL architectures with certain types of PDEs. It presents the challenges in training PIDL architecture by contrasting PIDL performances in learning the first-order scalar hyperbolic LWR conservation law and its second-order parabolic counterpart. The outcome indicates that PIDL experiences challenges in incorporating the hyperbolic LWR equation due to the non-smoothness of its solution. On the other hand, the PIDL architecture with the parabolic version of the PDE, augmented with the diffusion term, leads to the successful reassembly of the density field even with the shockwaves present. Thereafter, the implication of PIDL limitations for traffic state estimation and prediction is commented upon, and readers’ attention is directed to potential mitigation strategies. Lastly, a PIDL framework with nonlocal traffic flow physics, capturing the driver reaction to the downstream traffic conditions, is proposed. In summary, this dissertation showcases the vast capability of the developed physics-informed deep learning paradigm for traffic state estimation in terms of efficiently utilizing meager observation for precise reconstruction of the data field. Moreover, it contemplates the practical ramification of PIDL for TSE with the hyperbolic flow conservation law and explores the remedy with sampling strategies of training instances and adding the diffusion term. Ultimately, it paints the picture of potent PIDL applications in TSE with nonlocal physics and suggests future research directions in PIDL for traffic state predictions.
To My Mom
ACKNOWLEDGMENTS

I would like to express my gratitude to my advisor, Dr. Shaurya Agarwal, who gave me my first job in academia, along with his constant support and guidance. I would also like to thank the faculty members of my doctoral committee — Dr. Yanjie Fu, Dr. Samiul Hasan, and Dr. Patrick Sun for their mentorship during my time at University of Central Florida (UCF). At the same time, I appreciate the work of my collaborators — Mike Muhlmeyer at Northrop Grumman; Dr. Pushkin Kachroo, Dr. Aaron Saiewitz, Dr. Robyn Raschke at University of Nevada, Las Vegas; Dr. Animesh Biswas at University of Nebraska-Lincoln. Meanwhile, I would like to acknowledge Dongjie Wang, Dr. Ou Zheng, Shengxuan Ding, and Dr. Daniel Wang at UCF; Dr. Caio Davi at Hewlett Packard, and Dr. Rongye Shi at Columbia University for their advice and support of my work in this dissertation. I am grateful to my friends at UCF — Dr. Yina Wu, Dr. Qing Cai, Dr. Jinghui Yuan, Dr. Lishengsa Yue, Zijin Wang, Tianyi Li, Ren Hu, Xiangpeng Li, Yiting Wang, Yeting Sun, Bowen Li, Yebowen Hu, Yuyang Zhang, Shanshan Feng, Jinxiang Cheng, and Boyu Zhang; and my teammates at UrbanITY Lab — Shakib Mustavee, Muhammad Shahbaz, and Md Mahmudul Islam. Special thanks to Dr. Saumya Gupta, Shanya, and my mom.
# TABLE OF CONTENTS

LIST OF FIGURES ................................................................. xi

LIST OF TABLES ................................................................. xiv

CHAPTER 1: INTRODUCTION ......................................................... 1

1.1 Background and Motivation ................................................. 1

1.2 Traffic State Estimation ..................................................... 4

1.3 Research Questions ......................................................... 6

1.4 Goals and Objectives ........................................................ 7

1.5 Research Plan ................................................................. 8

1.6 Contributions and Impact .................................................. 9

1.7 Dissertation Outline ....................................................... 11

CHAPTER 2: LITERATURE REVIEW .............................................. 12

2.1 Traffic State Estimation ..................................................... 13

2.1.1 Model-driven approaches ............................................. 14

2.1.2 Data-driven approaches .............................................. 15

2.1.3 TSE with Streaming Data in CAV Environment .................. 16

2.2 Physics-informed Deep Learning ........................................ 19
CHAPTER 3: PHYSICS OF TRAFFIC FLOW

3.1 Scalar Conservation Laws

3.1.1 Strong and Weak Solutions

3.1.2 Method of Characteristics

3.2 Traffic Flow Models

3.2.1 Lighthill-Whitham-Richards Model

3.2.2 Cell Transmission Model

3.3 Fundamental Diagram of Traffic Flow

3.3.1 Greenshields Model

3.3.2 Daganzo Model

3.3.3 Inverse-lambda Model

3.4 Nonlocal Traffic Flow Models

3.4.1 Kernel Functions

CHAPTER 4: PHYSICS-INFORMED DEEP LEARNING

4.1 Deep Learning Neural Network

4.2 Physics-informed Deep Learning

CHAPTER 5: DEVELOPING PIDL PARADIGM FOR TSE

5.1 Approach Overview

5.2 Formulation using LWR Model

5.2.1 LWR and Greenshields’ model
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.1.3</td>
<td>7.1.3</td>
<td>Reconstruction With Initial, Boundary, and Interior Inputs</td>
<td>83</td>
</tr>
<tr>
<td>7.2</td>
<td>7.2</td>
<td>Limitation of PIDL - Insights From Field Data</td>
<td>87</td>
</tr>
<tr>
<td></td>
<td>7.2.1</td>
<td>Data Description</td>
<td>87</td>
</tr>
<tr>
<td></td>
<td>7.2.2</td>
<td>Field Data Reconstruction Using PIDL With Hyperbolic PDE</td>
<td>88</td>
</tr>
<tr>
<td></td>
<td>7.2.3</td>
<td>Field Data Reconstruction Using PIDL With Parabolic PDE</td>
<td>89</td>
</tr>
<tr>
<td></td>
<td>7.2.4</td>
<td>Reconstruction Using Lax-friedrichs’ Numerical Scheme</td>
<td>90</td>
</tr>
<tr>
<td>7.3</td>
<td>7.3</td>
<td>Discussion on Empirical Results</td>
<td>91</td>
</tr>
<tr>
<td>8.1</td>
<td>8.1</td>
<td>Nonlocal Traffic States</td>
<td>95</td>
</tr>
<tr>
<td>8.2</td>
<td>8.2</td>
<td>Kernel Functions</td>
<td>96</td>
</tr>
<tr>
<td></td>
<td>8.2.1</td>
<td>The Constant Kernel</td>
<td>97</td>
</tr>
<tr>
<td></td>
<td>8.2.2</td>
<td>The Linearly Decreasing Kernel</td>
<td>97</td>
</tr>
<tr>
<td>8.3</td>
<td>8.3</td>
<td>PIDL with Nonlocal Physics</td>
<td>97</td>
</tr>
<tr>
<td>8.4</td>
<td>8.4</td>
<td>Field Data Validation</td>
<td>98</td>
</tr>
<tr>
<td>9.1</td>
<td>9.1</td>
<td>Summary of Work and Conclusions</td>
<td>103</td>
</tr>
<tr>
<td>9.2</td>
<td>9.2</td>
<td>Shortcomings and Future Work</td>
<td>105</td>
</tr>
<tr>
<td></td>
<td>9.2.1</td>
<td>Retraining Requirement for Traffic Prediction</td>
<td>105</td>
</tr>
<tr>
<td></td>
<td>9.2.2</td>
<td>The Inverse Problem of Model Identification</td>
<td>106</td>
</tr>
<tr>
<td>9.2.1</td>
<td>9.2.1</td>
<td>Retraining Requirement for Traffic Prediction</td>
<td>105</td>
</tr>
<tr>
<td>9.2.2</td>
<td>9.2.2</td>
<td>The Inverse Problem of Model Identification</td>
<td>106</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>REFERENCES</td>
<td>REFERENCES</td>
<td>109</td>
</tr>
</tbody>
</table>
## LIST OF FIGURES

1.1 Research Thrust I - Developing Physics-informed Deep Learning for TSE .......................... 8  
1.2 Research Thrust II - Analyzing and Improving PIDL Paradigm .......................... 9  
2.1 Traffic State Estimation Approaches ........................................................................ 16  
2.2 Application scenario using Fog architecture .................................................................... 18  
3.1 Shockwave solution ......................................................................................................... 27  
3.2 Rarefaction solution ....................................................................................................... 27  
3.3 Cell Transmission Model (CTM) of Traffic Flow ................................................................. 32  
3.4 Greenshields’ Fundamental Diagram ............................................................................. 33  
3.5 Daganzo’s Fundamental Diagram .................................................................................... 34  
3.6 Inverse-lambda Shaped Fundamental Diagram ................................................................. 35  
3.7 Nonlocal Traffic States .................................................................................................... 37  
3.8 The Constant Kernel ........................................................................................................ 38  
3.9 The Linearly Decreasing Kernel ....................................................................................... 39  
4.1 Architecture of a Deep Learning Neural Network ............................................................ 40  
4.2 Training Process of a Deep Learning Neural Network ....................................................... 42  
4.3 Training Process a Physics-informed Deep Learning (PIDL) Neural Network ................ 46
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Number</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1</td>
<td>PIDL &amp; DL Performance Comparison, Eulerian Traffic Observation</td>
<td>60</td>
</tr>
<tr>
<td>6.2</td>
<td>Computation Time and Error Comparison: Sample Size, Eulerian Traffic Observation</td>
<td>61</td>
</tr>
<tr>
<td>6.3</td>
<td>PIDL &amp; DL Performance Comparison, Lagrangian Traffic Observation</td>
<td>63</td>
</tr>
<tr>
<td>6.4</td>
<td>Computation Time and Error Comparison: Sample Size, Lagrangian Traffic Observation</td>
<td>64</td>
</tr>
<tr>
<td>6.5</td>
<td>Sensitivity Analysis for Various Penetration Rates</td>
<td>71</td>
</tr>
<tr>
<td>6.6</td>
<td>Sensitivity Analysis for Various Data Noise Levels</td>
<td>71</td>
</tr>
<tr>
<td>6.7</td>
<td>Estimated Parameters for Greenshields’ Fundamental Diagram</td>
<td>73</td>
</tr>
<tr>
<td>6.8</td>
<td>PIDL &amp; DL Performance Comparison</td>
<td>77</td>
</tr>
<tr>
<td>7.1</td>
<td>Density Reconstruction Results (with Initial and Boundary Inputs), Relative $L_2$ Error</td>
<td>83</td>
</tr>
<tr>
<td>7.2</td>
<td>Density Reconstruction Results (with Initial and Boundary Inputs &amp; CV Inputs), Relative $L_2$ Error</td>
<td>85</td>
</tr>
<tr>
<td>7.3</td>
<td>Relative $L_2$ Error with Choices of Diffusion Coefficient $\varepsilon$</td>
<td>90</td>
</tr>
<tr>
<td>8.1</td>
<td>Performance Analysis on Nonlocal PIDL: Inaccuracy in Reconstruction</td>
<td>100</td>
</tr>
</tbody>
</table>
CHAPTER 1
INTRODUCTION

1.1 Background and Motivation

The traffic delay\(^1\) due to congestion cost the U.S. economy $81 billion in 2022 [1]. On average, each worker lost 97 hours each year during commute due to longer wait time [2]. Transportation planning agencies have an interest in building transportation infrastructure with enhanced capacity and developing optimized strategies for utilizing existing roads and highways to accommodate the increasing volume of traffic. Intuitively, the solution to the congestion problem seems to reside in enlarging the capacity of existing infrastructure – broadening the roads and increasing the number of highway lanes. However, studies such as [3] [4] have shown that expanding road capacity triggers the effect of induced demand, attracting higher vehicle flows due to the increased capacity. Moreover, monetary and physical constraints exist in projects designed to increase road capacity. Many highways and roads in U.S. cities lack the space to add additional lanes. If more capacity cannot be easily added to the existing infrastructure, traffic management and control strategies serve as an alternative solution to the congestion mitigation problem. Active traffic management and control (ATMC) dynamically manages congestion based on prevailing traffic conditions to maximize transportation efficiency [5]. For instance, highway segments with ramp-control features experience smoother downstream flows, and higher vehicle speed [6]. By observing the conditions of vehicle flow and travel speed on a highway, ramp-control signals allow vehicular flows to join

---

the highway at a certain flow rate to ease congestion. Thus, accurate information on prevailing traffic conditions is essential for designing intelligent controllers.

Macroscopic traffic state variables denote the traffic conditions on road segments in a traffic network. In order to develop traffic control and management strategies, an accurate estimation of macroscopic traffic state variables, such as flow rate $f$, average vehicle speed $v$, and vehicle density $\rho$, is required. Through these indicators, transportation planners can perceive the congestion levels, recognize traffic demand, and even identify gridlocks and bottlenecks of a road network [7]. For instance, a severe decline in travel speed at a road section can implicate unusual events such as traffic incidents or dreadful congestion. Next, the trends and challenges in traffic sensing are discussed.

**Traditional Sensing Approaches:** Data on macroscopic traffic state variables can be collected through devices such as loop detectors and radars. These types of measurement devices are deployed to detect vehicles traveling on the freeways or arterial roads to calculate flow rate, travel speed, and density. Loop detectors are recommended to be installed every one-third of a mile [8]. However, due to the associated construction cost, measurement devices may not be stationed dense enough or evenly distributed on the roads of interest. Images from traffic cameras can also be used to measure traffic. However, these key measurements needed for transportation planning and management are often sparse and potentially noisy [9, 10]. Due to the combined factors such as the cost of sensor installation, the accuracy of vehicle detection methods, and limitation on data storage and transmission, these traffic state variables are often observed partially [11] [12]. For example, traffic data would only be recorded at selected locations with the scattered deployment of vehicle detectors on a highway system [13]. Besides, this type of collected data is often compromised with various levels of inaccuracy due to the existence of measurement noise in any type of detection and sensing devices [14]. Moreover, the data collected are routinely aggregated when transmitted from the sensors, worsening the time resolution of these measurements [15] [16].

**Newer Sensing Options:** Today, observations of traffic states are increasingly available
through another data source - *connected and autonomous vehicles* (CAV) [17]. CAV has the potential to unlock a plethora of road congestion management and safety applications. A widely accepted definition of *connected and autonomous vehicles* is a vehicle that operates at a certain level of autonomy and assists the driver in performing specific driving actions more effectively — thereby resulting in improved safety, fuel efficiency, and reduced travel times [18]. CAV promises a wide variety of active traffic management tools and enables the implementation of communication-based management applications. For example, using vehicle-to-vehicle (V2V) communication, CAVs can broadcast critical safety information to surrounding vehicles and deliver warning messages to other nearby vehicles [19]. Vehicle-to-infrastructure (V2I) communication allows data exchange between CAVs and transportation infrastructure such as traffic signals, stop signs, and roadside units (RSUs).

RSUs can communicate with a CAV within its communication range and collect vehicle data such as its position (latitude and longitude), heading angle, velocity, lateral acceleration, brake status, steering angle, turn signal status, vehicle length, and origin-destination. While at the same time, RSUs broadcast messages such as signal phase and timings (SPaT), basic safety messages (BSM), and other messages containing road conditions and travel times. By using V2I communication technology, drivers can also obtain real-time information on the congestion level [20] and routing information [21].

One of the challenges currently faced is the low penetration rate of CAV in the current fleet of road vehicles, posing obstacles in obtaining rich data on surrounding traffic conditions from vehicles. Other challenges in CAV-based sensing include elevated demand for network infrastructure, on-site computing capability, and low-latency data exchange. Moreover, sensor faults, cyber-attacks, and power and communication outages during natural disasters such as hurricanes may cause data sparsity. Besides, challenges remain in the reliable transmission of data from vehicles to cloud servers and solving the issue of signal loss in rural areas. This necessitates the development of computationally efficient frameworks that can work with low amounts of potentially noisy
input data for an accurate traffic state estimation.

1.2 Traffic State Estimation

Given the aforementioned challenges, traffic state estimation (TSE) becomes an important task. It refers to the inference of traffic state variables of road segments using partially observed traffic data [22]. Accurate TSE in real-time is essential for efficient traffic management as control strategies are implemented accordingly [23].

The practice of ramp control on freeways can illustrate the importance of TSE. It utilizes detected traffic flow data to recognize freeway traffic conditions, then alters the traffic signal timing to allow vehicles at the ramp to join traffic on the freeway according to the upstream and downstream flow levels [24]. Similarly, the dynamic toll price on freeways is another effective traffic management strategy [25]. The traffic management and operation applications rely heavily on accurately perceiving real-time traffic conditions as control strategies and measures are implemented accordingly. The problem presented is how to use limited and potentially noisy data to gain an accurate sense of traffic conditions in real-time. Apart from control strategies on freeways, signal controls on arterial roads also require accurate estimation of states (flow or queue lengths) to mitigate congestion and reduce travel time. Note that, arterial roads account for more than one million lane miles of roadway, connecting local and collector roads to the national highway systems [26].

The two main approaches of TSE are described next.

**Model-driven TSE Approaches:** Model-driven approaches deploy a model from the traffic flow theory in predicting traffic states. For model-driven TSE approaches to accurately reflect the actual state conditions, a calibrated traffic model is needed to represent the specifics of physical boundaries - such as maximum flow $f_m$, critical density $\rho_m$, and free-flow speed $v_f$, given the particular road condition and configuration. Variables $f_m$, $\rho_m$, and $v_f$ differ from segment to segment on a road and the free-flow speed $v_f$ usually is regulated by traffic laws and is not far from the speed limit indicated on road signs. But it can easily be impacted by weather conditions and
unforeseeable events including traffic accidents. Consequently, the task of a traffic model’s parameter estimation is also an essential component of obtaining a valid assessment of traffic conditions. Model-driven approaches sometimes rely on empirical evidence to gain knowledge on variables of free-flow speed \( v_f \) and maximum density \( \rho_m \).

**Data-driven TSE Approaches:** Data-driven approaches tend to establish the values of such variables through a large dataset of observed data. Empirical evidence may mirror the historical conditions of these parameters; however, counting on empirical data may be slow to adapt to the changing nature of traffic conditions. For instance, if the data does not include observation of travel speed after an accident, the TSE based on the model is likely to be far from the delayed traffic in reality. Considering real-time data in parameter estimation is viable to alleviate the problem of model adaptivity. Yet, the noise in data and unreliable readings from malfunctioning sensors will impede the task of real-time parameter estimation.

Deep learning for TSE can swiftly modify its prediction based on real-time data. However, the looseness in the collected data may bring the sensor or detection bias into the neural network, resulting in less reliable estimation results. Besides, this practice relies on high computation capacity to process the enormous volume of incoming traffic data promptly. This compels the construction of a high standard of infrastructure, which includes data collection devices, servers, and data processing centers to support this traffic state estimation method.

**Hybrid Approaches:** Physics-informed deep learning (PIDL), also referred to as physics-informed neural network (PINN), arms a neural network with the physical laws governing the system [27]. It empowers the deep learning neural network with the knowledge of the underlying relationship in the observed data to efficiently exploit limited input for estimation and prediction tasks [28]. By respecting the physical law expressed by a traffic flow model, PIDL represents a deep learning approach that incorporates the “physics-cost” — measurement of the violation of physical law into its cost function. This type of neural network is trained for supervised learning problems while considering the governing non-linear partial differential equations during the
training process [29].

Traffic flow models, such as the LWR (Lighthill-Whitham-Richards) model expressed in (Equation 3.19), illustrate the traffic stream behavior through the continuity equation with an assumption on the equilibrium of the relationship between traffic state variables $\rho$, $v$, and $f$. Therefore the conservation law of traffic, described in the form of the partial differential equation, can be utilized as a priori knowledge of traffic to construct a deep learning neural network for TSE based on limited inputs and creates a spatial-temporal map of the traffic states.

The approach combining the advantage of machine learning and the knowledge of governing physical equations of traffic flow is termed physics-informed deep learning (PIDL). Together, the data-driven approach and the physics of traffic flow have the potential to build a fast, resilient, and computation-friendly TSE strategy. Moreover, when the measurements of traffic states at fixed locations are unavailable due to the malfunction of sensing devices or incidents like cyber-attacks, the PIDL approach presents an ideal alternative in utilizing potentially sparse and noisy data to estimate and predict traffic state variables.

Problem Formulation of Traffic State Estimation: Mathematically, the problem of traffic state estimation is formulated as follows:

Let $\mathbb{P} = \mathbb{X} \times \mathbb{T}$ be a space-time domain and $\rho(x,t)$ represents the value of the field at $(x,t) \in \mathbb{P}$. On a given road $\mathbb{X}$, $x_i \subset \mathbb{X}$ are the discretized homogeneous road segments and $t_j \subset \mathbb{T}$ are the time intervals.

In $\mathbb{P}$, sparse data of the field $\rho(x,t)$ at $(x^d_i,t^d_j)$ are observed. Collectively, the observed $(x^d_i,t^d_j)$ constitutes a sub-domain $\mathbb{D} \subset \mathbb{P}$. Given that $\forall (x,t) \in \mathbb{D}$, $\rho(x,t)$ is known, the reconstruction and prediction problem becomes finding a mapping function $F(\cdot) : \rho(\mathbb{D}) \to \rho(\mathbb{P})$, which minimizes the reconstruction cost of $\rho(\mathbb{P})$.

1.3 Research Questions

Carefully analyzing the existing literature leads to the following research questions for this study:
1. Can the fundamental principles of traffic flow theory be harnessed to augment the power of machine learning in estimating traffic state?

2. Which form of the flow conservation law, a hyperbolic PDE or its parabolic variant, is better incorporated by the PIDL approach, in terms of the accuracy of traffic state estimation?

3. What is the effect of a diffusion term, which transforms the hyperbolic LWR PDE to a parabolic variation, in improving the state estimation results of PIDL?

4. What are the limitations of PIDL in TSE? What tools are available to transportation researchers to alleviate the weakness of PIDL?

5. Can the use of nonlocal physics mitigate some of the challenges PIDL faces in TSE?

### 1.4 Goals and Objectives

The overarching goal of this research is to develop a robust, resilient, and computationally efficient framework that is capable of online traffic state estimation. Specifically, this research’s objectives are:

- To develop a PIDL framework capable of working with in-stream data from CAVs as well as data from fixed sensors for TSE.

- To demonstrate the capability of PIDL in TSE based on the partial and noisy observations from loop detectors and CAVs.

- To design experiments using synthetic and realistic datasets coupled with different types of fundamental diagrams.

- To understand the shortcomings and propose and discuss mitigation approaches.
1.5 Research Plan

The research plan is divided into two research thrusts. **Research thrust I** has three tasks that aim to work with a low amount of noisy observation inputs while keeping the computing limitations in mind to achieve an accurate and timely estimation of traffic states. The research tasks are further explained below and in Figure 1.1.

1. Design a physics-informed deep learning (PIDL) approach for traffic state estimation with the Lighthill-Whitham-Richards (LWR) model.

2. Enhance the designed PIDL framework to incorporate the discretized cell transmission model (CTM).

3. Evaluate the PIDL performance with the (a) Eulerian, (b) Lagrangian, and (c) CAV-based traffic observation scenarios.

**Research Question 1**
Can physics from traffic flow theory augment deep learning in traffic state estimation (TSE)?

**Research Task 1**
Design a physics-informed deep learning (PIDL) approach for TSE with the LWR model.

**Research Task 2**
Formulate the training of PIDL neural network for TSE with the CTM.

**Research Task 3**
Evaluate PIDL performance for TSE with a selection of application scenarios.

---

**Research thrust II** is composed of three tasks (4, 5, and 6). They are summarized below and depicted in Figure 1.2:
4a. Exhibit the contradistinction in terms of reconstruction accuracy between utilizing a first order hyperbolic PDE, and its second order parabolic counterpart of the commonly-used LWR conservation law in traffic flow theory.

4b. Investigate the impact of discontinuity in PDE solution and the effect of a diffusion term on PIDL learning.

5a. Illustrate the limitations of PIDL in TSE while training with only the initial and boundary observations.

5b. Discuss the findings from the experimental results in light of existing literature and put forward suggestions to improve PIDL results in TSE with the LWR PDE.

6. Develop and evaluate a PIDL framework incorporating nonlocal physics.

![Figure 1.2: Research Thrust II - Analyzing and Improving PIDL Paradigm]

1.6 Contributions and Impact

Research thrust I focuses on developing a new approach to address the challenge of estimating traffic state using limited and potentially noisy traffic input. Equipped with the physics of traffic,
namely traffic flow theory, a deep learning neural network is proposed to estimate traffic conditions based on limited observation traffic data. Furthermore, it offers details of the proposed PIDL algorithm by demonstrating its application using the hydrodynamic traffic conservation law (LWR) model and a numerically more appropriate discrete cell transmission (CTM) model, and the incorporation of multiple fundamental diagrams is demonstrated. The inverse-lambda fundamental diagram has the practical consideration of the capacity drop when vehicle density has reached the critical level. Research thrust II includes a series of experiments designed by creating a circular testbed and using realistic field (NGSIM) data to understand the performance of PIDL while incorporating the hyperbolic and parabolic versions of the physics. Thereafter, the implication of PIDL limitations for traffic state estimation and prediction is commented upon, and readers’ attention is directed to potential mitigation strategies. Lastly, a PIDL framework with nonlocal traffic flow physics is proposed. In summary, the contributions of this dissertation are:

- Developed a physics-informed deep learning (PIDL) paradigm for traffic state estimation and reconstruction.
- Designed case studies to test the developed framework in various sensing scenarios and practical considerations.
- Highlighted the contrast between PIDL incorporating the hyperbolic LWR PDE and its parabolic variant using a mix of Eulerian and Lagrangian training inputs.
- Discussed the difficulties encountered when training a PIDL neural network with the LWR conservation law, and put forward the mitigation strategies to improve the performance.
- Presented a PIDL framework with nonlocal traffic flow physics, considering the integration of downstream traffic conditions.

**Research Impact:** This research showcases the vast capability of the developed physics-informed deep learning paradigm for traffic state estimation with regard to efficiently utilizing
meager observation for precise reconstruction of the data field. This dissertation also contemplates the practical ramification of PIDL for TSE with the hyperbolic flow conservation law and denotes the remedy with sampling strategies of training instances and adding diffusion terms. Finally, it paints the picture of potent PIDL applications in TSE with nonlocal physics and suggests future research directions in PIDL for traffic state predictions. The findings could lead to a better approach for TSE, efficiently utilizing traffic observations to obtain a precise perception of traffic conditions, ultimately leading to better control and management of traffic on the transportation infrastructure, and relieving traffic congestion.

1.7 Dissertation Outline

The rest of the dissertation is organized as follows: chapter 2 presents the literature review of traffic state estimation (TSE) and physics-informed deep learning (PIDL). Chapter 3 provides a background of scalar conservation law and the physics of traffic flow. Chapter 4 gives the details of physics-informed neural network architecture. Chapter 5 develops a PIDL paradigm for TSE. Chapter 6 validates the capability of PIDL with case studies using synthetic and field data. Chapter 7 analyzes the limitations of PIDL and discusses potential mitigation approaches. Chapter 8 demonstrates the integration of nonlocal physics into PIDL framework. Finally, chapter 9 concludes the dissertation with an outlook on the applications of PIDL in the transportation domain and future research directions.
CHAPTER 2
LITERATURE REVIEW

Macroscopic traffic state variables\(^1\), such as flow rate \(f\), mean speed \(v\), and vehicle density \(\rho\), denote the traffic conditions on the road infrastructures in a traffic network. Through these measurements, urban planners and policymakers can perceive the congestion levels, understand traffic demand, and even recognize gridlocks and bottlenecks of a road network [7]. For example, a sharp deterioration in travel speed at a road section can indicate particular events such as traffic incidents or disturbances. Observation of such traffic state variables can be obtained through a variety of methods: inductive loop detectors, acoustic sensors, and video cameras are widely used for vehicle detection and classification applications [30, 31, 32]. Nevertheless, the crucial measurements of traffic conditions, necessitated for traffic management and planning are frequently scanty and likely noisy [10]. Due to the various factors such as the cost of sensor installation, the precision of vehicle detection methods, and constraints on data storage and communication, these traffic state variables are frequently observed partially [11, 12]. For instance, traffic data would only be registered at chosen locations with the scattered deployment of vehicle detectors on a highway system [13]. Additionally, this type of recorded data is often compromised with various levels of imprecision due to the presence of measurement noise in detection and sensing devices [14]. Moreover, the data collected are routinely aggregated, worsening the temporal resolution of these measurements [15, 16].

The challenge is how to efficiently utilize limited, sparsely sampled (in spatial and temporal

domains), and potentially noisy data to gain a clear sense of traffic conditions in real time. Tools that fill the voids in traffic measurements and provide a reliable description of traffic conditions are referred to as traffic state estimation (TSE) techniques. In other words, TSE relates to the inference of traffic state variables of road segments using partially observed traffic data [22]. Accurate and prompt TSE is of the essence for effective traffic management since control strategies are implemented accordingly [23]. For instance, the usage of ramp control on freeways exemplifies the value of TSE. It uses the measured traffic flow data to estimate freeway traffic conditions, then alternates the traffic signals to allow vehicles at the ramp into the traffic stream according to the upstream and downstream flow levels [24].

2.1 Traffic State Estimation

Traffic state estimation (TSE) refers to the procedure of inferring traffic conditions — namely traffic flow, velocity, and speed — along a road infrastructure over time by using obtained traffic observations either through sensing devices or broadcast by connected vehicles [22]. Given the impediment discussed above in traffic data collection and the importance of TSE applications in transportation planning, practitioners and researchers often use a priori knowledge to estimate traffic states. These estimation procedures can be categorized as model-driven approaches and data-driven approaches, based on the type of a priori assumption they rely on [22]. Model-driven approaches have the advantage of interpretability and are efficient with a limited amount of training data available [33]. However, they may not generalize well with unobserved traffic data and needs recalibration when an anomaly occurs due to severe weather or traffic accidents. Meanwhile, data-driven approaches are generalizable at the cost of expensive data acquisition for a huge amount of training observations [33].
2.1.1 Model-driven approaches

Model-driven approaches deploy a model from the traffic flow theory in predicting traffic states. Models such as the cell transmission model (CTM) [34], and the switching mode model (SMM) [35] have been proposed to represent traffic flows. Lighthill-Whitham-Richards (LWR) model, along with higher-order models such as Payne-Whitham (PW) model [36, 37], Aw-Rascle-Zhang (ARZ) model [38, 39] are widely used because of their accurate traffic characterization and fair computation cost.

Kalman filter [40] and its various extensions — such as extended Kalman filter [41, 42], ensemble Kalman filter [43, 44], unscented Kalman filter [45] — are commonly used to solve the task of TSE efficiently and calibrate the traffic flow models. The Kalman filter and its variations determine the most presumable state for the observed data, system variables of the model, and noises [46]. However, a few major challenges remain in the model-driven approach for TSE. The estimation of model parameters, changing road conditions (e.g., due to lane closures, road construction projects), and suitability of a chosen model for the exact road location of interest remain major hurdles in adopting the traffic flow model approach for TSE [22]. Data-driven Koopman theoretic approaches have been explored recently for model discovery [47].

Suppose the selected traffic flow model can convincingly capture the relationship between traffic state variables such as flow and density observed in reality. In that case, the model-driven approach — based on the physics of traffic flow — can precisely predict the traffic states in unobserved areas with no data collection devices. In addition, it can yield a higher spatial and temporal resolution of traffic state data at locations where data collection technology is deployed. On the other hand, the dependence on the traffic flow model brings the vulnerability of unfit model adoption. Using standard traffic flow models with empirical evidence somewhat lowers this kind of risk. Nevertheless, the robustness of estimation in the event of an anomaly such as a traffic accident or inclement weather condition leaves room for improvement in the estimation result.

If the adopted traffic flow model can accurately illustrate the relationship between traffic state
variables observed in reality, the model-driven approach — based on the physics of traffic flow — can precisely predict traffic states in unobserved areas (with no data collection devices) and provide higher resolution of traffic state data (at data collection locations).

However, the dependence on the traffic flow model brings the risk of adopting an unfit model in TSE. Using common and proven traffic flow models somewhat lowers this kind of risk, nevertheless, the robustness of estimation in the event of an anomaly such as a traffic accident or inclement weather condition leaves room for improvement.

2.1.2 Data-driven approaches

With the advancement of statistical tools and machine learning, data-driven methods have become another type of prominent approach for TSE. The data-driven approach of traffic state estimation addresses the issue with traffic flow model accuracy from another source: the data collected on the roadways. Instead of relying on the formulated traffic flow models based on empirical evidence or historical experience, it applies rapidly emerging machine learning (ML) techniques to recognize the relationship between traffic state variables. Data-driven TSE adopts the insights of historically observed traffic data for estimation and prediction tasks. It enables the model to discover the underlying traffic data structure, eliminating the requirement of fitting a traffic flow model suitable for each road segment involved. The ML algorithm adapts to the particular environment pattern, such as congestion or rush-hour traffic, based on the input data it gets in real-time. Because of its adaptability, the data-driven TSE result is expected to be more reliable when a traffic pattern anomaly is present compared to a model-based approach. In this kind of approach, probabilistic principal component analysis [48], kernel regression [49], k-nearest neighbors [50], Bayesian state-space estimation [51] and Deep learning [52], recurrent neural network (RNN) [53, 54] and long short-term memory (LSTM) [55, 56] neural network have all been experimented and applied in the literature.

However, the current machine learning (ML) based approaches rely overly on the obtained
traffic data, which leads to over-fitting when applied in a different traffic scenario. The lack of robustness limits the applications of the experimental ML models. For instance, the probabilistic approach cannot distinguish some temporal patterns from long-term trends [48], and $k$-nearest neighbors may not be an ideal approach when unusual traffic patterns occur [50].

State-of-the-art approaches such as LSTM and RNN-type neural networks have given impressive performance in capturing the nonlinear relationships among traffic states [57]. However, LSTM and RNNs are known for the hindrance in the training process as updating the weights, and bias parameters consume immense memory and computational resources. In other words, these algorithms and their variants are not suitable for hardware acceleration. Moreover, these algorithms are affected by different random weight initialization and are also prone to over-fitting. The relationship between the conventional TSE approaches and the proposed PIDL approach is shown in Figure 2.1.

![Figure 2.1: Traffic State Estimation Approaches](image)

2.1.3 TSE with Streaming Data in CAV Environment

A widely accepted definition of connected and autonomous vehicles (CAVs) is a vehicle that operates at a certain level of autonomy and assists the driver in performing specific driving actions more effectively — thereby resulting in improved safety, fuel efficiency, and reduced travel times [18]. For this dissertation, a vehicle is considered as a CAV that is connected and may have autonomy.
of some level.

The advancement in sensing and computing provides endless opportunities to utilize rich data generated together by CAVs and connected infrastructure to develop real-time cognitive awareness of road conditions for transportation planning and management. However, this type of communication-based application has elevated demand for network infrastructure in pursuit of robust on-site computing capability and low-latency data exchange. Fog computing, which has emerged as an appropriate choice for CAV and connected infrastructure applications, is computationally constrained compared to cloud-based centralized architectures. Moreover, sensor faults, cyber-attacks, power, and communication outages during natural disasters such as hurricanes may cause data sparsity. This necessitates the development of computationally efficient algorithms that can work with low-input data.

Communication-based applications for CAV require significant roadside communication as well as computing infrastructure. As a result, the demand for real-time data processing using cloud computing has grown exponentially. One of the challenges for cloud computing is network latency and congestion issues, as often it is processed in geographically distant data centers. Moreover, cloud computing infrastructure is centralized and poses significant threats to data privacy and security, along with practical challenges such as transmission latency and network efficiency [58].

On the other hand, there is an abundant processing power untapped or generally under-utilized in on-site edge nodes, also termed as cloudlets [59] or fog nodes [60] — which are placed at the edge of the network and in close proximity to the end devices where data originates [61]. Edge devices such as local traffic controllers, and roadside units can receive and record traffic data reported by intelligent vehicles or smart onboard devices regarding vehicle location and velocity. These edge devices are in close proximity to end devices (CAVs in this case) — where data is being generated and used.

In other words, edge computing pushes the brainpower and communication capabilities into the end device via programmable logic controllers and programmable automation controllers. If
interconnected via LAN, edge devices can constitute an efficient network for data collection and real-time computation — often referred to as ‘fog’ computing or fogging. The concept of fog computing was developed to harness the data collection, storage, and computing power of end devices and serve as an intermediate layer between the edge layer and cloud servers — thus decreasing network latency and improving system response time [62].

Hence, fog computing emerges as an appropriate choice for CAV and connected infrastructure applications, meeting the demand on low-latency and real-time computing power, location-aware communication of cognitive awareness applications [63]. An application scenario employing fog infrastructure is illustrated in Figure 2.2.

![Figure 2.2: Application scenario using Fog architecture](image)

Dedicated short-range communication (DSRC) is a wireless communication protocol providing the means of V2V and V2X information exchange. DSRC is particularly designed for vehicle safety purposes to have low latency associated with the data transmission process. United States Department of Transportation (USDOT) and National Highway Traffic Safety Administration (NHTSA) have announced the adoption of DSRC and mandate it for new light vehicles [64] [65]. Its nature of short-range communication typically allows V2V and V2X communication
within a 300-meter range [66].

If deployed intermittently, edge devices such as RSUs may not locate within the DSRC communication range, restricting data sharing between RSUs. Fog infrastructure provides a fitting solution to this obstacle by connecting the otherwise isolated RSUs. It harnesses the data collection capability of the RSUs and serves as an intermediate layer between edge devices and the cloud server. This tiered architecture reduces communication frequency to the cloud and overcomes bottlenecks in the network by exchanging processed information with the cloud.

Currently, as conventional vehicles constitute the majority of the fleet, a common scenario on the road is that only a few cars are equipped with V2V and V2X communication capabilities. Intelligent vehicles communicate and broadcast vehicle information to edge devices such as roadside units within the communication range. Data from diverse road segments is then transmitted to fog infrastructure for perceiving and estimating the traffic conditions in the entire transportation network.

The discussion above highlights the importance of intelligent infrastructure for time-critical CAV applications. While considering fog architecture, the two main challenges that emerge include data sparsity and the computational efficiency of deployed algorithms.

2.2 Physics-informed Deep Learning

Deep learning (DL) neural network is a powerful machine learning method increasingly used in many TSE applications [67, 68, 69]. However, DL neural network also comes with shortcomings, such as the high requirement of training data and computing power, over-fitting, and transferability issues, limiting its appeal for time-critical applications, which calls for the role of physics in aiding the training process of a neural network in TSE [70, 71]. Physics-informed deep learning (PIDL), also referred to as physics-informed neural network (PINN), arms a neural network with the governing equations of a physical system [29, 27]. It empowers the deep learning neural network with knowledge of the underlying relationship in observed data to efficiently use the limited data input
for estimation and prediction tasks [72, 28].

Since the introduction of its architecture [72], variants of PIDL have been proposed to learn the solution of partial differential equations (PDEs). For instance, Galerkin method-based hp-VPINN was introduced to solve PDEs with non-smooth solutions [73]. A Bayesian approach to PINN is presented for forward and inverse problems [74], and the idea of physics-informed adversarial training to solve PDE is proposed [75]. Particle swarm optimization is also put forward to PIDL training [76]. A diverse range of mechanical engineering and computational science applications have also been proposed, signifying its advantage in utilizing the governing equation to accurately capture the physical system. Among its diverse implementations in the mathematical domain, the PIDL approach has been adopted for solving the free boundary problems [77], high-dimensional PDE [78], uncertainty quantification [79], and time-dependent stochastic PDE [80]. In the field of fluid dynamics, PIDL is employed to model the velocity and pressure fields [81], vortex-induced vibration [82], and fluid flows without the use of simulation data [83]. And on the engineering side, the modeling of cardiovascular flow [84], nano-optics [85], and proxy modeling in solid mechanics [86] have all witnessed the effectiveness of the PIDL approach.

PIDL has been adopted in the field of traffic state estimation (TSE) [87, 88, 89]. Researchers have experimented with PIDL to estimate both the traffic state and the fundamental diagram depicting the relationship between traffic states [68]. Second-order traffic models have also been considered in the application of PIDL for TSE [90]. Amid the PIDL applications that have been demonstrating encouraging results, the focal point is to use the neural network to learn the solutions of deterministic PDE commanding a physical system [91]. As the PIDL model ciphers the underlying relationship between state variables, it incorporates the governing equation as a priori knowledge into calculating cost. If the PDE of interest is smooth and has a strong solution, at the same time, is paired with an adequate number of collocation points where the physics cost is optimized, then PIDL accordingly is capable of achieving good accuracy in learning the solution to the PDE [92]. However, recent research points to the limitations of PIDL for learn-
ing certain types of PDEs, such as hyperbolic conservation laws. No strong solution exists to Lighthill-Whitham-Richards (LWR), which is a one-dimensional scalar conservation law and a commonly-used transportation model, given it is a hyperbolic PDE [24]. No empirical evidence exists that PIDL can learn an LWR PDE with acceptable accuracy given only boundary and initial condition data, which is the default experimental setup for a variety of reconstruction problems in transportation and other domains [29, 93].

2.3 Summary and Research Gap

Firstly, Traffic state estimation (TSE) is an imperative task in traffic management, and various approaches — either based on traffic flow models, or machine learning techniques — have been developed to perform TSE. The restraints on obtaining traffic state data call for the adoption of TSE approaches that can exploit a limited amount of observed data for training and produce a precise estimation of traffic states. Hence, compared to the approach of blindly feeding traffic data into a learning model, the fundamental principles of traffic flow theory should be harnessed to augment the power of neural networks for TSE. Moreover, when sequential data of traffic states at fixed locations are not available, or during the downtime of sensing devices due to malfunction or cyber-attacks, the PIDL approach presents an attractive alternative in utilizing potentially noisy data for TSE. It has the potential to build a fast, resilient, and computation–friendly TSE approach. This dissertation develops a PIDL paradigm for TSE. Secondly, recent research points to the challenges of training a PIDL neural network with hyperbolic PDEs to which the solution has discontinuity present. The LWR conservation law in traffic flow theory is hyperbolic without the diffusion term (and parabolic with diffusion). There is a need to analyze the performance of adopting LWR as physics in PIDL for TSE and propose mitigation strategies for the challenges PIDL encountered in TSE. So far, no existing literature examines the challenges and limitations of PIDL with the LWR PDE for TSE, and this dissertation aims to fulfill this research gap in investigating the difficulty encountered in training PIDL with LWR and discussing the remedy strategies. Finally,
there is a growing interest in the literature about modeling the traffic flow with nonlocal traffic states as it captures the natural driving behavior of reacting to downstream traffic conditions. The integral form of nonlocal traffic density potentially can also provide a way to diffuse the discontinuity around the shockwaves, improving the TSE result. No studies in the literature contemplate incorporating the nonlocal physics of traffic states in PIDL for TSE, therefore this research also develops PIDL with nonlocal traffic states as an approach to strengthening the PIDL paradigm for TSE.
CHAPTER 3

PHYSICS OF TRAFFIC FLOW

This chapter provides a background on various physics-based traffic flow models that would be useful in designing a physics-informed deep learning neural network for TSE. It covers the concepts of scalar conservation laws, traffic flow models, fundamental diagrams, and the nonlocal physics of traffic flow.

3.1 Scalar Conservation Laws

In one dimension, a general form of the scalar conservation law is described by (Equation 3.1) where \( f(u) \) is the flux function depending on the location \( x \) and time \( t \). The conserved variable is \( u = u(x, t) \). Equation (Equation 3.1) becomes a Cauchy problem when the initial condition \( u(x, 0) = u_0(x) \) is provided [94].

\[
\partial_x f(u) + \partial_t u = 0, \quad x \in \mathbb{R}, \quad t \geq 0 \quad (3.1)
\]

The flux function \( f(u) \) can take many forms, the linear advection equation is a basic example of the scalar conservation law where the flux function \( f(u) \) takes the form of \( f(u) = \lambda u \) with a constant \( \lambda \), shown in (Equation 3.2).

\[
\lambda \partial_x u + \partial_t u = 0, \quad x \in \mathbb{R}, \quad t \geq 0 \quad (3.2)
\]

---

An initial value problem for the linear advection equation is given by (Equation 3.3) and its solution is exhibited in (Equation 3.4).

\[
\begin{align*}
\partial_x f(u) + \partial_t u &= 0, \quad x \in \mathbb{R}, \quad t \geq 0 \\
u(x, 0) &= u_0(x) = f(x_0), \quad x \in \mathbb{R} \\
u(x, t) &= u_0(x - \lambda t), \quad t \geq 0
\end{align*}
\tag{3.3}
\]

A simple nonlinear partial differential equation is the Burgers' equation, and it is one of the commonly used models as the scalar conservation law. The classical form of Burgers’ equation is presented in (Equation 3.5) where the \(\varepsilon \partial_{xx}u\) is the viscosity term.

\[
\begin{align*}
u \partial_x u + \partial_t u - \varepsilon \partial_{xx}u &= 0, \quad x \in \mathbb{R}, \quad t \geq 0 \\
\tag{3.5}
\end{align*}
\]

When \(\varepsilon = 0\), (Equation 3.5) becomes the inviscid Burgers’ equation, and the flux function takes the form of \(f(u) = u^2/2\), shown in (Equation 3.6). Plugging \(\lambda = u\) into (Equation 3.4) will get the solution of the inviscid Burgers’ equation in (Equation 3.7).

\[
\begin{align*}
\partial_x (u^2/2) + \partial_t u &= u \partial_x u + \partial_t u = 0, \quad x \in \mathbb{R}, \quad t \geq 0 \\
u(x, t) &= u_0(x - ut), \quad t \geq 0 \\
\tag{3.6}
\end{align*}
\]

The viscosity term is a diffusion term that flattens discontinuities and ensures a smooth solution.

### 3.1.1 Strong and Weak Solutions

Given the initial value problem (Equation 3.8),
\[ \frac{\partial}{\partial x} f(u) + \frac{\partial}{\partial t} u = 0, \quad x \in \mathbb{R}, \ t \geq 0 \]  
(3.8)

\[ u(x, 0) = u_0(x), \quad x \in \mathbb{R} \]

If \( u_0(x) \in C^1(\mathbb{R}) \), then the initial condition \( u_0(x) \) is continuously differentiable and (Equation 3.8) becomes (Equation 3.9) by virtue of the chain rule.

\[ f'(u) \frac{\partial}{\partial x} u + \frac{\partial}{\partial t} u = 0, \quad x \in \mathbb{R}, \ t \geq 0 \]
(3.9)

\[ u(x, 0) = u_0(x), \quad x \in \mathbb{R} \]

In domain \( \Omega \subset \mathbb{R} \), a solution to (Equation 3.9) is a \textbf{strong solution} (also referred to as “classical solution”) if it satisfies (Equation 3.8), and is continuously differentiable on the domain \( \Omega \).

When no strong solution to (Equation 3.8) exists, the smoothness requirement can be relaxed to find \textbf{weak solutions}, even if these solutions are not differentiable or even continuous.

Multiplying the scalar conservation law (Equation 3.1) with a function \( \psi : \mathbb{R} \times \mathbb{R}^+ \to \mathbb{R} \), and given the initial condition \( u(x, 0) = u_0(x) \), we have (Equation 3.10).

\[ \int_0^\infty \int_{-\infty}^\infty (\frac{\partial}{\partial x} f(u) + \frac{\partial}{\partial t} u) \psi \ dx \ dt = \int_0^\infty \int_{-\infty}^\infty (f(u) \psi_x + u \psi_t) \ dx \ dt + \int_{-\infty}^\infty u(x, 0) \psi(x) \ dx = 0 \]  
(3.10)

Notice in (Equation 3.10), the requirement on smoothness is lessened as the derivative terms of \( u \) and \( f(u) \) are eliminated. If (Equation 3.10) satisfies all \( \psi(x) \), then \( u(x, t) \) is the weak solution of (Equation 3.8).

It is worth noting that there is \textbf{no strong solution to the LWR conservation law}. Nevertheless, a diffusive term can be added to avoid breakdown and ensure a strong solution by making the hyperbolic conservation equation become a parabolic PDE.
3.1.2 Method of Characteristics

The method of characteristics is used to solve quasilinear partial differential equations, converting
the PDEs into ordinary differential equations (ODEs). Consider (Equation 3.1) and its solution
\( f(u) = u(x, t) \), let \( x = x(t) \) solve the ODE in (Equation 3.11):

\[
\dot{x}(t) = u(x(t), t) \tag{3.11}
\]

From (Equation 3.11) observe that,

\[
\frac{d}{dt} u(x(t), t) = \frac{dx}{dt} u_x + u_t \tag{3.12}
\]

Combining (Equation 3.1) and (Equation 3.12), we reach (Equation 3.13) that can propagate
the solution \( u(x(t), t) \) with the initial condition \( u_0(x) \).

\[
\frac{du}{dt} = 0, \quad \frac{dx}{dt} = u \tag{3.13}
\]

A simple discontinuous solution of the conservation law (Equation 3.8) is given by
(Equation 3.14).

\[
\begin{cases} 
  u_L, & x < \lambda t, \\
  u_R, & x \geq \lambda t,
\end{cases} \tag{3.14}
\]

If \( u_L \neq u_R \), (Equation 3.14) is termed a shock wave. With the shock speed \( \lambda \), it connects \( u_L \) to
the value of \( u_R \). Consider the following scalar Riemann problem (Equation 3.15) where \( \rho_L < \rho_R \):

\[
v_f(1 - \frac{\rho(x, t)}{\rho_m}) \frac{\partial \rho(x, t)}{\partial x} + \frac{\partial \rho(x, t)}{\partial t} = 0
\]

\[
\rho(x, 0) = \begin{cases} 
  \rho_L, & x < 0, \\
  \rho_R, & x \geq 0,
\end{cases} \tag{3.15}
\]
The characteristic speed at $t = 0$ is given in (Equation 3.16). As $\rho_L < \rho_R$, the characteristic speed $\lambda(\rho_L)$ on the left is greater than the right $\lambda(\rho_R)$ and develops a shock curve, shown in Figure 3.1.

$$\lambda(\rho) = f'(\rho) = v_{f}(1 - \frac{2\rho}{\rho_m})$$  \hspace{1cm} (3.16)

If the problem in (Equation 3.15) is modified with the initial condition that $\rho_L > \rho_R$, the value of the characteristic speed on the left $\lambda(\rho_L)$ will be smaller than the value of speed on the right $\lambda(\rho_L)$. One of the possible solutions to (Equation 3.15) with $\rho_L > \rho_R$ is the symmetry rarefaction solution, which is stable to perturbation to the initial data. The solution is given in (Equation 3.17), and shown in Figure 3.2.

$$\rho(x, t) = \begin{cases} 
\rho_L, & \frac{\lambda(\rho_L)}{t} < \lambda(\rho_L), \\
\frac{\lambda(\rho_R) - \lambda(\rho_L)}{\rho_R - \rho_L} \frac{\lambda(\rho_L)}{t}, & \lambda(\rho_L) \leq \frac{\lambda(\rho_R) - \lambda(\rho_L)}{\rho_R - \rho_L} < \lambda(\rho_R), \\
\rho_R, & \lambda(\rho_R) \leq \frac{\lambda(\rho_R)}{t} 
\end{cases}$$  \hspace{1cm} (3.17)
3.2 Traffic Flow Models

Macroscopically, traffic flow is expressed using variables such as mean velocity $v(x, t)$ a function of location $x$ and time $t$, traffic flow $q(x, t)$, and vehicle density $\rho(x, t)$. Spacing $s$ is the headway distance and it is the inverse of density $\rho$. Traffic flow is considered a continuum flow with a density profile associated with a compressible liquid [95, 96]. The traffic flow velocity is related to the density profile, time, and location.

Traffic flow models use empirical data and the developed hypotheses to model traffic conditions [97]. Let the flow rate $q$ indicate the number of vehicles that pass a set location in a unit of time. Average speed $v$ is the mean value of speed among vehicles traveling on a road segment. Density $\rho$ represents the number of vehicles in a unit road of space. Together, these quantities narrate the temporal and spatial development of traffic conditions.

Making use of traffic flow models, model-based TSE conveys the principles observed in the physical world about the relationship between traffic state variables such as flow and density. There are several traffic flow models discussed in the literature. Continuous-time models include the Lighthill-Whitham-Richards (LWR) model, along with higher-order models such as Payne-Whitham (PW) model [36, 37], Aw-Rascle-Zhang (ARZ) model [38, 39]. Discretized models include cell transmission model (CTM) [34] and switching mode model (SMM) [35]. The first order LWR Model and its discretized version CTM model are the most used traffic flow models for their simplicity and accuracy. The LWR model and CTM are explained next.

3.2.1 Lighthill-Whitham-Richards Model

At a specified location $x$ and a designated time $t$, flow $q(x, t)$ and density $\rho(x, t)$ have the following relationship with the cumulative flow $N(x, t)$: Cumulative flow $N(x, t)$ depicts the number of vehicles which have passed location $x$ by the time $t$. Density $\rho(x, t)$ is the partial differential of cumulative count $N(x, t)$ with respect to $x$. Flow $q(x, t)$ is the partial differential of cumulative flow $N(x, t)$ with respect to $t$. These connections between variables are exhibited in (Equation 3.18).
\[
\begin{align*}
\rho(x,t) &= -\frac{\partial N(x,t)}{\partial x} \\
q(x,t) &= \frac{\partial N(x,t)}{\partial t}
\end{align*}
\]  

(3.18)

**LWR conservation law** is a continuity equation, which holds for all macroscopic models, and formulates the conservation of vehicles during traffic flow. This equation relates the change of density with the gradient of flow. When the flow is considered as a static function (portrayed by a fundamental diagram), it leads to a first order continuity equation, also referred to as Lighthill-Whitham-Richards (LWR) model [98] - which is a hyperbolic partial differential equation (PDE).

Various fundamental diagrams can be considered in conjunction with LWR models. Second-order models, such as the Payne-Whitham model [99], consider velocity change as a function of local density, local velocity, gradients, and other external factors [100]. For practical reasons, the LWR model is often discretized in both space and time using the Godunov discretization scheme. This discretized model is also referred to as the cell transmission model (CTM).

When the cumulative count \( N(x,t) \) is differentiable in both the time and space domain, Lighthill-Whitham-Richards (LWR) model [98] [101] relates \( N(x,t) \) with flow \( q(x,t) \) and density \( \rho(x,t) \). Equation (Equation 3.18) leads to the conservation law of traffic flow, which is referred to as LWR traffic flow partial differential equation (PDE).

The formulation of the LWR conservation law in the Eulerian coordinate system is shown in (Equation 3.19).

\[
\text{For } (x,t) \in \mathbb{R} \times \mathbb{R}^+ : \frac{\partial q(x,t)}{\partial x} + \frac{\partial \rho(x,t)}{\partial t} = 0
\]

(3.19)

It is worth noting that data provided by intelligent vehicles, such as vehicle velocity \( v(n,t) \) and spacing \( s(n,t) \) with respect to vehicle \( n \) and time \( t \), would be in Lagrangian coordinates [13]. The Lagrangian formulation of the conservation law is shown in (Equation 3.20).
For \((n, t) \in \mathbb{Z}^+ \times \mathbb{R}^+\):
\[
\frac{\partial v(n, t)}{\partial n} + \frac{\partial s(n, t)}{\partial t} = 0
\]  
(3.20)

**Generalized Solutions of LWR**

For a conservation law,
\[
\rho_t + [q(\rho)]_x = 0
\]  
(3.21)

with the initial condition,
\[
\rho(x, 0) = \rho_0(x),
\]  
(3.22)

where \(\rho_0 \in L^1_{loc} (\mathbb{R}; \mathbb{R})\), for a given smooth vector field \(q : \mathbb{R} \to \mathbb{R}\) the solution in the distributional sense is defined as follows [102]:

**Definition 2.1** A measurable locally integrable function \(\rho(x, t)\) is a solution in the distributional sense of the Cauchy problem (Equation 3.21) if for every \(\phi \in C^\infty_0 (\mathbb{R} \times \mathbb{R}^+) \to \mathbb{R}\)
\[
\iiint_{\mathbb{R} \times \mathbb{R}^+} [\rho(x, t) \phi_t(x, t) + q(\rho(x, t)) \phi_x(x, t)] \, dx \, dt + \int_{\mathbb{R}} \rho_0(x) \phi(x, 0) \, dx = 0.
\]  
(3.23)

**Weak Solutions**

A measurable locally integrable function \(\rho(x, t)\) is a weak distributional solution of the Cauchy problem (Equation 3.21) if it is a distributional solution in \((0, T) \times \mathbb{R}\) satisfying (Equation 3.22) and if \(\rho\) is a continuous function from \([0, T]\) into \(L^1_{loc}\). Assume \(\rho(x, t) = \rho(x^+, t)\), then the continuity condition can be formulated as
\[
\lim_{t \to 0+} \int_{\mathbb{R}} |\rho(x, t) - \rho_0(x)| \, dx = 0
\]  
(3.24)
These conditions give meaning to sets of measure zero for functions in $L^1$. For an $L^1$ function established on the two dimensions $(x, t)$, the domain given by $t = 0$ is a set of measure zero.

Therefore, to implement an initial function in this context we necessitate the $L^1$ continuity of the map $t \to \rho(\cdot, t)$ as exhibited in equation (Equation 3.24). Furthermore, we also necessitate establishing the function’s boundary value for a given $x$ value for a fixed time $t$. This is achieved by requiring $\rho(x, t) = \rho(x^+, t) = \lim_{\omega \to x^+} \rho(\omega, t)$. For further details refer to [102].

The traffic density equation should be consistent with the entropy Kružkov solution (see [103]).

**Definition 2.2 (Kružkov Solution)** The Kružkov entropy solution is a function $\rho : [0, \infty) \to L^1_{\text{loc}}(\mathbb{R})$, such that $\forall k \in \mathbb{R}, \phi > 0 \in C^0(\mathbb{R} \times \mathbb{R}^+)$ with the compact support of $\phi$ is in $t > 0$, assuming the flow $q$ locally lipschitz, we have

$$\iint [\|\rho - k\|\phi_t + (q(\rho) - q(k)) \text{sgn}(\rho - k)\phi_x]dxdt \geq 0$$

and there exists a set $E$ of zero measure on $[0, T]$, such that for $t \in [0, T] - E$, the function $\rho(x, t)$ is defined almost everywhere in $\mathbb{R}$, and for any ball $K_r = \{|x| \leq r\}$

$$\lim_{t \to 0^-} \int_{K_r} |\rho(x, t) - \rho_0(x)|dx = 0.$$  \hspace{1cm} (3.26)

It is essential to point out that Kružkov entropy solutions have been proved equivalent to vanishing viscosity solutions for hyperbolic conservation laws [102, 104].

### 3.2.2 Cell Transmission Model

Cell transmission model (CTM) is a discretized model which represents the traffic by dividing the road into consecutive homogeneous sections (cells) [34]. Each cell is created with the equal length $l = v_f \cdot \Delta t$, in which $v_f$ is the free-flow speed. Vehicle flow enters the cell downstream as shown in Figure 3.3: $y_{i-1,i}(k)$ represents the flow from upstream cell $i - 1$ to downstream cell $i$ at time step $k$, and $y_{i,i+1}(k)$ represents the flow from upstream cell $i$ to downstream cell $i + 1$ at time step $k$.  

31
\( n_{i-1}(k) \), and \( n_{i+1}(k) \) conveys the numbers of vehicle in each cell at time step \( k \). The resultant flow at the junction of two cells is determined using Godunov approach described in the later sections.

The flow of vehicles between the cells and the number of vehicles present in them comply with (Equation 3.27), in which \( n_i(k) \) represents the number of vehicles present in cell \( i \) at time step \( k \); and \( y_{i-1,i}(k) \) represents the vehicle flow from cell \( i-1 \) to \( i \) at time step \( k \). For \( i \in \mathbb{Z}^+, k \in \mathbb{Z}^+ \):

\[
 n_i(k + 1) - n_i(k) = y_{i-1,i}(k) - y_{i,i+1}(k) \tag{3.27}
\]

CTM was initially proposed using a trapezoidal fundamental diagram, in which the flow-density relationship is formulated as (Equation 3.28). The road capacity is \( q_m \), and the backward wave speed is \(-w\).

\[
 q = \min\{\rho v_f, q_m, w(\rho_m - \rho)\} \tag{3.28}
\]

3.3 Fundamental Diagram of Traffic Flow

The relationship between the macroscopic traffic flow variables is referred to as the fundamental diagram. It is essential to mention that the fundamental diagram is not a physical law. It varies with the road, the driver behavior, the vehicles, and driving conditions such as lighting. Several fundamental diagrams exist and are used as applicable in the situation. Some of the commonly-used ones are: piece-wise affine speed-density relationship [35], Greenshield’s model [105], triangular fundamental diagram [106], and inverse-lambda shaped fundamental diagram [107].

The fundamental diagram of traffic flow gives the relationship between traffic state variables
q, v, and ρ, allowing us to hypothesize about the average driving behavior on the observed road segment [100].

3.3.1 Greenshields Model

Greenshields fundamental diagram is one of the most used and simplest models of traffic flow theory. It makes a simplifying assumption that the mean velocity has a linear relationship with the density. The relationship between traffic variables is described in (Equation 3.29), where ρ_m is the maximum density (also known as jam density), and v_f is the free-flow speed. The relationship is also shown in Figure 3.4.

![Greenshields' Fundamental Diagram](image)

Figure 3.4: Greenshields’ Fundamental Diagram

33
\[
\begin{align*}
q(x, t) &= \rho(x, t) v_f \left( 1 - \frac{\rho(x, t)}{\rho_m} \right) \\
v(x, t) &= v_f \left( 1 - \frac{\rho(x, t)}{\rho_m} \right)
\end{align*}
\] 

(3.29)

3.3.2 Daganzo Model

An alternative model has been offered by Daganzo [108] to represent the flow-density relationship by two straight lines, with the introduction of a third parameter - critical capacity \(q_c\) - at which the maximum flow is reached. Daganzo’s fundamental diagram is presented in Figure 3.5 and the formulation is shown in (Equation 3.30).

---

Figure 3.5: Daganzo’s Fundamental Diagram
\[ q(\rho) = \begin{cases} 
\frac{q_c \rho}{\rho_c} & \text{if } \rho \leq \rho_c, \\
q_c \left(1 - \frac{\rho - \rho_c}{\rho_m - \rho_c}\right) & \text{if } \rho > \rho_c,
\end{cases} \quad (3.30) \]

3.3.3 Inverse-lambda Model

Capacity drop is often observed when traffic density reaches \( \rho_c \) [109] [110]. A modification of Daganzo’s fundamental diagram - inverse-lambda shaped fundamental diagram - is proposed to reflect this phenomenon [111]. The modified fundamental diagram is exhibited in Figure 3.6 and the relationship between the traffic states are given in (Equation 3.31). The adjusted parameter \( q_c \) now has two values - \( q_{c1} \) before the capacity drop, and \( q_{c2} \) after the drop.

Figure 3.6: Inverse-lambda Shaped Fundamental Diagram
\[ q(\rho) = \begin{cases} 
q_{c1} \frac{\rho}{\rho_c} & \text{if } \rho \leq \rho_c, \\
q_{c2} \left(1 - \frac{\rho - \rho_c}{\rho_m - \rho_c}\right) & \text{if } \rho > \rho_c, 
\end{cases} \] (3.31)

3.4 Nonlocal Traffic Flow Models

The traditional fundamental diagrams depict the relationship between local traffic state variables such as speed \( v \) and density \( \rho \). However, this idealistic association of state variables may not be practical in congested road segments due to drivers’ behavior [39]. Therefore, second-order traffic flow models are developed to represent the conservation connection between density and speed [38]. Another branch of research tries to tackle this problem of the unrealistic representation of traffic state relationship with the introduction of nonlocal traffic state [112, 113]. It applies a convolutional kernel with the traffic density downstream to produce a weighted average value of density and relates it to local speed [114]. This type of modeling is developed with the consideration of drivers regulating the speed of vehicles due to the perception of traffic conditions ahead. A Lax-Friedrichs-type numerical method can be used to solve for the solution of conservation laws with nonlocal physics [115]. The potential downside is the computation cost of the numerical scheme used to solve the nonlocal conservation laws since the flux function is in the integral form. Alternatively, a discontinuous Galerkin scheme can be adapted to approximate the nonlocal one-dimensional traffic conservation laws [116].

Through the fundamental diagram (FD), the local velocity \( v(x, t) \) at location \( x \) and time \( t \) can be determined by the local density \( \rho(x, t) \). Therefore \( v(x, t) \) can also be expressed as \( v(\rho(x, t)) \). With the nonlocal version of traffic states, the velocity at location \( x \) and time \( t \) is regulated by the nonlocal density \( \rho_n \) (hence the notation “n”), which is obtained by a convolution of density conditions within the perception window. The scenario is illustrated in Figure 3.7 and mathematical formulation is presented in (Equation 3.32).
\[ \rho_n(x, t) = \rho(x, t) * \theta(x) = \int_{x}^{x+w} \rho(\tau, t) \theta(\tau - x) d\tau \]

(3.32)

\[ v(x, t) = v(\rho_n(x, t)) \]

The Greenshields’ FD in (Equation 8.2) is subsequently modified to reflect the relationship between the local velocity and the nonlocal density, given in (Equation 3.33).

\[ \begin{align*}
q(x, t) &= \rho(x, t) v_f \left( 1 - \frac{\rho_n(x, t)}{\rho_m} \right) \\
v(x, t) &= v_f \left( 1 - \frac{\rho_n(x, t)}{\rho_m} \right)
\end{align*} \]

(3.33)

Observe that in (Equation 3.33), the local velocity \( v(x, t) \) has the linear relationship with the convoluted nonlocal density \( \rho_n(x, t) \), instead of the local density \( \rho(x, t) \).

The LWR conservation law with nonlocal traffic states is expressed in (Equation 3.34) with respect to density \( \rho(x, t) \).

\[ \frac{\partial \rho(x, t)}{\partial t} + v_f \left( 1 - \frac{\rho_n(x, t)}{\rho_m} \right) \frac{\partial \rho(x, t)}{\partial x} - \rho(x, t) v_f \frac{\partial \rho_n(x, t)}{\partial x} = 0 \]

(3.34)

Note that in (Equation 3.34), the nonlocal density \( \rho_n(x, t) \) and the local density \( \rho(x, t) \) are treated as two separate variables. The partial derivative terms with respect to \( x \) are calculated...
separately. Next, the kernels used for calculating the convolution of density are explained.

3.4.1 Kernel Functions

In (Equation 3.32), $\theta(x)$ represents the kernel function in the perception window (hence the notation “w”) and has its property listed in (Equation 3.35). The perception window is also referred to as look-ahead window.

$$\int_{x_0}^{x_0+w} \theta(x) dx = 1 \quad (3.35)$$

Two kernel functions — the constant kernel, and the linearly decreasing kernel are formulated below, in (Equation 3.36) and (Equation 3.37) respectively, and are illustrated in Figure 3.8 and Figure 3.9.

*The Constant Kernel*

When applying the constant kernel, the nonlocal traffic density is the average of the density values in the perception window.

$$\theta(x) = \frac{1}{w}, \quad x \in [x_0, x_0+w] \quad (3.36)$$

Figure 3.8: The Constant Kernel
The Linearly Decreasing Kernel

When applying the linearly decreasing kernel, the nonlocal traffic density is a weighted average of the density values in the perception window, with the density value at locations closer to the vehicle being assigned a greater weight.

\[
\theta(x) = \frac{2}{w} \left( 1 - \frac{x - x_0}{w} \right), \quad x \in [x_0, x_0 + w]
\] (3.37)

Figure 3.9: The Linearly Decreasing Kernel
CHAPTER 4
PHYSICS-INFORMED DEEP LEARNING

This chapter\(^1\) covers the basic concepts of deep learning architecture and the design of a physics-informed deep learning neural network.

4.1 Deep Learning Neural Network

Development in Deep learning (DL) neural networks has made it an appropriate tool in the computational modeling of a physical system, which is often governed by complex non-linear functions [117]. A feed-forward deep learning neural network consists of numerous hidden layers between the input layer and the output layer. Computation units on each layer, mirroring the functions of biological neurons, enable the neural network learning capacity of a system’s non-linear dynamics. The architecture of a typical deep learning feedforward neural network is drawn in Figure 4.1.

Figure 4.1: Architecture of a Deep Learning Neural Network

---

Connections between adjacent layers can be expressed as (Equation 4.1), in which $z_{i+1}$ is the output of neurons on the $i+1$ layer, $\sigma(\cdot)$ represents the activation function, and $W_{i+1}$ and $b_{i+1}$ are the weight matrix and bias vector in the $i+1$ layer, respectively.

$$\forall i \in [1, n], \quad z_{i+1} = \sigma(W_{i+1}z_i + b_{i+1})$$ (4.1)

The training process is driven by the motivation of minimizing a cost function, which is built to measure the inaccuracy of the learning outcome in restoring the original system dynamics. When a mean square error (MSE) is used as the measurement of cost in a deep learning neural network, the cost function can be formulated as (Equation 4.2), in which $N$ is the number of predictions, and $\hat{u}(x, y, z, t)$ is the prediction of the variable $u(x, y, z, t)$.

$$J = MSE_{(\hat{u}(x, y, z, t), u(x, y, z, t))} = \frac{1}{N} \sum_{i=1}^{N} |u(x, y, z, t) - \hat{u}(x, y, z, t)|^2$$ (4.2)

Each neuron has its own weight $w$ and bias $b$, which constitute the weight matrix $W$ and bias vector $b$ on each hidden layer. During the training process, the $W$ and $b$ are repeatedly updated in search of a smaller cost. A cost threshold $\varepsilon$ is established to direct the termination of training, once a satisfactory cost, small than $\varepsilon$, is obtained. The training process of a deep learning neural network on variable $u(x, y, z, t)$ is demonstrated in Figure 4.2.

Automatic Differentiation (AD), also called algorithmic differentiation, is a set of techniques for precisely and efficiently evaluating the derivative of numeric functions specified by the computational algorithm. It substitutes the domain of the variables to assimilate the derivative values and replaces the operators per the chain rule to propagate the derivatives [118].

AD addresses the weakness of alternative groups of computation methods: susceptibility of error in manual differentiation [119] and numerical differentiation, due to round-off and truncation errors [120]; and the enigmatic, complex expression resulting from symbolic differentiation [121]. To deploy the standard optimization methods such as LM-BFGS in deep learning, AD produces
Figure 4.2: Training Process of a Deep Learning Neural Network

quantitative derivative evaluations instead of expressions, benefiting the computational accuracy and efficiency [118].

**Optimization of a Deep Learning Neural Network:** There are many optimization algorithms to train a neural network, gradient descent is one of the popular algorithms to perform optimization [122]. It minimizes the cost function $J(\theta)$ by updating the parameters $\theta \in \mathbb{R}$ in the opposite direction of the gradient of $J(\theta)$ with respect to the parameters. Batch gradient descent (BGD), mini-batch gradient descent (m-BGD), and Stochastic gradient descent (SGD) are some of the most common methods in the groups of techniques used in search of local minimum.

BGD is formulated as (Equation 4.3). The step it takes to update the parameters is the learning rate $\eta$. It calculates each update using the entire training set; this can be very slow, and it doesn’t allow online model updates.

$$
\Delta \theta = - \eta \cdot \nabla \theta J(\theta)
$$

(4.3)

SGD circumvents BGD’s problem of redundancy by updating the parameters for each training sample. However, the frequent corrections cause high levels of fluctuation in the value of the cost function. m-BGD takes advantage of both algorithms and updates the parameters using a mini-
batch of training samples, reducing the variance of samples and the time of convergence. Brief information on the training approaches used in this work is provided below:

1. **L-BFGS-B**: Limited memory, boundary constraints Broyden–Fletcher–Goldfarb–Shanno algorithm [123] is one of the default optimizers of `scipy.optimize.minimize` in the scientific computing library SciPy [124]. Under the default setting, the optimization process terminates when the difference of cost $f_{itol}$ between iterations is less than $2.22\times10^{-16}$.

2. **Adam**: Adaptive moment estimation (Adam) [125] takes advantage of the Momentum [126] and the RMSProp [127] optimization algorithms by monitoring the accumulation of both the gradient and the squared gradient, using $\beta_1$ and $\beta_2$ as the decay terms shown in (Equation 4.4).

\[
\Delta \theta_{i,t} = \nabla J(\theta_{i,t}) \\
G_{i,t} = \beta_1 G_{i,t-1} + (1 - \beta_1) \Delta \theta_{i,t} \\
E_{i,t} = \beta_2 E_{i,t-1} + (1 - \beta_2)(\Delta \theta_{i,t})^2 \\
\theta_{i,t+1} = \theta_{i,t} - \alpha \frac{E_{i,t}}{G_{i,t} + \epsilon} \Delta \theta_{i,t}
\]  

(A.4.4)

A few essential components that warrant contemplation when training a DL neural network are as follows:

**Learning Rate** - It determines the steps in adjusting the value of weights in the neural network. A relatively large learning rate has the advantage of quicker convergence, producing the prediction results in a shorter period. However, it may omit the better weight configuration and therefore sacrifice the output accuracy.

**Network Size** - The network size is determined by the number of layers $n_l$ and the number of hidden units $n_h$. It reflects the complexity of the neural network, which also has an impact on
output accuracy. A complex neural network may be beneficial in yielding accurate results, but it may also be prone to over-fitting \[128\], and generally takes a long time for convergence.

**Convergence Time** - is the time a neural network takes to converge; this measure is included as estimation of traffic states in real-time is preferable in traffic management \[42\]. A longer computation time from a complex neural network limits its adaptability for application.

Deep learning for TSE can swiftly modify its weights configuration based on real-time data. However, sensor and detection bias may be introduced in the estimation due to the limited availability of traffic data, hindering the reliability of learning results. Besides, computation capacity is critical to process the extensive incoming traffic data promptly. It compels constructing the computational infrastructure of a higher standard, including data servers and processing centers, to support this data-driven approach.

**Relative Percent L\(_2\) Error** - measures the accuracy of neural network output. Differ from the cost function, this error metric is calculated after the completion of neural network training, based on the neural network’s performance with the test data. A normalized error measurement using Frobenius norm is shown in (Equation 4.5). \(P\) is the matrix form of vehicle density \(\rho(x,t)\), and \(\hat{P}\) is the estimation of \(P\), where \((x,t) \in \mathbb{R} \times \mathbb{R}^+\). Let \(N_1\) and \(N_2\) be the number of bins after discretizing the density field in space and time, respectively. I.e., \(N_1 \cdot N_2\) is the total number of grid points to be estimated. Note that this error formulation is also referred to as root mean square percentage error (RMPSE).

\[
L^2_{\text{error}} = \frac{\|P - \hat{P}\|_F}{\|P\|_F} \times 100\% = \sqrt{\sum_{j=1}^{N_1 \cdot N_2} \left| \hat{\rho}(x(j), t(j)) - \rho(x(j), t(j)) \right|^2} \times 100\% \quad (4.5)
\]

**Regularization** - Regularization is a group of methods used to prevent over-fitting when fitting a function or a model appropriately on the given training set \[129\]. With the learned set of parameters \(W\), a regularization term can be formulated as (Equation 4.6). \(M\) represents the number of parameters in \(W\). The associated penalty \(E\) (cost) becomes L1-norm (Lasso regularization) when
\( q = 1 \) and L2-norm (Ridge regularization) when \( q = 2 \).

\[
E = \sum_{j=1}^{M} |w_j|^q
\]  

(4.6)

### 4.2 Physics-informed Deep Learning

Physics-informed deep learning (PIDL) is a type of DL method where a neural network is trained to solve learning tasks while respecting the law of physics [29]. With the inherent physical laws encoded as a priori knowledge, the resulting neural network forms data-efficient approximators to process input information and give reconstruction/prediction results [72]. When input data is inadequate, scanty or noisy, PIDL allows the neural network to make full use of the data with the aid of physics, which explains the underlying relationship in the data and improves the prediction results.

To evaluate the outputs from the neural network in terms of compliance with the governing physical laws, the physics cost is computed at a set of spatiotemporal points \((x, y, z, t)\), termed as collocation points, which can be chosen by Latin hypercube sampling (LHS) [130]. To differentiate the deep learning cost (DL-cost) in (Equation 4.2) and the physics-cost, \( J_{DL} \) is used to depict the DL-cost, and \( J_{PHY} \) is the penalty of incompliance to the physics. \( J_{DL} \) and \( J_{PHY} \) are computed individually in (Equation 4.7). \( N_u \) symbolizes the number of training samples, and \( N_f \) is the number of collocation points.

\[
\begin{align*}
J_{DL} &= \frac{1}{N_u} \sum_{k=1}^{N_u} |u(x, y, z, t) - \hat{u}(x, y, z, t)|^2 \\
J_{PHY} &= \frac{1}{N_f} \sum_{k=1}^{N_f} |f(x, y, z, t)|^2
\end{align*}
\]  

(4.7)

In (Equation 4.7), cost function \( f(x, y, z, t) \) is configured to quantify the non-compliance of the physics. The training process of a PIDL neural network is shown in Figure 4.3. The exemplary cost function of \( J_{PHY} \) is given as \( f = \lambda_1 u + \lambda_2 u_x + \lambda_3 u_{xy} + \lambda_4 u_{zz} + \lambda_5 u_t \). The cost terms \( u_x, u_{xy}, u_{zz}, \),...
and \( u_r \) are partial derivatives of the output \( \hat{u}(x, y, z, t) \) with respect to a few combinations of input coordinates \((x, y, z, t)\). The \( \lambda \) parameters represent the weights of cost terms.

Figure 4.3: Training Process a Physics-informed Deep Learning (PIDL) Neural Network

The cost function of PIDL includes the DL-cost in (Equation 4.2) and the physics-cost \( J_{PHY} \), which examines the compliance of governing law. Hyperparameter \( \mu \) is introduced to adjust the weights of \( J_{DL} \) and \( J_{PHY} \) and the cost function \( J \) of PIDL is given as (Equation 4.8).

\[
J = \mu \cdot J_{DL} + (1 - \mu) \cdot J_{PHY}
\]

(4.8)

where parameter \( \mu \) balances the weights of \( J_{DL} \) and \( J_{PHY} \) in the cost function.
CHAPTER 5
DEVELOPING PIDL PARADIGM FOR TSE

PIDL\textsuperscript{1} empowers a deep learning neural network with the system’s governing physical laws as a priori knowledge [29]. The fundamental diagram of traffic flow and the conservation law serve as meaningful know-how in training a neural network to recognize the underlying relationship among traffic variables. Given the unique advantage of physics-informed deep learning (PIDL) in efficiently utilizing limited input data and the physics, this chapter provides details of the PIDL approach for traffic state estimation (TSE) by using the LWR PDE (conservation law of traffic flow) with Greenshields’ and the inverse-lambda fundamental diagrams; and the cell transmission model. Other physical models, such as second order flow models or discretized first order models can be obtained by following the steps laid out in this chapter.

5.1 Approach Overview

The PIDL neural network gains the knowledge of the governing physical law by incorporating the non-compliance cost of conservation law $J_{PHY}$ into the cost function $J$. The training iteration is repeated if the sum of the estimation cost and physics cost is greater than the designated threshold. Otherwise, the fine-tuned PIDL gives the estimation as output. A maximum of allowed learning iterations - $i_{\text{max}}$ is set to prevent the learning process from running eternally in the event of no change in total cost. These steps are graphically presented in Figure 5.1.

The design of the cost function is one of the primary components of PIDL. Next section illustrates the design of PIDL paradigm using LWR and CTM traffic flow models.

5.2 Formulation using LWR Model

The conservation law of traffic flow described in (Equation 3.19) establishes the relationship of traffic state variables with respect to location \( x \) and time \( t \). It needs to be integrated into the learning process of a PIDL neural network. The process of computing the DL-cost term utilizes the observed data \( O = \{(x^j, t^j) | j = 1, 2, \cdots, N_o\} \), whereas the physics cost is computed on the collocation points \( C = \{(x^k, t^k) | k = 1, 2, \cdots, N_c\} \).

Note that since the DL estimates the entire grid, there is no restriction on the number of collocation points, meaning that collocation points (where the physics cost is computed) are not restricted to the observation points, rather they can be any subset of the entire grid points. The cost terms are described as (Equation 5.1), where \( \hat{\rho}(x^j, t^j) \) is the estimated density by the DL component of the PIDL approach.

\[
\begin{align*}
J_{DL} &= MSE(\rho(x,t), \hat{\rho}(x,t)) = \frac{1}{N_o} \sum_{j=1}^{N_o} \left| \rho(x^j, t^j) - \hat{\rho}(x^j, t^j) \right|^2 \\
J_{PHY} &= MSE(0, \frac{\partial \hat{q}(x,t)}{\partial x} + \frac{\partial \hat{\rho}(x,t)}{\partial t}) = \frac{1}{N_c} \sum_{k=1}^{N_c} \left| \frac{\partial \hat{q}(x^k, t^k)}{\partial x} + \frac{\partial \hat{\rho}(x^k, t^k)}{\partial t} \right|^2
\end{align*}
\] (5.1)
5.2.1 LWR and Greenshields’ model

Plugging the relationship between variables $\rho$, $v$, and $q$ from (Equation 3.29) into the Eulerian formulation of conservation law in (Equation 3.19), the physical law can be written as (Equation 5.2) and (Equation 5.3).

For $(x, t) \in [x_0, x_n] \times [t_0, t_m]$,

$$
\rho_m \left( 1 - \frac{2v(x, t)}{v_f} \right) \frac{\partial v(x, t)}{\partial x} - \frac{\rho_m}{v_f} \frac{\partial v(x, t)}{\partial t} = 0 \tag{5.2}
$$

For $(x, t) \in [x_0, x_n] \times [t_0, t_m]$,

$$
v_f \left( 1 - \frac{2\rho(x, t)}{\rho_m} \right) \frac{\partial \rho(x, t)}{\partial x} + \frac{\partial \rho(x, t)}{\partial t} = 0 \tag{5.3}
$$

Observe that both the equations provide the same physical law - the only difference is their dependent variable. Equation (Equation 5.2) formulates the law in terms of velocity $v(x, t)$, whereas (Equation 5.3) formulates it in terms of density $\rho(x, t)$.

Consequently, one can formulate the cost function of a PIDL neural network reconstructing a density-field $\rho(x, t)$ as (Equation 5.4):

$$
\begin{align*}
J_{DL} &= \frac{1}{N_o} \sum_{i=1}^{N_o} \left| \rho(x_{o_i}, t_{o_i}) - \hat{\rho}(x_{o_i}, t_{o_i}) \right|^2 \\
J_{PHY} &= \frac{1}{N_c} \sum_{k=1}^{N_c} \left| v_f \left( 1 - \frac{2\hat{\rho}(x_{c_k}^k, t_{c_k}^k)}{\rho_m} \right) \frac{\partial \hat{\rho}(x_{c_k}^k, t_{c_k}^k)}{\partial x} + \frac{\partial \hat{\rho}(x_{c_k}^k, t_{c_k}^k)}{\partial t} \right|^2
\end{align*} \tag{5.4}
$$

And the cost function of a PIDL neural network in velocity-field is composed as (Equation 5.5).

$$
\begin{align*}
J_{DL} &= \frac{1}{N_o} \sum_{i=1}^{N_o} \left| v(x_{o_i}, t_{o_i}) - \hat{v}(x_{o_i}, t_{o_i}) \right|^2 \\
J_{PHY} &= \frac{1}{N_c} \sum_{k=1}^{N_c} \rho_m \left( 1 - \frac{2\hat{v}(x_{c_k}^k, t_{c_k}^k)}{v_f} \right) \frac{\partial \hat{v}(x_{c_k}^k, t_{c_k}^k)}{\partial x} - \frac{\rho_m}{v_f} \frac{\partial \hat{v}(x_{c_k}^k, t_{c_k}^k)}{\partial t} \right|^2
\end{align*} \tag{5.5}
$$
Both (Equation 5.2) and (Equation 5.3) are hyperbolic PDEs. A second order diffusive term can be added to make the PDE become parabolic and secure a strong solution. For example, (Equation 5.3) will become (Equation 5.6) where $\varepsilon$ is a constant of a small value.

For $(x, t) \in [x_0, x_n] \times [t_0, t_m]$,

$$v_f \left( 1 - \frac{2 \rho(x, t)}{\rho_m} \right) \frac{\partial \rho(x, t)}{\partial x} + \frac{\partial \rho(x, t)}{\partial t} = \varepsilon \frac{\partial^2 \rho(x, t)}{\partial x^2}$$  \hspace{1cm} (5.6)

The second order diffusion term ensures the solution of PDE is continuous and differentiable, avoiding the breakdown and discontinuity in the solution to the PDE.

From the cost function of PIDL (Equation 4.8), the physics-cost term can be viewed as a regularization agent of the learning process [131]. It prevents over-fitting of the training samples in both estimation and prediction tasks by penalizing updates of parameters under which the reconstruction doesn’t satisfy the physical law.

### 5.2.2 LWR and inverse-lambda model

When using the inverse-lambda fundamental diagram, given the relationship between traffic state variables flow $q$ and density $\rho$ in (Equation 3.31), the physics cost $J_{PHY}$ can be formulated as (Equation 5.7). Note that the DL cost ($J_{DL}$) remains the same as shown previously for the Greenshields’ model. The collocation points in this case are separated into two subsets $C_1 = \{(x_{c1}^v, t_{c1}^v) | v = 1, 2, \cdots, N_{c1}\}$ where $\hat{\rho}(x_{c1}^v, t_{c1}^v) \leq \rho_c$, and $C_2 = \{(x_{c2}^w, t_{c2}^w) | w = 1, 2, \cdots, N_{c2}\}$ where $\hat{\rho}(x_{c2}^w, t_{c2}^w) > \rho_c$. 

50
\[
\begin{align*}
J_{PHY1} &= \frac{1}{N_{c1}} \sum_{v=1}^{N_{c1}} \left[ \frac{\partial \hat{\rho}(x_{c1}^v, t_{c1}^v)}{\partial t} + \frac{q_{c1}}{\rho_c} \frac{\partial \hat{\rho}(x_{c1}^v, t_{c1}^v)}{\partial x} \right]^2 \\
J_{PHY2} &= \frac{1}{N_{c2}} \sum_{w=1}^{N_{c2}} \left[ \frac{\partial \hat{\rho}(x_{c2}^w, t_{c2}^w)}{\partial t} - \frac{q_{c1}}{\rho_m - \rho_c} \frac{\partial \hat{\rho}(x_{c2}^w, t_{c2}^w)}{\partial x} \right]^2 \\
J_{PHY} &= \frac{N_{c1} \cdot J_{PHY1} + N_{c2} \cdot J_{PHY2}}{N_{c1} + N_{c2}}
\end{align*}
\]

(5.7)

Note that \(N_{c1}\) and \(N_{c2}\) are the number of collocation points in the free flow region and congested region, respectively; \(\hat{\rho}(x_{c1}^v, t_{c1}^v)\) and \(\hat{\rho}(x_{c2}^w, t_{c2}^w)\) are the estimated density by the DL component of the PIDL approach.

### 5.3 Formulation using Cell Transmission Model (CTM)

This section presents the cost function formulation using the discrete CTM. Note that it can be coupled with any suitable fundamental diagram. Recall that CTM solves the LWR PDE through a finite difference scheme, also referred to as Godunov’s numerical method. Consider a spatial-temporal representation of traffic states illustrated in Figure 5.2 where the colors denote the variety of traffic state values. Let \(N_1\) and \(N_2\) be the number of bins after discretizing the density field in space and time, respectively. Thus, \(N_1 \cdot N_2\) is the total number of grid points to be estimated.

Making the algorithm conform to the CTM model will lead to the following physics cost in (Equation 5.8):

\[
J_{PHY} = \frac{1}{N_1 \cdot N_2} \sum_{i=1}^{N_1} \sum_{k=1}^{N_2} \left[ \hat{n}_i(k+1) - \hat{n}_i(k) \right] \left[ \hat{y}_{i-1,i}(k) - \hat{y}_{i,i+1}(k) \right]^2
\]

(5.8)

CTM assumes uniform density \(\rho\) in cells at each time step \((n_i(k) = \rho_i(k) \cdot \Delta x)\) and uniform flow \(q\) between cells during each time interval \((y_{i-1,i}(k) = q_{i-1,i}(k) \cdot \Delta t)\). Therefore, the DL-cost and the physics-cost with CTM become (Equation 5.9):

\[
J_{PHY} = \frac{N_{c1} \cdot J_{PHY1} + N_{c2} \cdot J_{PHY2}}{N_{c1} + N_{c2}}
\]
Figure 5.2: Discretization of space-time in CTM

\[
\begin{align*}
J_{DL} &= \frac{1}{N_o} \sum_{j=1}^{N_o} |\rho(x_j^i, t_j^i) - \hat{\rho}(x_j^i, t_j^i)|^2 \\
J_{PHY} &= \frac{1}{N_1 \cdot N_2} \sum_{i=1}^{N_1} \sum_{k=1}^{N_2} |[\hat{\rho}_i(k + 1) - \hat{\rho}_i(k)] \cdot \Delta x - [\hat{q}_{i-1,i}(k) - \hat{q}_{i,i+1}(k)] \cdot \Delta t|^2 
\end{align*}
\]

(5.9)

Similar to the previous discussion on the LWR model, the cost function of PDDL is comprised of the DL-cost \( J_{DL} \) and the physics cost \( J_{PHY} \) given in (Equation 5.9). Note that \( J_{DL} \) is computed in a similar fashion using \( N_o \) observations. Whereas, the formulation of \( J_{PHY} \) differs from the LWR case. Also, unlike the case of LWR, here the entire grid \( (N_1 \cdot N_2) \) is used as collocation points. This makes the representation easier, however, a subset of \( N_1 \cdot N_2 \) can also be used. Hyperparameter \( \mu \) can adjust the weights of \( J_{DL} \) and \( J_{PHY} \) as shown in (Equation 4.8). The Godunov numerical solution, which computes the flow \( q_{i-1,i} \) between the adjacent cells \( i - 1 \) and \( i \) is described next.
5.3.1 Godunov Numerical Solution

The density conditions of both cells determine the flow rate at the boundary of two adjacent cells. It can be computed using Godunov’s numerical scheme [24, 132]. The Godunov method discretizes the first-order traffic flow models such as the LWR model by solving the Riemann problem with the initial condition of upstream density $\rho_{i-1}$ and downstream density $\rho_i$ [104].

Either a shockwave or a rarefaction wave will originate from the junction of the two densities. A shockwave develops when $q'(\rho_{i-1}) > q'(\rho_i)$, and a rarefaction develops when $q'(\rho_{i-1}) < q'(\rho_i)$. The speed of the shockwave is given by (Equation 5.10):

$$s = \frac{dx_s(t)}{dt} = \frac{[q(\rho_{i-1}) - q(\rho_i)]}{\rho_{i-1} - \rho_i}$$  \hspace{1cm} (5.10)

where $x_s(t)$ is the position of the shockwave as a function of time. If the shock speed is positive, then the inflow at the junction between the two traffic densities will be a function of upstream traffic density. In contrast, if the shockwave speed is negative, then the inflow at the junction between the two traffic densities will be a function of downstream traffic density.

Let $Q(\cdot, \cdot)$ be a function denoting the flow at the boundary of two cells which can be determined as follows:

$$Q(\rho_{i-1}, \rho_i) = q(\rho^*(\rho_{i-1}, \rho_i))$$  \hspace{1cm} (5.11)

where $\rho^*$ denotes the flow-dictating density. $\rho^*$ is determined by the following relationship emanating from the Godunov numerical solution:
\begin{equation}
\rho^* = \begin{cases} 
\rho_{i-1}, & \text{if } q'(\rho_{i-1}), q'(\rho_i) \geq 0 \\
\rho_i, & \text{if } q'(\rho_{i-1}), q'(\rho_i) \leq 0 \\
\rho_{i-1}, & \text{if } q'(\rho_{i-1}) \geq 0 \geq q'(\rho_i) \\
& \text{and } q(\rho_{i-1}) > q(\rho_i) \\
\rho_i, & \text{if } q'(\rho_{i-1}) \geq 0 \geq q'(\rho_i) \\
& \text{and } q(\rho_{i-1}) < q(\rho_i) \\
\rho_c, & \text{if } q'(\rho_{i-1}) < 0 < q'(\rho_i)
\end{cases}
\end{equation}

Here \( \rho_c \) is the critical density.

Hence depending on the traffic densities on the left and right side of the junction, flow at the junction can have three possible values, i.e. \( Q(\rho_{i-1}, \rho_i) \) can have three distinct values, \( q(\rho_{i-1}) \), \( q(\rho_i) \), or \( q(\rho_c) \). Note that \( Q(\rho_{i-1}, \rho_i) = q_{i-1,i} \). Once the flow-dictating density \( \rho^* \) is identified, the corresponding flow \( q(\rho^*) \) can be obtained using any suitable fundamental diagram.
CHAPTER 6
APPLICATION AND VALIDATION

This chapter\(^1\) details the selection of training instances, and case studies to validate PIDL for TSE, in Eulerian, Lagrangian, and CAV settings. The discussion at the end of the chapter reflects back on the role of physics as a regularization agent in training the PIDL neural networks for TSE.

6.1 Selection of Learning Instances

Data collection and sensing may introduce potential biases, often due to human factors and sensing limitations. These biases can persist in the models that are trained on the data, highlighting the importance of selecting appropriate subsets of the data to minimize these biases. However, in many instances, the selection of training instances is limited by the availability and positions of the sensors. For traffic sensing, the sensors are either fixed on the roadside at fixed intervals (such as loop detectors) or they are moving with the traffic stream (like probe vehicles, CAVs).

The sampling cases of traffic state data are demonstrated in Figure 6.1. Numerical PDE solvers such as Lax-Friedrichs’ numerical scheme [133] can be used as a state reconstruction tool; however, it requires the complete information on the initial and boundary conditions as shown in Figure 6.1a which is not practically feasible. On the other hand, the PIDL approach can utilize any given amount of inputs from the boundaries for training; in Figure 6.1b, 20% of the initial and boundary data are shown as an exemplar training setting. Figure 6.1c represents the Eulerian traffic data that can be gathered from roadside sensors or loop detectors installed at predetermined

---

locations along the road infrastructure (shown at \( x = 0, 0.25, 0.5, 0.75, 1.0 \) in this instance). Finally, Figure 6.1d exhibits the Lagrangian traffic data that can be collected by connected vehicles, which can muster traffic state information at various locations along the vehicle trajectories. Notice the gaps between the Eulerian data in Figure 6.1c; these represent the occasional sensor failures or malfunctions, resulting in data loss in this scenario.

### 6.2 PIDL with Eulerian and Lagrangian Sensing

Given the nature of sparsity in traffic data, the ability to gain accurate perception with limited input is highly valuable. In this section, experiments are conducted to validate the capability of the PIDL neural network for TSE. For comparison, a DL neural network is built with the same learning architecture — matching number of layers \( n_l \), unchanged number of neurons in each hidden layer \( n_h \) — and tasked with the identical estimation task. The mere difference is that PIDL has the a priori knowledge of the underlying physics, i.e. conservation law of traffic flow. Physics-cost, \( J_{PHY} \), is integrated into the cost function of PIDL.

#### 6.2.1 Data Description

A synthetic traffic dataset on a road segment is generated by using the Lax-Hopf method as described in [134], under a no upstream flow and no downstream flow condition. Traffic density value \( \rho(x,t) \) is created for time \( t \in [0, 50] \) seconds and location \( x \in [0, 1000] \) meters. The temporal resolution of this dataset is 0.1-second, and the spatial resolution is 2-meter. In other words, there are 501 timestamps: \( t = 0, 0.1, 0.2, \ldots, 50 \); and 501 locations: \( x = 0, 2, 4, \ldots, 1000 \).

Because there is no upstream and downstream flow, the boundary conditions are formulated as follows:

\[
\frac{\partial q(0, t)}{\partial x} = \frac{\partial q(1000, t)}{\partial x} = 0
\]  

(6.1)

The free-flow speed \( v_f \) is set as 30 meters per second (67 miles per hour) and maximum
Figure 6.1: Selection of Learning Data Instances
density $\rho_m$ as 0.1 vehicles per meter (161 vehicles per mile). The relationship between traffic states (speed $v$, flow $q$, and density $\rho$) is modeled using Greenshields’ relationship as described in (Equation 3.29). The generated synthetic dataset is shown in Figure 6.2.

In the experiment, the value of parameter $\mu$ in (Equation 4.8) is set as 0.5 to give equal weight to $J_{DL}$ and $J_{PHY}$ in calculating the cost $J$. An 11-layer architecture is utilized with 50 neurons on each hidden layer to construct both PIDL and DL for the estimation task. The optimization algorithm used in both neural networks is the limited-memory Broyden–Fletcher–Goldfarb–Shanno (LMBFGS) method. PIDL and DL are built by using TensorFlow, and TSE tasks are conducted on an Intel Core i7-8700 CPU @ 3.20GHz, with a RAM size of 16 GB.

Firstly, a scenario is investigated in which detected traffic input comes from sensors at fixed locations (Eulerian measurement frames). The accuracy of the estimation result is evaluated by using the accuracy metric defined in (Equation 6.2).
\[
ACC = \left(1 - \frac{\|P - \hat{P}\|_F}{\|P\|_F}\right) \times 100\% = \left(1 - \frac{\sqrt{\sum_{j=1}^{N_1-N_2} |\hat{\rho}(x^{(j)}, t^{(j)}) - \rho(x^{(j)}, t^{(j)})|^2}}{\sum_{j=1}^{N_1-N_2} |\rho(x^{(j)}, t^{(j)})|^2}\right) \times 100\% \tag{6.2}
\]

6.2.2 Eulerian Traffic Observation Scenario

Traffic data at 5 locations \((x = 2m, 250m, 500m, 750m, 1000m)\) are selected to train the neural network. These locations include the boundaries of the road segment (2m and 1000m) and sample locations in between (250m, 500m, and 750m). Adding difficulty to the estimation task, 1000 samples are randomly chosen from the 2500 samples at locations selected as training data. 1000 samples represent 0.4 percent of data from the entire road segment. The result is shown in Figure 6.3, in which training samples are also marked.

![PIDL Estimation](image1.png)

![DL Estimation](image2.png)

Figure 6.3: PIDL & DL Prediction Result, Eulerian Traffic Observation

With the aid of physics, PIDL efficiently utilizes the training data and produces estimation results bearing the resemblance of the entire experiment dataset shown in Figure 6.2. In contrast, the meager training samples are insufficient for the DL neural network to gain insightful knowledge
of traffic conditions in the entire road segment. It produces an inscrutable estimation result in this scenario. Table 6.1 details the accuracy and computation time of PIDL and DL for comparison (bold font indicates better performance).

Table 6.1: PIDL & DL Performance Comparison, Eulerian Traffic Observation

<table>
<thead>
<tr>
<th>Sensor Location</th>
<th>Computation Time (s)</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PIDL</td>
<td>DL</td>
</tr>
<tr>
<td>Fixed</td>
<td>5.7</td>
<td>31.3</td>
</tr>
</tbody>
</table>

*Varying Sample Size*

The impact of varying training sample sizes on the estimation result of PIDL and DL is further inspected. Using a sample size from 250 (0.1 percent of data) to 2500 (1 percent of data) as training data, the accuracy and computation time of PIDL and DL estimation are shown in Figure 6.4. The result on accuracy and computation time is given as the average of 5 repeated experiments with different (but of the same size) randomly selected training sets.

The detailed performance data of PIDL and DL is summarized in Table 6.2, with the bold font indicating better performance. Results show that PIDL exceeds the performance of DL at all sample levels by a large margin. It also shows that PIDL produces the estimation result in a shorter period of time.

This scenario involved using input data from sensors at fixed locations and demonstrated the powerful capability of PIDL in utilizing extremely limited data to produce accurate estimation results of traffic states. Results highlight the value of PIDL in TSE applications where sensors such as loop detectors are often installed sparsely at predetermined locations on the road. It demonstrates that the knowledge of traffic flow theory (LWR in this case) can effectively aid a deep learning neural network in the task of estimating traffic states using limited sensor data.
Figure 6.4: Accuracy and Convergence Time Comparison, Eulerian Traffic Observation

Table 6.2: Computation Time and Error Comparison: Sample Size, Eulerian Traffic Observation

<table>
<thead>
<tr>
<th>Sample Size</th>
<th>Convergence Time (s)</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PIDL</td>
<td>DL</td>
</tr>
<tr>
<td>250</td>
<td>3.6505</td>
<td>9.6049</td>
</tr>
<tr>
<td>500</td>
<td>2.9087</td>
<td>6.7817</td>
</tr>
<tr>
<td>750</td>
<td>8.9076</td>
<td>27.4786</td>
</tr>
<tr>
<td>1000</td>
<td>8.1593</td>
<td>5.8254</td>
</tr>
<tr>
<td>1250</td>
<td>8.6089</td>
<td>20.7073</td>
</tr>
<tr>
<td>1500</td>
<td>6.0234</td>
<td>2.1191</td>
</tr>
<tr>
<td>1750</td>
<td>9.7904</td>
<td>27.8704</td>
</tr>
<tr>
<td>2000</td>
<td>10.6343</td>
<td>61.4974</td>
</tr>
<tr>
<td>2250</td>
<td>2.3028</td>
<td>3.0996</td>
</tr>
<tr>
<td>2500</td>
<td>15.3145</td>
<td>3.8733</td>
</tr>
</tbody>
</table>
Now a scenario is explored in which data from randomized locations are considered to compare the performance between PIDL and DL. This scenario can be compared to the one where measurements are obtained in the Lagrangian frame of reference using probe vehicles or GPS data of connected vehicles.

6.2.3 Lagrangian Traffic Observation Scenario

1000 training samples (same as the previous scenario) are admitted at random locations and times, which allows us to mimic another traffic data collection method: through floating cars or probe vehicles. The prediction result is shown in Figure 6.5.

![Figure 6.5: PIDL & DL Prediction Result, Lagrangian Traffic Observation](image)

Both neural networks achieved acceptable estimation accuracy. Table 6.3 summarizes the performance of PIDL and DL in this scenario. Given the variance in time and location of training samples this time, it takes both neural networks a long time to give estimation results. DL accomplishes better estimation accuracy and PIDL has the advantage of curtailing computation time.

Note that Table 6.3 summarizes PIDL and DL performance under one sampling condition in which 1,000 training data were used. Before assuming DL achieved better accuracy, the following
Table 6.3: PIDL & DL Performance Comparison, Lagrangian Traffic Observation

<table>
<thead>
<tr>
<th>Sensor Location</th>
<th>Computation Time (s)</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PIDL</td>
<td>DL</td>
</tr>
<tr>
<td>Randomized</td>
<td>411.8</td>
<td>861.3</td>
</tr>
</tbody>
</table>

results using varying sample sizes should also be considered as they provide the fuller picture of performance comparison between PIDL and DL.

**Varying Sample Size**

The effect of varying training sample sizes in estimation performance is once again evaluated. The estimation accuracy and computation time of PIDL and DL are shown in Figure 6.6. It shows that both neural networks have the ability to acquire accurate estimation results and the difference is relatively insignificant. Again the result is the average of 5 experiments using diverse training sets randomly selected.

![Figure 6.6: Accuracy and Convergence Time Comparison, Lagrangian Traffic Observation](image_url)
The variety of observation locations where training data were obtained undoubtedly boosted the ability of DL in attaining more accurate estimation. Still, achieving similar estimation accuracy, PIDL converges at a faster speed. The shortened computation time gives PIDL a unique advantage in real-time applications during which timely estimation of traffic states is desired. The detailed performance data of PIDL and DL is summarized in Table 6.4.

Table 6.4: Computation Time and Error Comparison: Sample Size, Lagrangian Traffic Observation

<table>
<thead>
<tr>
<th>Sample Size</th>
<th>Convergence Time (s)</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PIDL</td>
<td>DL</td>
</tr>
<tr>
<td>250</td>
<td>119.7</td>
<td>262.9</td>
</tr>
<tr>
<td>500</td>
<td>150.4</td>
<td>353.8</td>
</tr>
<tr>
<td>750</td>
<td>130.8</td>
<td>640.1</td>
</tr>
<tr>
<td>1000</td>
<td>140.9</td>
<td>390.4</td>
</tr>
<tr>
<td>1250</td>
<td>95.6</td>
<td>570.1</td>
</tr>
<tr>
<td>1500</td>
<td>110.2</td>
<td>558.9</td>
</tr>
<tr>
<td>1750</td>
<td>138.4</td>
<td>480.7</td>
</tr>
<tr>
<td>2000</td>
<td>170.2</td>
<td>528.5</td>
</tr>
<tr>
<td>2250</td>
<td>175.6</td>
<td>409.6</td>
</tr>
<tr>
<td>2500</td>
<td>130.7</td>
<td>367.4</td>
</tr>
</tbody>
</table>

6.3 PIDL with CAV Sensing

This section provides details of the PIDL approach for traffic state estimation in the scenario with connected and autonomous vehicles (CAVs). Consider figure Figure 2.2, wherein a data exchange scenario between CAVs and edge devices like roadside units (RSUs), vehicle velocity $v(n, t)$ and location $x(n, t)$ information are broadcast by an CAV. If the CAV is within RSU communication range, RSU will receive the information from the CAV and pass it on to the fog. The fog layer will combine information from all RSU and will be able to reconstruct the velocity-field $v(x, t)$ (or
density-field $\rho(x,t)$). The reconstructed velocity field, $v(x,t)$, is a subset of the data broadcasted by CAVs, as data broadcasted in areas out of the communication range is not received by RSUs. The PIDL approach can be utilized to estimate and reconstruct the whole velocity field using a limited amount of data received from CAVs. The flow diagram of the PIDL approach in the CAV Setting is shown in Figure 6.7.

This case study is designed to resemble the application scenario in which the trajectory information of CAVs is captured by RSUs.

6.3.1 Data Description

The synthetic dataset is constructed to mimic vehicular traffic using the Lax-Hopf method as described in [134]. The test-bed consists of a 5000-meter road segment for 300 seconds ($x, t \in [0,5000] \times [0,300]$). The spatial resolution of the dataset is 5 meters and the temporal resolution is 1 second ($\Delta x = 5, \Delta t = 1$). In this case study, velocity-field $v(x,t)$ is the learning objective of the neural network. The experimental synthetic dataset is shown in Figure 6.8. The vehicle trajectory data is produced under the following assumptions and control measures:

1. **Vehicle speed-density relationship:** Greenshields’ fundamental diagram formulated is (Equation 3.29) applies in the relationship between vehicle speed and density. The free flow speed $v_f$ is 25 meters per second (90 kph or 56 mph). The maximum density $\rho_m$ is 0.15 vehicles per meter.

2. **Initial vehicle density:** At $t = 0$, density $\rho$ is 0.14 vehicle per meter (veh/m) at $x \in [0, 1500]$, 0.04 veh/m at $x \in [1500, 3500]$, 0.08 veh/m at $x \in [3500, 4000]$, and 0.1 veh/m at $x \in [4000, 5000]$.

3. **Upstream flow:** At $x = 0$, the upstream flow $q_{in}$ is 0.3 vehicle per second (veh/s) at $t \in [0, 60]$, 0.4 veh/s at $t \in [60, 180]$, and 0.1 veh/s at $t \in [180, 300]$.

4. **Downstream flow:** At $x = 5000$, the downstream flow $q_{out}$ is 0.2 vehicle per second (veh/s)
Figure 6.7: PIDL for TSE in CAV Setting
at $t \in [0, 60]$, 0.1 veh/s at $t \in [60, 120]$, and 0.3 veh/s at $t \in [120, 240]$ and 0 (to simulate a road closure or traffic accident) at $t \in [240, 300]$.

![Figure 6.8: Synthetic Vehicle Velocity $v(x, t)$ Data](image)

Recall that this case study is designed to utilize the CAV data as collected by RSUs, with the objective to estimate velocity field. It assumes that the RSUs are deployed every 1000 meters on the road segment; therefore, there are 6 RSUs installed on the 5000-meter road (the first one is installed at the initial location $x = 0$). The communication range of RSU is assumed to be 300 meters (DSRC technology) [135]. That means, vehicle information broadcast by CAVs at $x \in [0, 300]$ can be captured by the first RSU and the second RSU can log CAV data transmitted at $x \in [700, 1300]$. Data broadcasted at $x \in [300, 700]$ however is cutoff to the system. Data loss due to the limitation of the communication range of RSU is also considered. More details on the creation of the synthetic data can be found in [87].

The PIDL neural network for the experiment is built with a 10-hidden-layer architecture, with 40 neurons on each hidden layer. The optimization algorithm is LM-BFGS. The maximum learning iteration is 5000. For comparison, a DL neural network with the same architecture but unaware of
the flow conservation law is also trained in this case study.

6.3.2 Experiment with CAV Penetration Rate

Analyses with various penetration rates of CAV in the traffic stream are performed. The computation time and the relative percent $L_2$ error metric defined in (Equation 4.5) are used to evaluate the performance. Each neural network’s performance is evaluated 3 times using 3 training datasets obtained with different sampling seeds. This experiment also assumes that there is a 10 percent data loss; hence 90 percent of the transmitted data within the communication range of RSUs are recorded.

**1% Penetration Rate**

First, the PIDL’s performance with a vehicle fleet consisting of 1 percent CAVs and 99 percent conventional vehicles is investigated. Under this scenario, only 4 vehicles in the experiment can broadcast information through the vehicle to infrastructure (V2I) communication to RSUs. The estimation results of PIDL and DL (for comparison) are presented in Figure 6.9. Vehicle velocity data logged by RSUs are also marked.

**Interpretation:** Notice that with the aid of the physical law of traffic flow, PIDL outperforms DL with higher estimation accuracy and achieves shorter computation time, which means greater feasibility of time-sensitive applications. The physics-informed approach accommodates the reality of (training) data sparsity by leveraging the collocation points. As discussed before, collocation points are coordinate pairs of location and time $(x, t)$ where the physics-cost can be computed to quantify the disobedience of current states in terms of the governing conservation law. The physical equations provide tangible insights into the underlying relationship between traffic state variables such as flow and velocity. And the physics cost assessed by PIDL serves as a torchlight guiding the optimization process of parameters in the neural network.
Next, the performance of the PIDL approach considering a 10 percent CAV fleet is presented. The estimation result is shown in Figure 6.10. In both penetration level scenarios, the estimation error and computation time reported are the mean averages of 3 experiments.

**Interpretation:** With the higher availability of vehicle data in this scenario, the accuracy performance of DL is slightly better, suggesting the richness in input data compensates for the absence of physical law awareness. However, the accuracy achieved by DL comes at the expense of computation time. In contrast, PIDL achieves similar estimation accuracy in a much shorter time. The experiment once again demonstrates the benefit of incorporating physical law in training neural networks. It is observed that while accomplishing comparable estimation accuracy in some cases, the PIDL method always has the advantage of quick convergence and proves to be a desirable approach for real-time applications.
Table 6.5 provides the results for various penetration levels (1%, 10%, 25%, 50%, 75%, and 100%). **Bold** font indicates better performance.

**Interpretation:**

With the aid of physics, PIDL consistently surpasses the estimation accuracy metric of DL. It is also observed that with the increasing availability of training data as the CAV fleet expands, the time to convergence of PIDL and DL both decreases (when the CAV penetration rate is above 25%). This suggests that with the abundant training data, the estimation tasks are becoming increasingly undemanding, and both neural networks (PIDL & DL) are able to produce the results in shorter periods of time.

**Sensitivity Analysis on the Data Noise Level**

Lastly, sensitivity analysis is conducted with regard to the noise level in the training data to examine the performance of PIDL and DL. Even with the best available sensing technology, the
Table 6.5: Sensitivity Analysis for Various Penetration Rates

<table>
<thead>
<tr>
<th>CAV Penetration Rate</th>
<th>Computation Time</th>
<th>Relative L₂ Error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PIDL</td>
<td>DL</td>
</tr>
<tr>
<td></td>
<td>Computation Time</td>
<td>Relative L₂ Error (%)</td>
</tr>
<tr>
<td>1%</td>
<td>33.7s</td>
<td>75.0s</td>
</tr>
<tr>
<td>10%</td>
<td>238.1s</td>
<td>473.6s</td>
</tr>
<tr>
<td>25%</td>
<td>379.2s</td>
<td>643.8s</td>
</tr>
<tr>
<td>50%</td>
<td>148.6s</td>
<td>250.3s</td>
</tr>
<tr>
<td>75%</td>
<td>96.8s</td>
<td>124.7s</td>
</tr>
<tr>
<td>100%</td>
<td>21.8s</td>
<td>45.1s</td>
</tr>
</tbody>
</table>

The benefit of physics in training is more evident with added data noise, as the results show PIDL outperforms DL under noisy data scenarios, and the difference is more pronounced as the noise level grows.

**Summary of Results**

1. The computation time of PIDL is better for all penetration rates.
2. For 1% penetration rate, the relative $L_2$ error for PIDL was 31.7% as compared to 42.9% for DL.

3. With a 15% noise added to the sensor data, the relative $L_2$ error for PIDL was at 22.9% compared to 30.8% for DL.

6.4 PIDL for TSE with Realistic NGSIM Data

This experiment is designed to resemble scant observations from probe vehicles using the trajectory information from Next Generation Simulation (NGSIM) data [136]. It examines the performance of the physics-informed deep learning approach in recreating the velocity field. The vehicle trajectory data collected on the interstate 80 freeway (I-80) in Emeryville, California is chosen for this study.

6.4.1 Data Description

The road segment with recorded vehicle trajectory is about 1600 feet long [137], and the vehicle velocity and positional information are extracted from video recordings [138]. The 15-minute vehicle velocity-field obtained from 4:00pm - 4:15pm on April 13th, 2005 on I-80 freeway is shown in Figure 6.11. Shock waves can be observed in the figure, indicating congestion propagating backward on the freeway.

The velocity field is constructed from vehicle trajectory data using a binning method with spatial resolution $\Delta x$ of 20 feet and the temporal resolution $\Delta t$ of 5 seconds. The average speed of vehicles in each bin is calculated. The resulting velocity field consists of 180 temporal bins and 80 spatial bins.

6.4.2 Implementation Details

The performance of PIDL in regenerating the velocity field is investigated with the acquaintance of a few samples from the NGSIM dataset. The Greenshields’ fundamental diagram is deployed to
Figure 6.11: NGSIM Vehicle Velocity-Field, I-80 Freeway, 4:00pm - 4:15pm

represent the relationship between traffic state variables. The LWR conservation law is still used as the governing physical equation.

The recorded NGSIM data shows vehicles experiencing various levels of congestion [139]. In this experiment, traffic parameters are unknown and need to be estimated.

The density-velocity relationship is illustrated in Figure 6.12. The estimated values are summarized in Table 6.7. It is worth pointing out that the vehicle density $\rho$ is the summation of vehicles on the 5 lanes of the I-80 freeway in Emeryville, California.

Table 6.7: Estimated Parameters for Greenshields’ Fundamental Diagram

<table>
<thead>
<tr>
<th>Estimated Parameter Value</th>
<th>Free-flow Speed</th>
<th>Jam Density</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v_f = 46.64 \text{ ft/s}$</td>
<td>$\rho_m = 0.20 \text{ veh. / ft}$</td>
<td></td>
</tr>
</tbody>
</table>

Training samples from the velocity field are randomly selected (to mimic scant probe vehicle observations), providing the neural network observations at various spatial-temporal locations.
6.4.3 Results and Discussion

The reconstructed velocity-field using 1440 (10%) and 2000 (14%) samples are presented in Figure 6.13 and Figure 6.14. The accuracy of the estimation result is evaluated by using the relative percent $L_2$ error defined in (Equation 4.5).

Observe from Figure 6.13 and Figure 6.14 that in both scenarios, PIDL captured the characteristics of the NGSIM data and reconstructed the shockwaves from limited data samples. In contrast, no meaningful traffic insights can be extracted from the velocity field restored by DL. DL neural networks converged quicker at the expense of reconstruction accuracy. PIDL neural networks underwent noticeably more iterations updating the weight matrices and bias vectors before convergence and acquiring a lower cost value. It highlights the limited training iterations of DL as compared to PIDL. It may be due to the DL algorithm converging to a local minimum.

The case study also assumes a scenario in which the computation capacity is limited, and there is a constraint on the number of training iterations. The performance of PIDL and DL are examined after both neural networks are only allowed 5000 iterations for training. In this case, PIDL still
Figure 6.13: Reconstructed Velocity-Field based on 1440 (10%) Samples from NGSIM
Figure 6.14: Reconstructed Velocity-Field based on 2000 (14%) Samples from NGSIM
outperforms DL in terms of securing a lower value of cost. The cost and training iterations at convergence and the cost at the 5000-iteration limit of PIDL and DL are detailed in Table 6.8. Better performances are highlighted in bold.

Table 6.8: PIDL & DL Performance Comparison

<table>
<thead>
<tr>
<th>Training Samples</th>
<th>Relative L₂ Error (%) at Convergence</th>
<th>Training Iterations at Convergence</th>
<th>Relative L₂ Error (%) at 5000-Iteration</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PIDL</td>
<td>DL</td>
<td>PIDL</td>
</tr>
<tr>
<td>1440 (10%)</td>
<td>6.8</td>
<td>26.7</td>
<td>13014</td>
</tr>
<tr>
<td>2000 (14%)</td>
<td>9.4</td>
<td>17.6</td>
<td>14693</td>
</tr>
</tbody>
</table>

Although both neural networks are built upon the same architecture and are given the same training samples, the difference in cost functions provides the PIDL with a unique advantage in efficiently utilizing limited observation and producing estimation. The distinction of the PIDL cost function should be considered when comparing the performance between PIDL and DL based on the cost at convergence.

Summary of Results:

1. With 10% data, the relative L₂ error for PIDL was 6.8% compared to 26.7% for DL.

2. The experiment results demonstrate that PIDL achieved superior performance in TSE with a limited amount of training data, compared to DL.

6.5 Discussion on Case Study Results

Recall the cost function of PIDL in (Equation 4.8), the physics cost can be viewed as a regularization agent of the learning process [131]. Given the limited observation of traffic, a regular neural network has meager input in the training process to identify meaningful parameter values (weights
and biases, for instance) needed for traffic state estimation. In order to produce a realistic assessment, procedures need to be considered to prevent both under-fitting and over-fitting problems for accuracy and robustness. Under this context, laws from the traffic flow theory provide valid guidelines in fitting the model specification to acquire knowledge of the traffic in unobserved areas. It can prevent over/under-fitting training samples by penalizing learning where reconstruction does not satisfy the physical law.

Under-fitting commonly occurs when the input data is complex, and the structure of a learning model is naive. As pointed out earlier, given the complexity of deep learning models and the limited availability of the training data, under-fitting is not a major concern in our case. Preventing the over-fitting problem to ensure acceptable model performance is the prominent task here. Over-fitting happens when a model brings impaired generalization from observed data into the unseen field [140]. It excessively relies on the training input and loses the ability to adapt to the unseen test data. Regularization counters the issue of over-fitting by reducing the complexity of the model, limiting the weights assigned to features deemed less influential in producing the desired output [141]. It introduces a penalty term to the cost function to restrict the model from learning more features or assigning heavier weights to trivial features during training.

Common practices to avoid over-fitting include “early-stopping”, which deals with the phenomenon where testing accuracy ceases to improve after a certain amount of training [142]; “network-reduction” [143], which reduces the size of the learning network to limit model complexity, ultimately to quash the noise or irrelevant information in the training data. Data expansion [144] is another strategy that falls into this category to combat over-fitting by creating more training inputs through either acquisition of new data or re-sampling existing observation points.

However, all the above-mentioned methods have respective shortcomings. In order to implement “early-stopping”, the hyper-parameter of the training iteration limit (the point at which training must be stopped), needs to be determined through empirical evidence. Note that a premature stopping induces the “under-fitting” problem. Similarly, “network reduction” necessitates
an informed decision on reducing the model complexity while preserving the capacity of learning the underlying relationship presented in the training data. When new data is not easily available, which is often the case with TSE, bias associated with the re-sampling strategy may be introduced in the data expansion process.

By incorporating the physics cost $J_{PHY}$ into the cost function in the “physics-informed” framework, physical laws from the traffic flow theory act as a regularization agent in the training process and let the binding rules of traffic state with respect to space and time play a role in guiding the optimization. It penalizes the parameters and configuration of the neural network that does not comply with the system of laws imposed. This brings improvement in estimation accuracy and convergence time, as demonstrated in the case studies, owing to the two following unique advantages.

The first advantage of the PIDL model in preventing over-fitting is the information gained supplied by the physics. The partial derivatives of traffic state with respect to time and space in the conservation law reflect the relationship between traffic state in adjacent locations, and the incremental changes as the state evolves. This information on how traffic states transform presents a powerful tool for utilizing the sparse training input by constructing an educated estimation of traffic states at neighboring locations or succeeding timestamps.

The second benefit of the incorporation of physics cost $J_{PHY}$, by design, is the introduction of collocation points in the training process. Collocation points are coordinate pairs of location and timestamp $(x, t)$. Albeit the size of training input is small, a much greater amount of collocation points can be selected, and $J_{PHY}$ at collocation points are assessed to verify compliance with governing laws of physics. In other words, even if the ground truth of the traffic state for training is in short supply, the “physics-informed” neural network can calculate the physics cost at any given location $x$ and time $t$ and minimize it. The aforementioned advantages of PIDL are consequential in precise and prompt TSE. The next chapter discusses the potential limitations of PIDL.
CHAPTER 7
ANALYSING THE LIMITATIONS OF PIDL

In this chapter, the limitation of PIDL is investigated through a circular simulation testbed and the realistic NGSIM dataset. The case studies in this chapter paint the picture of the challenges in training a PIDL neural network with the hyperbolic LWR PDE and data with discontinuity present. In the end, this chapter offers a discussion on the lessons learned from the empirical results and exhibits four remedy strategies: (1) adding a diffusion term to switch the PDE to its parabolic version, (2) including observation and collocation points around the shocks, (3) incorporating interior training instances, and (4) altering the learning architecture of the PIDL neural network.

7.1 The Limitation of PIDL - Insights From Circular Testbed

This case study compares the PIDL reconstruction accuracy between learning the LWR conservation law (hyperbolic) and its parabolic form. The datasets used in this case study are synthetic vehicle density datasets generated on a ring road, which is represented by Figure 7.1. All neural networks are configured with the same learning architecture. 10000 collocation points are assigned in the density field to compute the physics-cost. The learning rate of the Adam optimizer is set to 0.001, and the number of training iterations is set to 8000.

7.1.1 Data Description

The datasets for this case study simulate vehicular traffic on a ring road. The location $x$ and time $t$ are normalized as $x, t \in [0, 1.0] \times [0, 3.0]$. The road is evenly divided into 240 spatial units with $\Delta x = 1/240$, and time is similarly separated into 960 units and each timestep represents the...
progression of $\Delta t = 1/320$.

The traffic state variable of interest in this case study is the vehicle density $\rho(x, t)$. The assumed physical model is LWR conservation law, paired with Greenshield’s fundamental diagram (FD) formulated in (Equation 3.29). The values of the free flow speed $v_f$ and the maximum density $\rho_m$ are normalized as well as both are set to 1. The dataset using the LWR conservation law (hyperbolic PDE) in (Equation 5.3) is first generated, shown in Figure 7.2a. Based on the same initial and boundary conditions, and only adding a diffusion term to the LWR PDE to make it parabolic, as explained in (Equation 5.6), the dataset using the parabolic form of LWR PDE is configured, illustrated in Figure 7.2b. The relative mean squared value of the difference between these two datasets in Figure 7.2 is 0.35%, indicating almost identical density values of the datasets.

Given the sampling choices, two subsets of training data inputs about the vehicle density $\rho(x, t)$ are designated: (1) initial and boundary condition data inputs and (2) interior data inputs. The interior data of the density field can be collected by either roadside detectors (Eulerian) or connected vehicles (Lagrangian). In the case study, the Lagrangian data from CVs are selected as the interior data.

1. Initial condition data are the vehicle density values $\rho(x, 0)$ as $t = 0, x \in [0, 1.0]$. Boundary
condition data include vehicle density values at the first location $\rho(0, t), t \in [0, 3.0]$, and at the last location $\rho(1.0, t), t \in [0, 3.0]$.

2. Interior data (CV data in the case study) $\rho(x, t)$ comes from the CV fleet in the traffic. They can be gathered at any randomly selected location along the vehicle trajectory and reflect the density value $\rho(x, t)$, given $x \in [0, 1.0]$ and $t \in [0, 3.0]$.

The initial condition data on vehicle density can be registered through a still image, recorded by devices such as roadside video cameras or drones. The boundary condition data can be obtained from a stationary detector deployed along a freeway at the start location $x = 0$, and the end location $x = 1.0$ (normalized).

### 7.1.2 Reconstruction With Initial and Boundary Inputs

Firstly, PIDL results based only on training inputs about the initial and boundary conditions are evaluated. Four levels of available training inputs are selected (10%, 20%, 50%, and 90% of...
the total numbers of initial and boundary data), and use both L-BFGS-B and Adam optimizers to reconstruct the hyperbolic LWR PDE and its parabolic variation with the diffusion term. The results are shown in Table 7.1 (best results are shown in **bold**). The reconstructed density fields, trained with 10% initial and boundary inputs, are shown in Figure 7.3.

Table 7.1: Density Reconstruction Results (with Initial and Boundary Inputs), Relative $L_2$ Error

<table>
<thead>
<tr>
<th>Initial and Boundary Inputs $N_{oi}$</th>
<th>Relative $L_2$ Error</th>
<th>L-BFGS-B (parabolic)</th>
<th>L-BFGS-B (hyperbolic)</th>
<th>Adam (parabolic)</th>
<th>Adam (hyperbolic)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10%</td>
<td><strong>1.64e-02</strong></td>
<td>3.09e-01</td>
<td>6.30e-02</td>
<td>3.03e-01</td>
<td></td>
</tr>
<tr>
<td>20%</td>
<td><strong>3.68e-03</strong></td>
<td>3.03e-01</td>
<td>4.73e-02</td>
<td>2.98e-01</td>
<td></td>
</tr>
<tr>
<td>50%</td>
<td><strong>5.47e-03</strong></td>
<td>2.52e-01</td>
<td>8.07e-02</td>
<td>2.86e-01</td>
<td></td>
</tr>
<tr>
<td>90%</td>
<td><strong>4.77e-03</strong></td>
<td>2.85e-01</td>
<td>3.61e-02</td>
<td>2.83e-01</td>
<td></td>
</tr>
</tbody>
</table>

Among all training settings, it is observed that PIDL models achieved much higher accuracy (lower relative $L_2$ error) in reconstructing the parabolic PDE with the diffusion term compared to the ones with the hyperbolic LWR PDE. Trained with the L-BFGS-B optimizer and 10% boundary and initial observations, the relative $L_2$ error of PIDL reconstruction is 0.0164, which is merely 5.3% of the 0.309 error of the same metric in reconstructing the hyperbolic LWR PDE.

Several snapshots of the reconstruction are also taken, at time $t = 0, 0.5, 1.0, 1.5, 2.0, 2.5$ for comparison. From Figure 7.3b, notice that PIDL learning the hyperbolic PDE encountered difficulties in reconstructing the density state at locations where the discontinuity of density data occurs. However, in Figure 7.3a, equipped with the diffusion term in the parabolic PDE, the reconstruction result is smoothed and closely aligned with the ground truth.

### 7.1.3 Reconstruction With Initial, Boundary, and Interior Inputs

Subsequently, this subsection evaluates the reconstruction accuracy under the scenarios in which varying levels of data on the initial and boundary conditions and the interior conditions (CV in-
(a) Learning \textbf{parabolic conservation law} using L-BFGS-B optimizer, relative L$_2$ error: 0.0164

(b) Learning \textbf{hyperbolic conservation law} using L-BFGS-B optimizer, relative L$_2$ error: 0.309

(c) Learning \textbf{parabolic conservation law} using Adam optimizer, relative L$_2$ error: 0.0630

(d) Learning \textbf{hyperbolic conservation law} using Adam optimizer, relative L$_2$ error: 0.303

Figure 7.3: Density Reconstruction, Trained with 10\% Initial and Boundary Inputs
puts) are available. Two levels of available inputs $N_{o1}$ on the initial and boundary conditions are picked: (1) $N_{o1} = 108$ inputs, representing 5% of initial and boundary data; and (2) $N_{o1} = 432$ inputs, representing 20% of the available data. For the number of CV inputs $N_{o2}$, two settings are also selected: $N_{o2} = 1146$ and $N_{o2} = 4584$, accounting for 0.5% and 2% of the interior of the density field, respectively. The reconstruction result, measured by the relative $L_2$ error defined in (Equation 4.5) is shown in Table 7.2 (again, best results are tabulated in **bold**). The reconstruction results with 20% initial and boundary inputs ($N_{o1} = 432$), and 2% CV inputs ($N_{o2} = 4584$) are shown in Figure 7.4.

Table 7.2: Density Reconstruction Results (with Initial and Boundary Inputs & CV Inputs), Relative $L_2$ Error

<table>
<thead>
<tr>
<th>Initial and Boundary Inputs $N_{o1}$</th>
<th>CV Inputs $N_{o2}$</th>
<th>Relative $L_2$ Error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>L-BFGS-B (parabolic)</td>
<td>L-BFGS-B (hyperbolic)</td>
</tr>
<tr>
<td>5% 0.5%</td>
<td>8.33e-03</td>
<td>2.45e-01</td>
</tr>
<tr>
<td>5% 2%</td>
<td>6.05e-03</td>
<td>2.37e-01</td>
</tr>
<tr>
<td>20% 0.5%</td>
<td>4.50e-03</td>
<td>2.23e-01</td>
</tr>
<tr>
<td>20% 2%</td>
<td>3.88e-03</td>
<td>2.25e-01</td>
</tr>
</tbody>
</table>

Comparing to the results in Table 7.1, the inclusion of CV inputs in the training inputs of PIDL slightly improves the reconstruction accuracy with the hyperbolic PDE. However, the learning performances of PIDL architecture with the parabolic PDE are still far superior. With 20% initial and boundary observations, and 2% CV inputs, the PIDL model with L-BFGS-B optimizer achieves a relative $L_2$ error of 0.00388 for reconstructing the parabolic PDE, which is 1.72% of the relative $L_2$ error with the hyperbolic PDE, at 0.225.
Figure 7.4: Density Reconstruction, 20% Initial and Boundary Inputs & 2% CV Inputs
7.2 Limitation of PIDL - Insights From Field Data

This section will further shed light on the topic by examining the limitation with PIDL and compare it with Lax-Friedrichs’ numerical scheme [145] in learning the traffic density using the “Next Generation SIMulation” (NGSIM) dataset [136].

7.2.1 Data Description

The NGSIM dataset records traffic conditions using video cameras and processes the traffic state variables such as velocity and vehicle density through vehicle trajectories identified in the video recordings [146]. The vehicle density data used, illustrated in Figure 7.5, contains vehicle density for 45 minutes on a 2060-foot segment of US-101 freeway. Shockwaves of vehicles stopping due to traffic congestion, which back-propagates in space and forward-propagates in time, can be observed in the plot of vehicle density. This freeway segment has five lanes, one on-ramp, and one off-ramp. An additional lane is attached to the freeway between the locations of the on-camp and the off-ramp. The data used was collected from 7:50 a.m. to 8:35 a.m. in Los Angeles, California, on June 15th, 2005. During the first 12 minutes of the data, a free-flow zone is observed in the post-off-ramp area, while the areas before the off-ramp are experiencing stop-and-go wave traffic [147].

![Figure 7.5: Vehicle Density on US-101 Highway Segment, between 7:50 am and 8:35 am, NGSIM](image-url)
7.2.2 Field Data Reconstruction Using PIDL With Hyperbolic PDE

The density dataset is tabulated with spatiotemporal bins of $\Delta x = 20 ft$ and $\Delta t = 5 s$. At $t = 0$, the initial condition contains 104 data points along the 2060-ft road segment ($x = 0, 20, ..., 2060$). The lower bound condition at $x = 0$ and the upper bound condition at $x = 2060$ each has 540 boundary condition points ($t = 0, 5, ..., 2695$). Together, the initial and boundary condition data of vehicle density $\rho(x,t)$ are all used as training inputs of PIDL with the L-BFGS-B optimizer.

The governing physical equation of the PIDL architecture is the hyperbolic LWR conservation law paired with Greenshield’s fundamental diagram. The estimated value of maximum density $\rho_m$ is 0.12 vehicle per foot (sum of all traffic lanes), and the free-flow speed $v_f$ is estimated at 80 feet per second (54.54 miles per hour). The reconstruction result with PIDL is shown in Figure 7.6.

![Reconstruction of $\rho(x,t)$ (veh./ft.) in NGSIM](image)

Figure 7.6: Reconstruction with PIDL, Relative $L_2$ Error: 0.345

The relative $L_2$ error of PIDL reconstruction is 0.345. From the snapshots of $t =$
0, 450, 900, 1350, 1800, 2250, PIDL tries to mimic the evolution of the density field between the lower boundary \( x = 0 \) and the upper boundary \( x = 2060 \); however, it cannot overcome the challenge in learning the stochastic perturbation of the density state. It is also evident in the reconstruction plot of \( \rho(x, t) \) that the PIDL architecture overly generalizes the output and fails to capture any traffic patterns, such as the shockwaves present in the dataset.

7.2.3 Field Data Reconstruction Using PIDL With Parabolic PDE

For NGSIM US-101 data, the diffusion coefficient \( \varepsilon \) in (Equation 5.6) is estimated to be 0.13 for regular motor vehicles [148]. A range of values at 0.05, 0.1, 0.13, 0.15, 0.20 is selected for \( \varepsilon \) and reconstructs the density dataset using the LWR PDE with the addition of the diffusion term as the physics of PIDL. The reconstruction results with \( \varepsilon = 0.13 \) is shown in Figure 7.7.

![Reconstruction of \( \rho(x, t) \) (veh./ft.) in NGSIM](image)

**Figure 7.7: Reconstruction with PIDL, \( \varepsilon = 0.13 \), Relative L2 Error: 0.319**
It is observed that with the realistic NGSIM dataset, the addition of the diffusion term only slightly improves the accuracy, landing a relative L\textsubscript{2} error at 0.319. Similar to the troubles the PIDL neural network with hyperbolic LWR PDE encountered, the stochastic nature of traffic disturbance overcomes the PIDL’s ability to accurately capture the traffic state, with only boundary and initial observations. Our previous work [88] demonstrates that the inclusion of Lagrangian observations, such as CV data, will significantly improve PIDL performances in this case.

The sensitivity analysis is also conducted on the diffusion coefficient $\varepsilon$, and the results are presented in Table 7.3. With varying values of the diffusion coefficient $\varepsilon$, the conclusion from this analysis is unwavering: although the diffusion term smooths out the reconstruction in areas where state discontinuity is present, it cannot accurately estimate the traffic state with only initial and boundary observations.

Table 7.3: Relative L\textsubscript{2} Error with Choices of Diffusion Coefficient $\varepsilon$

<table>
<thead>
<tr>
<th>Value of $\varepsilon$</th>
<th>Relative L\textsubscript{2} Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.05</td>
<td>3.44e-01</td>
</tr>
<tr>
<td>0.10</td>
<td>3.23e-01</td>
</tr>
<tr>
<td>0.13</td>
<td>3.19e-01</td>
</tr>
<tr>
<td><strong>0.15</strong></td>
<td><strong>3.10e-01</strong></td>
</tr>
<tr>
<td>0.20</td>
<td>3.24e-01</td>
</tr>
</tbody>
</table>

7.2.4 Reconstruction Using Lax-friedrichs’ Numerical Scheme

The vehicle density dataset from NGSIM can also be reconstructed by using the Lax-Friedrichs’ differencing method [104] with the complete initial and boundary conditions. The reconstruction is pictured in Figure 7.8.

Along with a smaller relative L\textsubscript{2} error, the most significant improvement in the reconstruction result by the Lax-Friedrichs’ method is the capability to capture and rebuilds the pattern of
shockwaves in the dataset, based only on the inputs from the initial and boundary condition. The advantage of the numerical scheme resides in the ability to propagate the discontinuity in the density field based on the method of characteristics, while the PIDL architecture struggles with the reconstruction task.

7.3 Discussion on Empirical Results

Recent examinations have elucidated the challenges associated with training and drawbacks of certain data representations in PIDL. In several instances, unstable convergence occurs in the gradient-descent-based PIDL training, especially when the underlying PDE solution has high-frequency features [79]. This pathological behavior observed in PIDL training is due to the multi-scale interactions between the cost terms in optimizing the neural network cost [149]. It leads to stiffness in the gradient flow dynamics, ultimately inducing a severe constraint on the learning rate and adding detriment to the stability of the training process. PIDL, which often deploys fully-connected hidden layers, faces the challenge termed “spectral bias” that cannot reasonably assimilate a nonlinear hyperbolic PDE when its solution involves shocks [150].

**Potential mitigation directions:** The potential approaches to improve the PIDL paradigm include (a) switching the hyperbolic physics with the parabolic counterpart by adding the diffusion term, (b) incorporating more observation or collocation points around the shocks, (c) including more interior training instances (e.g., Lagrangian measurements), (d) modifying the fully-connected learning architecture of the neural network. Next, these approaches are discussed below.
For a one-dimensional hyperbolic PDE with a non-convex flux function, its analytical solution can be depicted by a simple piecewise continuous function, and the stability of its solution can be significantly improved by adding a diffusion term to the inherent PDE. With smoothing around the shock by the diffusion, the neural network can recuperate the actual scale and location of the shock, solves the PDE in its parabolic form, and leads to precise approximation results [151]. However, common practices assume the coefficient $\epsilon$ to be zero when fitting a realistic traffic dataset with LWR conservation law [152, 68], leaving out the diffusion effect. From Figure 7.3a and Figure 7.3b in section 7.1, the inclusion of the diffusion term significantly improved the reconstruction accuracy of a PIDL neural network. The deficiency of PIDL with hyperbolic LWR PDE is not rooted in the learning architecture of the neural network, or the hyperparameter selection [151]. The diffusion term enhances the stability of the gradient optimization process in training the neural network around the areas with state discontinuity and shockwaves.

Increasing observations or collocation points along the shock trajectories in the training of the neural network forms another approach [153]. However, one challenge in this approach would be identifying the shock location. As has been observed, PIDL struggles with approximating the vehicle density where localized non-linear discontinuity exists in the data; adding artificial dissipation could improve the learning result of the hyperbolic conservation law [154, 155].

Based on the results in section 7.2, PIDL with the parabolic variant of LWR PDE cannot overcome the random perturbation in the traffic dataset and accurately estimate the traffic density based on pure observation of the initial and boundary conditions. Our previous work suggests the benefit of including Lagrangian observation in this setting for the task of TSE [88].

Alternatively, recent studies have started tweaking the design of the deep learning architecture in PIDL to circumvent the issue of learning underlying hyperbolic PDEs with discontinuity and no strong solution. The adoption of attention-based recurrent neural networks is introduced to capture the localized shock waves in the nonsmooth solution to governing equations [153]. It mitigated the challenges by substituting the conventional fully connected feedforward architecture.
in PIDL with recurrent neural networks and attention mechanisms. Additionally, convolutional neural network architectures have also demonstrated efficiency in assimilating data with nonlinear shock features or high-frequency components in the PDE solution [156] [157]. It is pointed out that the optimization of a PIDL neural network in learning a hyperbolic PDE can be a futile process due to the fact that the pointwise residual blows up during approximating the exact solution, and an alternative optimization approach can be found by using the residual relate to the Kružkov entropy condition to replace the pointwise residual [158].

In practice, the task of TSE involves realistic traffic data in which a diffusion term is inherent - drivers will gradually slow down the speed of their vehicles in participation of congestion or when a slowdown is visually perceivable. Therefore when adopting physics-informed deep learning for TSE, this nature of smoothness around shockwaves should be considered as part of the “physics”, which illustrates the underlying relationship between traffic states. Recall the diffusion term in the parabolic form of LWR PDE is weighted by the parameter $\epsilon$. The value of $\epsilon$ is associated with drivers’ behavior (reaction time to press the brake, for example) and should be tuned according to the traffic dataset before plugging in the conservation equation for the realistic reconstruction of the traffic data. Other approaches that transportation planners and agency practitioners can adopt is to include Lagrangian sensing data [87, 88], increasing the number of observation samples in the shockwave areas [153], and domain decomposition and projection onto the space of high-order polynomials [73].
CHAPTER 8
IMPROVING PIDL FRAMEWORK WITH NONLOCAL PHYSICS

Given the limitations of PIDL for TSE exhibited in Chapter 7, this chapter presents another approach to strengthen the PIDL paradigm for TSE with nonlocal physics. Recent research has shown advancements in extending the traffic models and conservation laws to the nonlocal versions [159, 160, 161], as from the driver’s perspective, the driving behavior involves reaction to the preceding traffic, which corresponds to the spatial integration of local traffic state in front of the driver.

Recall in chapter 3 the formulation of the LWR conservation law in the Eulerian coordinate system with the spatiotemporal location \( \mathbf{X} = (x, t) \) denote the independent variables, in (Equation 8.1).

\[
\text{For } (x, t) \in \mathbb{R} \times \mathbb{R}^+ : \quad \frac{\partial q(x, t)}{\partial x} + \frac{\partial \rho(x, t)}{\partial t} = 0 \quad (8.1)
\]

And the Greenshields’ fundamental diagram (FD) which depicts the relationship between velocity and density is given in (Equation 8.2):

\[
\begin{align*}
q(x, t) &= \rho(x, t) v_f \left( 1 - \frac{\rho(x, t)}{\rho_m} \right) \\
v(x, t) &= v_f \left( 1 - \frac{\rho(x, t)}{\rho_m} \right)
\end{align*} \quad (8.2)
\]

Pairing the LWR partial differential equation and the Greenshields’ fundamental diagram results in two ways to express the flow conservation law, using the density variable \( \rho(x, t) \) as in (Equation 8.3).

\[
v_f (1 - \frac{2 \rho(x, t)}{\rho_m}) \frac{\partial \rho(x, t)}{\partial x} + \frac{\partial \rho(x, t)}{\partial t} = 0 \quad (8.3)
\]
This chapter develops a PIDL neural network with the nonlocal version of traffic states as the prior knowledge of physics and demonstrate the improvement in state reconstruction results. For readers’ convenience, the physics of nonlocal traffic flow models in chapter 3 is presented here again for the formulation of PIDL with nonlocal traffic states.

8.1 Nonlocal Traffic States

Through the fundamental diagram (FD), the local velocity $v(x, t)$ at location $x$ and time $t$ can be determined by the local density $\rho(x, t)$. Therefore $v(x, t)$ can also be expressed as $v(\rho(x, t))$. With the nonlocal version of traffic states, the velocity at location $x$ and time $t$ is regulated by the nonlocal density $\rho_n$ (hence the notation “n”), which is obtained by a convolution of density conditions within the perception window. The scenario is illustrated in Figure 8.1 and mathematical formulation is presented in (Equation 8.4).

$$
\begin{align*}
\rho_n(x, t) &= \rho(x, t) * \theta(x) = \int_{x}^{x+w} \rho(\tau, t) \theta(\tau - x) d\tau \\
v(x, t) &= v(\rho_n(x, t))
\end{align*}
$$

The Greenshields’ FD in (Equation 8.2) is subsequently modified to reflect the relationship between the local velocity and the nonlocal density, given in (Equation 8.5).
\[
q(x, t) = \rho(x, t) v_f \left(1 - \frac{\rho_n(x, t)}{\rho_m}\right) \\
v(x, t) = v_f \left(1 - \frac{\rho_n(x, t)}{\rho_m}\right)
\]

(Equation 8.5)

Observe that in (Equation 8.5), the local velocity \(v(x, t)\) has the linear relationship with the convoluted nonlocal density \(\rho_n(x, t)\), instead of the local density \(\rho(x, t)\).

The LWR conservation law with nonlocal traffic states is expressed in (Equation 8.6) with respect to density \(\rho(x, t)\).

\[
\frac{\partial \rho(x, t)}{\partial t} + v_f \left(1 - \frac{\rho_n(x, t)}{\rho_m}\right) \frac{\partial \rho(x, t)}{\partial x} - \rho(x, t) \frac{\partial \rho_n(x, t)}{\partial x} \frac{v_f}{\rho_m} \frac{\partial \rho_n(x, t)}{\partial x} = 0
\]

(Equation 8.6)

Note that in (Equation 8.6), the nonlocal density \(\rho_n(x, t)\) and the local density \(\rho(x, t)\) are treated as two separate variables. The partial derivative terms with respect to \(x\) are calculated separately, compared to (Equation 5.3). Next, the kernels used for calculating the convolution of density are explained.

### 8.2 Kernel Functions

In (Equation 8.4), \(\theta(x)\) represents the kernel function in the perception window (hence the notation “w”) and has its property listed in (Equation 8.7). The perception window is also referred to as look-ahead window.

\[
\int_{x_0}^{x_0+w} \theta(x) dx = 1 
\]

(Equation 8.7)

Two kernel functions — the constant kernel and the linearly decreasing kernel are formulated below, in (Equation 8.8) and (Equation 8.9) respectively, and are illustrated in Figure 8.2 and Figure 8.3.
8.2.1 The Constant Kernel

When applying the constant kernel, the nonlocal traffic density is the average of the density values in the perception window.

\[
\theta(x) = \frac{1}{w}, \quad x \in [x_0, x_0 + w]
\]  

(8.8)

![Figure 8.2: The Constant Kernel](image)

8.2.2 The Linearly Decreasing Kernel

When applying the linearly decreasing kernel, the nonlocal traffic density is a weighted average of the density values in the perception window, with the density value at locations closer to the vehicle being assigned a greater weight.

\[
\theta(x) = \frac{2}{w} \left(1 - \frac{x - x_0}{w}\right), \quad x \in [x_0, x_0 + w]
\]  

(8.9)

8.3 PIDL with Nonlocal Physics

The overall PIDL approach for TSE with nonlocal traffic state is illustrated in Figure 8.4. During the process of optimizing the neural network, the convolution kernel will be applied to the tentative state estimation output to construct the field of nonlocal traffic state, which is then used to
compute the physics-cost term $J_{PHY}$ and consequently produces the total cost $J$ of the interim state estimation result. Once the cost of the optimized output meets the convergence requirement, PIDL returns this state estimation result as the final output. The algorithm of PIDL with nonlocal physics (using density $\rho(x, t)$ as an example) is stated in Algorithm 1.

Algorithm 1 PIDL Optimization with Nonlocal Physics

Require: Density observations $P_O = \{\rho(x^j_c, t^j_o)|j = 1, 2, \cdots, N_o\}$

Require: Collocation points $C = \{(x^k_c, t^k_c)|k = 1, 2, \cdots, N_c\}$

Initialize neural network $\Psi(\theta)$

while Convergence not reached do

Generate density estimation $\hat{P}$

Compute $J_{DL} = MSE(\rho(x, t), \hat{\rho}(x, t))$ = \(\frac{1}{N_o} \sum_{j=1}^{N_o} |\rho(x^j_c, t^j_o) - \hat{\rho}(x^j_c, t^j_o)|^2\)

Compute $J_{PHY} = MSE(0, \frac{\partial \rho(x, t)}{\partial t} + v_f (1 - \frac{\rho m}{\rho n}) \frac{\partial \rho(x, t)}{\partial x} - \rho(x, t) \frac{v_f}{\rho m} \frac{\partial \rho n(x, t)}{\partial x} - \rho(x, t) \frac{v_f}{\rho m} \frac{\partial \rho n(x, t)}{\partial x})$

Compute $J = \mu * J_{DL} + (1 - \mu) * J_{PHY}$

if Convergence reached then

Output density estimation $\hat{P}$

else

Retrain $\Psi(\theta)$

end if

end while

8.4 Field Data Validation

The vehicle density data of I-80 freeway from the NGSIM dataset is used as field data for validation of PIDL with nonlocal physics. The vehicular flow was recorded on April 13th, 2005 from 4:00
Figure 8.4: Approach Overview of PIDL with Nonlocal Physics for TSE
pm to 4:15 pm in Emeryville, California. The density data is plotted in Figure 8.5.

![Vehicle Density Field](image)

**Figure 8.5: NGSIM Vehicle Density-Field, I-80 Freeway, 4 : 00pm - 4 : 15pm**

The PIDL reconstruction results of the density field, with nonlocal physics — using the constant kernel and the linearly decreasing kernel, paired with a 60-ft perception window — are summarized in Table 8.1.

**Table 8.1: Performance Analysis on Nonlocal PIDL: Inaccuracy in Reconstruction**

<table>
<thead>
<tr>
<th>Model Type</th>
<th>Convolution Kernel</th>
<th>Perception Window</th>
<th>RSE (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Local LWR</td>
<td>N/A</td>
<td>20.70</td>
</tr>
<tr>
<td>2</td>
<td>Nonlocal LWR</td>
<td>Constant</td>
<td>60-ft</td>
</tr>
<tr>
<td>3</td>
<td>Nonlocal LWR</td>
<td>Linearly decreasing</td>
<td>60-ft</td>
</tr>
</tbody>
</table>

The additional regularization effect is observed with the use of nonlocal physics and incremental improvement in terms of reconstruction accuracy using the linearly decreasing kernel with a 60-ft perception window. With nonlocal physics, the training time of the neural network is also drastically reduced while the shockwaves are preserved in the reconstruction results.
Figure 8.6: Reconstruction with Local LWR

Figure 8.7: Reconstruction with a Linearly Decreasing Kernel and a 60-ft Perception Window
Figure 8.8: Reconstruction with a Constant Kernel and a 60-ft Perception Window
CHAPTER 9
CONCLUSION

This chapter reflects back on the proposed physics-informed deep learning paradigm for traffic state estimation and the results from the case studies. It also discusses the applicability of physics as a regularization agent in the framework of PIDL; and the limitation of PIDL with hyperbolic partial differential equations and with only boundary and initial observations. It concludes the dissertation with suggestions for future work.

9.1 Summary of Work and Conclusions

Given the high cost of congestion to the economy and the waste of time due to traffic delays for commuters each day, there is a dire need for accurate perceptions of traffic conditions on road infrastructures to manage traffic flow efficiently. To achieve this goal, this dissertation develops a physics-informed deep learning (PIDL) paradigm for traffic state estimation (TSE), combining the strength of the underlying physical laws of traffic flow and deep learning techniques. Results indicated that the presented PIDL paradigm has the capability to work with a meager amount of training data, and outperform the benchmark deep learning method in terms of estimation accuracy and computation time.

Chapter 5 in this dissertation formulates PIDL for TSE with the Lighthill-Whitham-Richards (LWR) flow conservation law (paired with both the Greenshields’ and the inverse-lambda fundamental diagrams) and cell transmission model (CTM). The case studies in chapter 6 employ both synthetic and field datasets, and contemplate many practical considerations: spatial separation of roadside units (RSUs), communication range, packet loss, and penetration levels of connected vehicles. The results from the case studies demonstrate the potent capability of PIDL in utilizing scantily available traffic data for accurate and real-time TSE. For example, the accuracy of PIDL
reconstruction of the density field in subsection 6.2.2 is 73.7% while a DL neural network with the same architectural but not the knowledge of LWR PDE only achieves an accuracy of 37.0% (results are in Table 6.1). The results in Table 6.3 reveal that with Lagrangian traffic observations, PIDL has the advantage of faster convergence, producing the output in 411.8 seconds while DL takes 861.3 seconds. The sensitivity analyses regarding the penetration rate of CAV and the level of data noise in section 6.3 and the case study with the realistic next generation simulation (NGSIM) dataset in section 6.4 further affirm the supremacy of PIDL in terms of estimation accuracy and applicability for real-time applications, over the benchmark deep learning approach.

This work also exhibits the difficulties of training a PIDL neural network to reconstruct a certain type of partial differential equation (PDE) - the hyperbolic PDE for which a strong solution cannot be obtained. The non-smooth weak solution to conservation law-based traffic flow models (such as the LWR PDE) causes PIDL failure in capturing the scale and location of the discontinuity. The case studies in chapter 7 showcase the stark differences between the learning result using PIDL with the first order hyperbolic LWR PDE and its parabolic counterpart, in which the additional diffusion term secures a strong solution and leads to pinpoint approximation. It is observed that the neural network fails to approximate the nonlinear relationship of a hyperbolic PDE in areas where shockwaves are present, whereas the diffusion term in the parabolic PDE ensures improved data estimation in these areas and thus ameliorates the reconstruction result. Four remedy strategies are discussed to improve the PIDL estimation result, including switching the PDE to the parabolic version, adding collocations points around the discontinuity, incorporating interior data instances for training, and revising the learning structure of the neural network. The framework of incorporating nonlocal physics into the PIDL framework for TSE is introduced in chapter 8, capturing the practical nonlocal relationship between traffic state variables due to drivers’ reaction time and perception of downstream traffic conditions.

In summary, this dissertation presents the development of a physics-informed deep learning paradigm for traffic state estimation, with various realistic considerations of the application per-
spective such as the form of traffic observation (Eulerian, Lagrangian, or CAV measurements), data availability, and noise levels. The case studies using both synthetic and real-world traffic datasets validate the effectiveness in terms of estimation accuracy of the proposed approach. This research work also investigates the limitation of PIDL with the Lighthill-Whitham-Richards conservation law. It comments on the mitigation strategies that transportation researchers can adopt to strengthen PIDL for TSE. In the end, the consideration of nonlocal physics in the PIDL paradigm is developed as another approach to improve the PIDL paradigm, incorporating the nonlocal relationship between traffic states.

9.2 Shortcomings and Future Work

Apart from the shortcomings and challenges discussed in previous chapters, this section highlights the requirement of repeatedly training the current PIDL architecture for traffic state prediction as the shortcoming of the proposed paradigm. Moreover, it suggests the inverse problem of flow model identification as one of the future works of this research.

9.2.1 Retraining Requirement for Traffic Prediction

Traffic state estimation is an integral task of transportation research; at the same time, developing tools for traffic prediction can also be of great interest to traffic operation practitioners and policymakers in recognizing the evolution of traffic conditions. For instance, predicting the change in traffic volume and future traffic flow in the event of a major interruption or a natural disaster will be essential in making corresponding arrangements in advance. To accomplish this task of traffic prediction, the current PIDL paradigm requires retraining to incorporate streaming data instances of real-time traffic conditions. Each time traffic observations obtained by sensors or connected vehicles are updated, the parameters of the neural network need to be recalibrated to reflect the ever-changing conditions of the current traffic state, hence impeding its applicability for real-time applications. To alleviate this issue, the learning architecture of PIDL needs to be amended to pre-
serve the majority of trained parameters after model convergence. New training instances of traffic will only update the last few layers of the neural network to influence the output as the prediction. The currently high computation cost of retraining for prediction tasks is a shortcoming of the proposed PIDL paradigm and points to a future research direction of modifying the PIDL neural network structure to accommodate incoming traffic observation with limited parameter updates to the neural network for swift applications in traffic state prediction.

9.2.2 The Inverse Problem of Model Identification

Another future research direction is adopting the PIDL framework to identify the underlying relationships between the traffic state variables, given observations of the traffic condition. For example, with measurements of traffic harvested from video cameras on a freeway, can PIDL identify the plausible traffic flow model, be it LWR, CTM, or a model depicted by a higher order PDE, portraying the traffic flow relationship? If so, can it further estimate the values of the critical parameters, such as the maximum speed and the jam density? This dissertation focuses on the forward problem which infers the traffic state conditions with the given physics from the traffic flow models. The inverse problem of model identification by a PIDL neural network opens up a new avenue of research that is worth examining to expand PIDL applications in transportation.
This journal is 100% open access, which means that all content is freely available without charge to users or their institutions. All articles are currently published under Creative Commons licenses (either CCBY or CCBY-NC-ND), and the author retains copyright. Users are allowed to read, download, copy, distribute, print, search, or link to the full texts of the articles published under CCBY, or use them for any other lawful purpose, as long as proper attribution is given. Articles published under CCBY-NC-ND are also available to users under the same conditions as CCBY, but the reuse cannot be for commercial purposes or change the work in any way.

The articles in this journal are peer reviewed in accordance with the requirements set forth in the IEEE PSPB Operations Manual (sections 8.2.1.C & 8.2.2.A). Each published article was reviewed by a minimum of two independent reviewers using a single-blind peer review process, where the identities of the reviewers are not known to the authors, but the reviewers know the identities of the authors. Articles will be screened for plagiarism before acceptance.

Corresponding authors from low-income countries are eligible for waived or reduced open access APCs.
REFERENCES


