IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 26 September 2025, accepted 2 October 2025, date of publication 6 October 2025, date of current version 24 October 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3618282

RESEARCH ARTICLE

Real-Time Traffic Insights With Physics-Informed Neural Networks: Integrating the Aw-Rascle Model and LLMs

TEWODROS SYUM GEBRE[1], SIMACHEW ENDALE ASHEBIR[2], JEFFREY BLAY[1], MATILDA ANOKYE[3], VENKTESH PANDEY[4], AND LEILA HASHEMI-BENI[1], (Member, IEEE)

[1]Built Environment, College of Science and Technology, N.C. A&T State University, Greensboro, NC 27411, USA
[2]Data Science and Analytics Program, College of Science and Technology, N.C. A&T State University, Greensboro, NC 27411, USA
[3]Applied Science and Technology Program, College of Science and Technology, N.C. A&T State University, Greensboro, NC 27411, USA
[4]Department of Civil, Architectural, and Environmental Engineering, N.C. A&T State University, Greensboro, NC 27411, USA

Corresponding author: Leila Hashemi-Beni (lhashemibeni@ncat.edu)

This work was supported in part by Microsoft's Accelerate Foundation Models Academic Research Initiative (AFMR) and in part by the National Science Foundation (NSF) grant under Award 2401942.

ABSTRACT Traffic congestion and inefficiencies in transportation networks pose significant challenges to road safety, travel times, and environmental sustainability. Traditional traffic management systems, typically reliant on sparse sensor data and rigid models, often fail to provide accurate, reliable, and user-friendly insights. This paper introduces a novel Physics-Informed Neural Network-Based Traffic State Estimator (PINN-TSE), framework that integrates the Aw-Rascle traffic flow model with advanced machine learning and natural language processing (NLP) techniques. By combining physics-informed modeling with data-driven learning, the framework ensures accurate and physically consistent predictions of traffic density and velocity. A multicomponent loss function balances data fidelity with physical constraints, while Large Language Models (LLMs) generate contextualized and interpretable traffic insights through a chat-based web interface. The system is designed to handle diverse user queries from precise spatio-temporal inputs to broad, general inquiries, making it highly adaptable for real-world deployment. Validated on real-world data from the US-101 highway, PINN-TSE demonstrated strong performance in capturing shockwave dynamics and transitions between traffic regimes. It achieved mean absolute errors (MAE) of 2.4 vehicles per mile (vpm) for density and 3.98 mph for velocity, representing improvements of 60% and 73%, respectively, over purely data-driven models. Furthermore, the shockwave speed error was reduced to 8%, significantly improving the reliability of traffic dynamic predictions. The system's ability to provide actionable insights, such as identifying congestion hotspots and suggesting alternative routes, highlights its practical utility in real-world traffic management. This work makes three key contributions: 1) a robust PINN-TSE framework that embeds physical laws into neural networks, 2) an intuitive LLM-powered interface for real-time traffic interaction, and 3) a demonstration of its effectiveness in real-world settings. By bridging the gap between complex traffic data and human decision-making, this study advances the field of intelligent transportation systems, offering a transformative solution to safer, more efficient, and sustainable traffic management.

INDEX TERMS Physics-informed neural network (PINN), traffic state estimation, Aw-Rascle model, traffic flow modeling, machine learning (ML), traffic information systems, large language model (LLM), hybrid modeling, spatio-temporal modeling, shockwave dynamics, adaptive loss function, driver reaction time, adaptive diffusion, NGSIM dataset, traffic management, intelligent transportation systems.

The associate editor coordinating the review of this manuscript and approving it for publication was Francisco J. Garcia-Penalvo.

I. INTRODUCTION

Traffic modeling serves as a cornerstone of modern transportation systems, providing essential insights into traffic

flow dynamics, congestion patterns, and optimal route planning. Its importance is underscored by the alarming statistic that road incidents in the United States alone account for over 350,000 fatalities annually, highlighting the urgent need for effective traffic management solutions [1]. Beyond safety, the transportation sector contributes significantly to global greenhouse gas emissions, with the World Bank reporting a staggering 20% share in 2023, further emphasizing the environmental imperative to optimize traffic flow and reduce congestion [2]. By improving traffic modeling, these adverse effects can be mitigated through more effective traffic management and planning, making advanced approaches indispensable. However, traditional traffic management systems often fall short due to their reliance on sparse sensor data, which leads to incomplete or unreliable predictions, particularly in areas with limited infrastructure. Sensor failures and coverage gaps exacerbate these challenges, limiting the overall effectiveness of these systems [3].

Current traffic modeling methods, which are used to estimate and predict traffic conditions, face significant limitations that hinder their effectiveness. These sensor-based systems rely on live data sources, such as inductive loops and cameras, meaning the accuracy of traffic modeling heavily depends on the quality and coverage of these inputs. Modern traffic modeling using machine learning approaches, while capable of uncovering complex patterns, are also restricted by the scope and precision of the data they are trained on, limiting their efficiency in estimating traffic, particularly in areas with sparse or incomplete data [4]. On the other hand, physics-based mathematical models, such as the Aw-Rascle model, are efficient for imposing physical laws and provide a theoretical foundation for understanding traffic dynamics; however, they often fail to incorporate complex or unobservable features that affect traffic patterns, reducing their real-world applicability. Beyond these technical challenges, another critical limitation lies in the accessibility of traffic models, as existing systems often fail to provide intuitive and personalized insights for end-users. These limitations collectively underscore the need for a better approach designed to provide reliable and accessible traffic estimates.

This paper aims to address these challenges by combining the strengths of existing methods while overcoming their limitations. By integrating the Aw-Rascle model with neural networks through the Physics-Informed Neural Networks (PINNs) approach, we aim to improve both accuracy and physical consistency in traffic state estimation. Unlike conventional systems that present only current traffic snapshots, our model predicts how traffic evolves over both time and space, offering a fuller picture that reflects past, present, and anticipated traffic conditions.

While map-based and conventional graphical user interfaces (GUIs) are common in current traffic systems, they often require users to interpret dense visual data and typically provide only static or real-time views without forecasting. These interfaces also lack flexibility in answering personalized, context-specific questions. To overcome these limitations, we integrate Large Language Models (LLMs) to enable intuitive, conversational access to traffic predictions. This approach allows users to ask questions in natural language and receive personalized, context-aware insights without needing to navigate complex maps or interfaces thereby enhancing accessibility and user engagement.

To overcome these limitations, our system integrates Large Language Models (LLMs) to facilitate natural language interaction. LLMs enable users to access complex traffic information through intuitive, conversational queries, allowing for flexible, personalized, and contextually relevant responses. This chat-based interface lowers barriers to use by eliminating the need for technical expertise or navigating complicated maps, thus broadening accessibility and enhancing user engagement.

The contributions of this paper are threefold. First, it presents a new Traffic State Estimator (PINN-TSE) that integrates the Aw-Rascle model with neural networks, leading to more accurate and physically consistent traffic predictions. Second, it uses LLMs to generate clear, user-friendly traffic insights, allowing for intuitive interactions in a chat-based web application. Finally, the framework is tested with real-world data, showing its potential to improve real-time traffic management and user experience. By addressing the limitations of current methods, this work offers a more balanced approach to traffic modeling and provides a foundation for future work in intelligent transportation systems.

This article is structured as follows. Section II reviews related work in comparable research areas. Section III outlines the methodology used to develop our model. Section IV presents the experimental results, followed by an in-depth discussion. Finally, Section V concludes the paper by summarizing the major achievements and suggesting directions for future research in this evolving field.

II. RELATED WORK

Traditional traffic management systems have long depended on live sensors, such as inductive loops, radar, and cameras, to monitor traffic conditions along roads and highways [5], [6], [7]. Although these sensors provide valuable real-time data, they often suffer from sparse spatial coverage, creating significant gaps and leaving certain areas unmonitored [8]. Furthermore, sensor failures and data inaccuracies are common, particularly in regions with limited infrastructure, reducing the reliability of traffic management systems [3], [9].

Similarly, real-time navigation apps, such as Google Maps and Waze, which enhance the user experience by utilizing GPS data, also rely heavily on sensor data. These apps help users with timely updates and route optimization, but they too are limited by sensor coverage and data accuracy. Gaps in the sensor network and inconsistencies in the

available data can lead to less reliable guidance, particularly in areas with inadequate sensor infrastructure [3]. Given these limitations, there is a growing need for innovative approaches that can integrate multiple data sources and provide accurate, real-time traffic insights to users [10]. Using advanced technologies, such as artificial intelligence (machine learning) has emerged as a promising solution to address these limitations and develop more robust traffic information systems [6].

Specifically, machine learning-based approaches have gained significant traction in traffic state estimation due to their ability to learn from data and improve performance over time. Early techniques, such as support vector machines (SVMs) and random forests, have been applied with varying degrees of success [11], [12]. For instance, Deng et al. presented a novel approach for freeway traffic state estimation that combines cluster analysis with a multiclass support vector machine (MSVM) [12]. More advanced methods, such as artificial neural networks (ANNs) and Long Short-Term Memory (LSTM) networks, have further improved prediction accuracy by recognizing patterns from historical data [13], [14], [15], [16]. For example, Habtie et al. proposed an ANN framework for real-time traffic state estimation, leveraging cellular network data to gather information across urban areas [13]. Abbas et al. investigated short-term traffic prediction using LSTM networks, demonstrating their ability to capture temporal dependencies in traffic data [14]. Xu et al. introduced a kernel K-nearest neighbors (kernel-KNN) algorithm for real-time traffic state prediction, highlighting the versatility of machine learning techniques in handling diverse datasets [17]. However, the effectiveness of these models depends heavily on the quality and quantity of training data, which can be challenging to obtain in areas with limited sensor coverage [4], [8]. Additionally, machine learning models often lack interpretability, making it difficult to understand the relationships between input data and estimated traffic conditions [18].

Exploring the theoretical and Mathematical models approaches, such as the Lighthill-Whitham-Richards (LWR) model and the Aw-Rascle model, they provide a theoretical foundation for understanding traffic flow dynamics based on conservation laws and driver behavior [19], [20], [21], [22]. The LWR model, a first-order model, describes traffic flow using a continuity equation for vehicle density, while the Aw-Rascle model, a second-order model, extends the LWR framework by incorporating driver reaction time and pressure term [22]. These models are particularly effective in capturing shockwave propagation and congestion patterns, making them valuable tools for traffic state estimation and control [23], [24], [25]. However, purely mathematical models face challenges such as computational complexity and limited adaptability to real-world variations in driver behavior and road conditions [26]. For instance, the Aw-Rascle model assumes a constant driver reaction-time, which may not hold true in all scenarios [22].

To address limitations in both pure machine learning and mathematical traffic models, hybrid approaches combining data-driven techniques with physical laws have gained traction [27]. Physics-Informed Neural Networks (PINNs) represent a key advancement by embedding traffic flow conservation laws—such as mass and momentum conservation—directly into neural network training [28], [29], [30]. Recent studies have applied PINNs specifically to traffic state estimation, demonstrating improved accuracy and physical consistency. Usama et al. developed a PINN-based framework for traffic network state estimation, effectively integrating physics constraints in complex traffic networks [31]. Huang et al. applied physics-informed deep learning to estimate traffic density and speed, validating their approach on real-world data [32]. More recent work by Lu et al. extended PINNs to jointly estimate traffic states and queue profiles, enhancing predictive capabilities for congestion management [33]. Di et al. provided a comprehensive survey of PINN applications in traffic, highlighting emerging trends and challenges [34]. These contributions collectively demonstrate the potential of PINNs to fuse physical interpretability with data-driven flexibility, surpassing traditional models in robustness and real-time applicability [6], [35], [36]. This synergy positions PINNs as a promising hybrid framework for next-generation traffic state estimation and management systems.

Developing a reliable traffic model using PINNs is an important step, but making the results accessible to a wide range of users requires an additional layer. By adding a natural language processing (NLP) interface, we can present the outputs of complex models in a conversational format that users can easily understand and interact with. This makes it possible for people to ask about traffic conditions, predictions, or recommendations using everyday language, without needing to understand the technical details behind the system. In recent years, the integration of NLP into machine learning systems has improved how users interact with technology, enabling the development of chat-based systems and voice assistants [37], [38], [39], [40]. In traffic management, NLP serves as a link between users and the model, translating natural language queries into inputs that the system can process and turning model outputs into clear, helpful responses.

This capability is particularly useful in traffic information systems, where users often want quick and practical answers. NLP-based interfaces allow users to ask questions like ‘‘What’s the traffic on US-101 at 5 PM?’’ and receive direct, personalized answers such as ‘‘Traffic is heavy, average speed 15 mph.’’ More advanced versions can include real-time context, such as incidents or events, to improve the accuracy of these responses. At the same time, the integration of NLP brings specific challenges. These include handling unclear or incomplete user input, matching language with the model’s required format, and producing responses that are both accurate and easy to understand. In addition to improving user

experience, NLP can support traffic authorities by automating common queries and reducing their daily workload, allowing more attention to be given to critical tasks like incident management [41]. When combined with robust modeling, NLP plays a key role in creating traffic systems that are both accurate and accessible.

Hybrid approaches combining physics, machine learning, and natural language processing (NLP) address the limitations of standalone methods. While physics-based models like Aw-Rascle offer mathematical rigor, they often lack adaptability, and machine learning techniques, though flexible, may suffer from reduced interpretability and physical consistency. By integrating these approaches, the proposed framework achieves both accurate traffic state estimation and effective user interaction. Building on Gebre et al. (2024), who applied NLP to structured traffic queries, this study extends the use of NLP to handle complex, multifaceted inputs involving both traffic and contextual factors, bridging a critical gap by leveraging LLMs for real-time, physics-informed insights [6]. The proposed framework integrates the Aw-Rascle model with neural networks to develop a robust PINN-TSE that ensures both accuracy and physical consistency. It also improves user interaction through a chat-based web application, refining NLP techniques to handle ambiguous inputs and extract relevant spatio-temporal information. These components together enhance real-time traffic monitoring and provide users with more informed, context-aware insights, making the framework a valuable addition to modern traffic management systems.

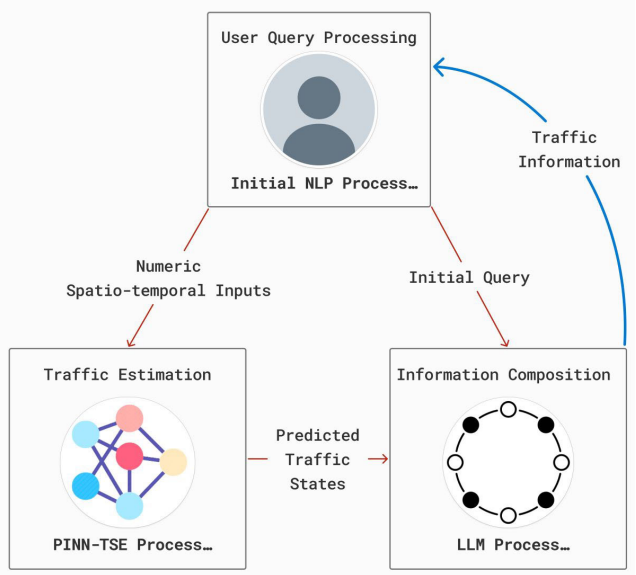


FIGURE 1. Architecture of the proposed traffic information system, integrating natural language processing with a physics-informed neural network (PINN-TSE) for traffic state estimation.

III. METHODOLOGY

The proposed framework introduces a novel approach to traffic information extraction by combining advanced machine learning techniques with physical traffic models. As shown in Figure 1, the framework consists of three key components that work together to process and respond to user queries. Users interact with the system via a chat interface, where they submit traffic-related questions in natural language. The initial NLP process extracts relevant information, such as time and location coordinates, which are necessary for traffic state estimation. This step prepares the data for integration with the subsequent PINN-TSE model.

The PINN-TSE model, which forms the core of the framework, uses these numerical inputs to predict traffic density ($\hat{\rho}$) and velocity ($\hat{v}$). By incorporating the physics-informed principles of the Aw-Rascle model, the PINN-TSE generates predictions that align with established traffic flow laws. This integration of physical constraints aims to address potential limitations of purely data-driven models, especially in scenarios with sparse or noisy data, such as sudden congestion or lane closures.

The final step involves the LLM process, which contextualizes the predicted traffic states alongside the user's original query. This step generates actionable insights, ensuring the system not only provides traffic predictions but also communicates them in a way that is both informative and user-friendly. Thus, the proposed framework, through the integration of these components, offers a flexible and adaptive approach to the extraction of traffic information, addressing key limitations commonly found in traditional systems. The following sections will provide a detailed exploration of each component and its role within the overall framework.

A. INITIAL NATURAL LANGUAGE PROCESSING (NLP) APPLICATION

User queries in traffic prediction systems can vary widely in structure and intent. Some users may provide direct, well-formed requests, while others may submit ambiguous, incomplete, or unconventional queries. This variability poses a challenge in accurately extracting the necessary parameters, such as time and location, to generate meaningful traffic predictions. Without robust query processing, the system risks delivering irrelevant or inaccurate results, undermining user trust and system effectiveness. To address this, we implemented a comprehensive query processing pipeline enabling precise parameter extraction and normalization.

Thus, the objective of the initial NLP processing component is to bridge the gap between user queries and the traffic estimation process by transforming natural language inputs into structured spatiotemporal data for the PINN-TSE. This process involves six key steps to extract spatiotemporal information from user input and convert it into a format compatible with the PINN-TSE model (see Figure 2).

The process begins with input cleaning, where user queries are standardized to remove irrelevant words, typos, or ambiguous phrases. This step ensures clarity and consistency in the input, preparing it for further processing. Following this, key information extraction is performed using

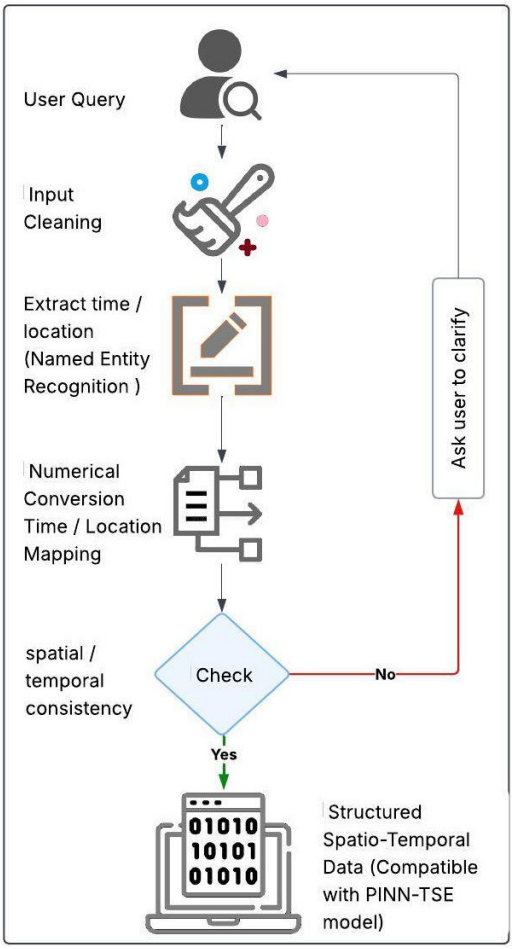


FIGURE 2. Initial NLP processing pipeline for extracting and preparing structured spatiotemporal data from user queries for input into the PINN-TSE model.

entity recognition (NER) and dependency parsing techniques. These methods isolate essential spatio-temporal elements, such as time and location coordinates, from the query. For example, in the query "What's the traffic like on US-101 100 ft upstream at 5 PM?", the location (100 ft) and time (5 PM) are identified and extracted.

Once the spatio-temporal elements are extracted, they are converted into a numerical format to ensure compatibility with the PINN-TSE model. Time is standardized in a 24-hour format, while location values are mapped to specific spatial points. This numerical conversion is critical for enabling the PINN-TSE model to process the data effectively. To maintain accuracy, validation mechanisms are incorporated to verify the consistency and correctness of the extracted information. For example, the system verifies whether the extracted time falls within a valid domain range of the traffic model or whether the location is within the boundaries of the test road. In cases of ambiguity or errors, the system prompts the user to clarify or default to the standard range to ensure robustness in the input data.

Moreover, the initial NLP processing component is designed to handle queries with varying levels of specificity. For queries that include both time and location, the framework directly retrieves traffic predictions for the specified values. When only one parameter is provided, the component samples multiple values for the missing dimension. For instance, if a query specifies only time, the framework samples several locations across the normalized range and provides predictions for each. Similarly, if only location is provided, it samples multiple time points across the normalized range. This adaptability ensures that diverse user inputs are accommodated, delivering meaningful and actionable traffic insights regardless of query completeness.

By automating the extraction and conversion of spatiotemporal information across different input scenarios, the initial NLP processing component helps mitigate potential technical issues. This ensures the framework's adaptability, enabling the system to handle various query formats and variations in user input.

B. PHYSICS INFORMED NEURAL NETWORK TRAFFIC STATE ESTIMATOR (PINN-TSE)

In this study, we propose a PINN-TSE that leverages a feed-forward neural network (FFNN) architecture to estimate traffic dynamics. The framework integrates the Aw-Rascle traffic conservation physical law into the neural network, ensuring that the model adheres to fundamental traffic flow principles. Here we will address the transformation of the Aw-Rascle partial differential equations (PDEs) into a system of regularization terms and their embedding into the network to enforce physical knowledge directly into the FFNN. We also discuss the modeling approach, starting with a simple modeling configuration and progressively introducing complexity to enhance the system's ability to capture complex traffic dynamics.

The Aw-Rascle model is a macroscopic traffic flow model that extends the Lighthill-Whitham-Richards framework. It introduces a pressure-like term into the velocity equation, enabling a more realistic representation of traffic behavior, particularly near shockwaves. The model is based on two key assumptions:

- The traffic flow is described by a continuum model, where the traffic density and flow are treated as continuous variables.
- The traffic flow is governed by the conservation of vehicles, which states that the rate of change of traffic density is equal to the difference between the inflow and outflow rates.

The model is formulated as a system of partial differential equations (PDEs) that describe the evolution of traffic density and flow over time and space. The mathematical formulation of the Aw-Rascle model is as follows:

$$\frac{\partial \rho}{\partial t} + \frac{\partial q}{\partial x} = 0 \tag{1}$$

$$q = \rho v(\rho) \tag{2}$$

$$\frac{\partial}{\partial t}(v + p(\rho)) + v\frac{\partial}{\partial x}(v + p(\rho)) = 0 \tag{3}$$

where ρ is the traffic density, q is the traffic flow, v is the velocity, and $p(\rho)$ is the pressure term.

The pressure term is given by:

$$p(\rho) = p_0 \left(1 - \frac{\rho}{\rho_{\max}}\right) \tag{4}$$

where p_0 is a constant (commonly the slope of the basic speed density diagram) and $\rho_{\max}$ is the maximum traffic density [42].

To integrate the physics of traffic flow into the neural network, we first derive mathematical constraints based on the conservation equations of the Aw-Rascle model (Equations 1 to 4). These constraints form the basis for residuals, which are introduced as regularization terms in the training process (Equations 5 and 6).

$$\hat{f}_1(t, x; \theta) = \frac{\partial \hat{\rho}}{\partial t} + \frac{\partial \hat{q}}{\partial x}, \tag{5}$$

$$\hat{f}_2(t, x; \theta) = \frac{\partial}{\partial t}(\hat{v} + \hat{p}(\hat{\rho})) + \hat{v}\frac{\partial}{\partial x}(\hat{v} + \hat{p}(\hat{\rho})) \tag{6}$$

where $\hat{\rho}$ and $\hat{v}$ denote the traffic density and speed estimated by the model. These residuals are integrated into the PINN-TSE loss function to enforce physical consistency during training. Thus, the residuals $\hat{f}_1$ and $\hat{f}_2$ represent constraints based on physical law that allow the model to produce predictions that remain consistent with the governing traffic flow equations.

Accordingly, the multicomponent loss function given in Equation 7 balances data fidelity with physical laws and thus includes four key terms: observed data loss, boundary condition loss, initial condition loss, and residual (PDE) loss. Each term is weighted to control its contribution to the optimization process, as detailed in Equations (7 through 11).

$$\mathcal{L}_\theta = \text{MSE}_{\text{obs}} + \text{MSE}_{\text{bc}} + \text{MSE}_{\text{ic}} + \text{MSE}_{\text{pde}} \tag{7}$$

$$\text{MSE}_{\text{obs}} = \frac{1}{N_{\text{obs}}} \sum_{i=1}^{N_{\text{obs}}} \Big[\beta_1 \left(\hat{\rho}(t_i, x_i; \theta) - \rho_i\right)^2 + \beta_2 \left(\hat{v}(t_i, x_i; \theta) - v_i\right)^2 \Big] \tag{8}$$

$$\text{MSE}_{\text{bc}} = \frac{1}{N_{\text{bc}}} \sum_{k=1}^{N_{\text{bc}}} \Big[\gamma_1 \left(\hat{\rho}(t_k, 0; \theta) - \hat{\rho}(t_k, 1; \theta)\right)^2 + \gamma_2 \left(\hat{v}(t_k, 0; \theta) - \hat{v}(t_k, 1; \theta)\right)^2 \Big] \tag{9}$$

$$\text{MSE}_{\text{ic}} = \frac{1}{N_{\text{ic}}} \sum_{l=1}^{N_{\text{ic}}} \Big[\delta_1 \left(\hat{\rho}(0, x_l; \theta) - \rho_l^0\right)^2 + \delta_2 \left(\hat{v}(0, x_l; \theta) - v_l^0\right)^2 \Big] \tag{10}$$

$$\text{MSE}_{\text{pde}} = \frac{1}{N_{\text{pde}}} \sum_{j=1}^{N_{\text{pde}}} \Big[\alpha_1 \hat{f}_1(t_j, x_j; \theta)^2 + \alpha_2 \hat{f}_2(t_j, x_j; \theta)^2 \Big] \tag{11}$$

In these equations:

- N_{obs} is the number of observations used for traffic density and velocity predictions.
- $\hat{\rho}(t_i, x_i; \theta)$ and $\hat{v}(t_i, x_i; \theta)$ are the predicted traffic density and velocity at time t_i and position x_i, respectively, while ρ_i and v_i are the actual observed values obtained through aerial traffic monitoring.
- β_1 and β_2 are weighting factors that can be adjusted to prioritize the prediction of traffic density or velocity. A higher weight indicates greater priority. By default, $\beta_1 = \beta_2 = 0.5$.
- N_{bc} is the number of boundary condition samples used to enforce consistency at the spatial domain boundaries.
- $\hat{\rho}(t_k, 0; \theta)$ and $\hat{\rho}(t_k, 1; \theta)$ are predicted densities at the left and right spatial boundaries at time t_k; similarly, $\hat{v}(t_k, 0; \theta)$ and $\hat{v}(t_k, 1; \theta)$ are predicted velocities.
- γ_1 and γ_2 are weighting factors for enforcing boundary conditions. Default values are $\gamma_1 = \gamma_2 = 0.5$.
- N_{ic} is the number of initial condition samples along the spatial domain.
- $\hat{\rho}(0, x_l; \theta)$ and $\hat{v}(0, x_l; \theta)$ are predicted initial traffic density and velocity at location x_l, while ρ_l^0 and v_l^0 are the corresponding known initial values.
- δ_1 and δ_2 are weighting factors for the initial condition loss terms. Default values are $\delta_1 = \delta_2 = 0.5$.
- N_{pde} is the number of samples used for evaluating the PDE residuals $\hat{f}_1$ and $\hat{f}_2$.
- $\hat{f}_1(t_j, x_j; \theta)$ and $\hat{f}_2(t_j, x_j; \theta)$ are the residuals derived from the governing partial differential equations evaluated at time t_j and position x_j.
- α_1 and α_2 are weighting factors for the PDE residuals, typically in the range [0.0, 1.0], with $\alpha_1 + \alpha_2 = 1$. By default, $\alpha_1 = \alpha_2 = 0.5$.

The neural network architecture proposed for the PINN-TSE model, depicted in Figure 3, consists of an FFNN with three hidden layers. In the initial configuration, each hidden layer consisted of 64 neurons. This initial configuration was chosen based on best practices in neural network modeling and computational efficiency considerations as in [6]. The input layer takes spatio-temporal independent variables (t, x), representing time and space coordinates, while the output layer predicts traffic density $(\hat{\rho})$ and velocity $(\hat{v})$. The three-layer architecture strikes a balance between model complexity and computational efficiency, providing sufficient capacity to capture nonlinear traffic dynamics while avoiding overfitting. The choice of 64 neurons per layer is informed by empirical evidence from similar PINN applications [6]. The design supports scalability and real-time application by enabling learning from both observed data and traffic flow physics, which promotes physically plausible predictions.

To validate the proposed architecture, the methodology is divided into two distinct phases. In the first phase, the model is tested using a synthetic dataset specifically designed to evaluate its stability and performance. This dataset includes carefully crafted initial and boundary conditions

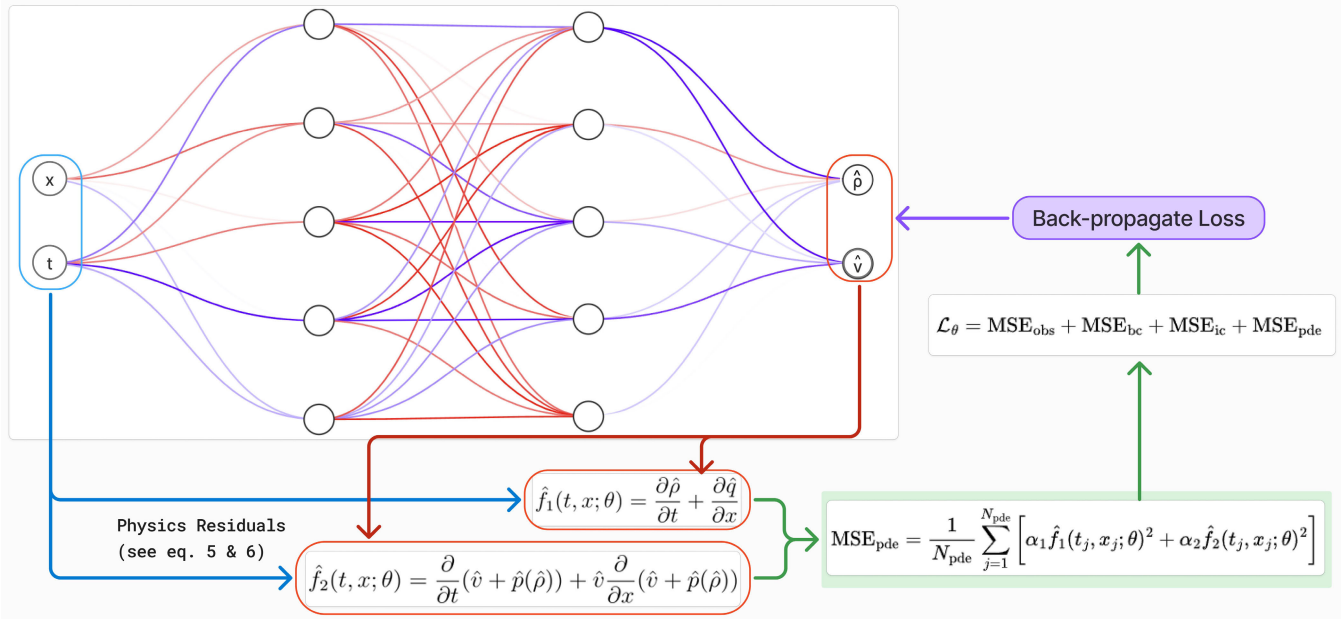


FIGURE 3. Architecture of the Physics-Informed Neural Network Traffic State Estimator (PINN-TSE), featuring a feedforward neural network (FFNN) for predicting traffic density and velocity from spatiotemporal inputs.

to rigorously test the integration of Aw-Rascle-driven loss terms. The primary objectives of this phase are to identify the optimal model architecture, refine the optimization strategy, and establish an effective input data handling approach. Performance is assessed using both quantitative metrics, such as mean squared error (MSE) curves, and qualitative analysis, including visual inspection of spatio-temporal traffic state estimation contours. The synthetic data provides a controlled environment to evaluate the model's ability to capture shockwave dynamics, transitions between traffic regimes, and adherence to physical constraints. This phase is planned to ensure that the architecture is robust and efficient before deployment in real-world scenarios.

In the second phase, the best-performing architecture is implemented on a real-world dataset, specifically traffic data from the US-101 freeway. To simulate practical scenarios with limited sensor coverage, only 8% to 10% of shockwave points from the dataset are used as training data. These points are identified using an algorithm that we design to detect shockwaves, which processes vehicle trajectories to locate abrupt speed differentials indicative of shockwaves. By focusing on these critical points, the model is trained to generalize effectively even in data-sparse environments, ensuring its applicability to real-world traffic management systems. This approach not only validates the model's practical utility but also demonstrates its ability to provide reliable predictions in scenarios where traditional sensor-based systems often fall short.

To highlight the benefits of the physics-informed approach, the performance of the PINN-TSE model is compared to that of a similar architecture without the physics component. This comparison evaluates the impact of integrating physical constraints into the neural network, particularly in scenarios with limited or noisy data. Metrics such as RMSE, MAE, and shockwave speed estimation accuracy are used to quantify the improvements brought about by the physics-informed approach. In addition, visual comparisons of predicted versus ground truth traffic states, such as contour plots of density and velocity fields, provide qualitative insights into the model's ability to capture sharp gradients and shockwave dynamics.

In general, following the initial NLP process, which extracts pure numeric spatio-temporal inputs from user queries, the PINN-TSE takes these values to predict traffic states. These predictions are then passed back to the system, where they are interpreted by a LLM to provide actionable traffic insights to the user.

C. LARGE LANGUAGE MODEL (LLM) INTEGRATION

The LLM Process plays a crucial role in transforming user queries and predicted traffic states into actionable and user-friendly insights. This component takes the initial user input, along with predicted traffic speed and density values from the PINN-TSE model, and generates contextualized traffic insights through prompt engineering. The process begins with parsing the user's query to understand the context and key parameters, such as the time window and location for traffic predictions. In addition to the initial NLP processing, this step ensures that the input falls within the predefined domain (US-101, Los Angeles, 3:00 PM to 3:15 PM, and a 2,100-foot stretch of road). If the input deviates from this domain, the user is prompted to refine their request using a system-generated prompt: "This chat-bot is developed to provide traffic information within the domain of 3:00 PM

to 3:15 PM on US-101, covering the 2,100-foot test road section. Ensure that your query is within these boundaries.''

Once the query is validated, the system integrates the predicted traffic conditions (density and speed) from the PINN-TSE model. These predictions are computed for the designated time window and location range, providing traffic density in vehicles per mile per lane and speed in miles per hour. The traffic model's output is used to assess whether the road segment is congested or uncongested based on predefined thresholds, such as comparing the density with the optimum traffic density (ρ_{opt}). A system prompt guides this assessment: ''You are an AI traffic guidance assistant. Given the query, the traffic predictions are for a 2,100-foot section of US-101 in Los Angeles between 3:00 PM and 3:15 PM. The predictions include density (in vehicles per mile per lane) and speed (in miles per hour). If traffic density exceeds (ρ_{opt}) vehicles per mile per lane, mark the area as congested. Otherwise, it is considered uncongested. If multiple predictions are available, highlight the maximum and minimum values for density and speed, along with their corresponding time and location.''

Using these predictions, the system synthesizes the data to generate actionable insights, such as identifying potential delays and congestion points. The prompts at this stage ensure that the insights are formatted and presented clearly and concisely. For example, if the query is ''What's the traffic like on US-101 near Los Angeles at 3:10 PM?'' and the PINN-TSE predicts high density and low speed, the LLM might respond with ''Traffic on US-101 near Los Angeles at 3:10 PM is congested, with a density of 45 vehicles per mile per lane and an average speed of 15 mph. Expect significant delays; consider alternative routes.''

The system is also designed to handle user follow-ups dynamically. If a user requests further clarification or more specific suggestions, the system adjusts its prompts and responses accordingly, ensuring that the conversation remains focused on the predefined traffic domain and time constraints. By integrating prompt engineering with the PINN-TSE predictions, the LLM Process ensures that the system provides accurate, contextualized, and actionable traffic insights, enhancing the user experience and supporting real-time traffic management decisions.

The seamless integration of the three components, initial NLP processing, traffic prediction via the PINN-TSE model, and LLM-powered information composition, creates a powerful and user-friendly interface for traffic information inquiry. The initial NLP step ensures accurate interpretation and contextualization of user queries, while the PINN-TSE model delivers precise and reliable traffic predictions. The LLM component enriches these predictions with personalized insights and follow-up capabilities, transforming raw data into actionable and user-friendly responses. Together, these steps bridge the gap between complex traffic data and everyday decision making, assisting users with informed decisions in real-time.

IV. RESULTS AND DISCUSSION

This section details the implementation of the three-step workflow outlined in the system architecture: initial NLP, traffic prediction using the PINN-TSE, and LLM-powered traffic information composition. Building on the methodology, we focus on the practical steps to develop and validate the system, ensuring that it transforms user queries into accurate and actionable traffic insights.

A. INITIAL USER QUERY PROCESSING

The implementation of the user query processing system began with the application of the `spaCy` NLP library. This library was selected for its lightweight architecture and efficient extraction of time and location information from user queries. `spaCy`'s named entity recognition module (NER) was used to identify entities labeled as `"TIME"` and `"QUANTITY"` that enable the parsing of straightforward queries such as *"What is the traffic at 4:04 PM and 251 feet?"*

For such queries, the system extracts time and location information, which are then normalized to match the system's internal representations. Time values are converted into `datetime` objects, and the difference in minutes from a reference time (3:00 PM) is calculated. This difference is normalized to a range of [0, 1], where 0 corresponds to 3:00 PM and 1 to 3:15 PM. Similarly, location values are normalized by dividing the extracted number by 2100, representing the length of the test road section. This normalization helps for consistency across queries and supports integration with downstream traffic prediction models. To improve reliability, the system validates the extracted values to ensure that they fall within the expected domain ranges mentioned above. If validation fails, the user is informed about the valid prediction range, and the system defaults to a general traffic prediction by sampling multiple time and location values.

The `spaCy`-based approach was effective for queries following a structured format, where numerical values were clearly separated from their units (e.g., *"251 feet"*). However, during testing, we observed performance degradation when queries deviated from this format. For instance, `spaCy` struggled with inputs like *"251feet"* (no space) or *"two hundred fifty-one feet"* (numbers in words), failing to extract relevant entities.

As the system evolved, additional challenges emerged. Users often submitted unrelated queries, such as *"What is the weather today?"* or *"Tell me a joke."* Processing these through the entire pipeline not only consumed computational resources but also generated irrelevant output.

To address these limitations, we integrate a large language model (LLM) into the pipeline for **query triage** as a preliminary step that determines whether a query is traffic-related. The LLM analyzes the intent and context of the query, forwarding only relevant queries for further processing. For instance, *"What is the traffic at 3:04 PM?"* is flagged as traffic-related, while unrelated queries are filtered early. Additionally, prompt engineering was used to instruct the

LLM to extract time and location data in predefined formats: time in "HH:MM AM/PM" (e.g., *"4:04 PM"*) and location in "XXXX feet" (e.g., *"251 feet"*). Testing showed the LLM effectively handled unconventional queries that spaCy could not process.

Through iterative testing and refinement, the query processing pipeline evolved into a robust system capable of handling the variability of real-world user input. By combining the flexibility of LLMs for nuanced understanding and incorporating triage, validation, and normalization, the framework achieved reliable and adaptable query processing. This integration ensured accurate input preparation for the PINN-TSE model, improving the system's ability to deliver precise traffic predictions.

B. PROGRESSIVE REFINEMENT OF PINN-TSE MODEL FOR CAPTURING TRAFFIC SHOCKWAVES

To evaluate the robustness of the proposed PINN-TSE framework, a series of controlled experiments were conducted using synthetic initial and boundary conditions. These tests served as a sanity check for the physics-informed regularization proposed in Methodology and a diagnostic tool to observe the model's behavior in capturing key traffic dynamics, particularly shockwave propagation and regime transitions.

The evaluation followed a progressive strategy, starting with a simple baseline model and gradually introducing additional physical realism and complexity. Each trial built on the previous one, addressing observed limitations and refining the model architecture, loss formulation, and physical consistency. The lessons learned from these trials inform the choice of hyperparameter and model design choices for subsequent real-world applications.

Trial 1: Baseline PINN Model with Default Settings

The first trial used a simple architecture: a fully connected neural network with three hidden layers of 64 neurons each and Tanh activations. The loss function included unweighted PDE residuals derived from the Aw-Rascle traffic equations, with a nonlinear pressure term $P(\rho) = \rho^{\gamma}$. Boundary conditions were imposed as fixed Dirichlet values at both ends of the spatial domain ($\rho_{\text{left}} = \rho_{\text{right}} = 0.5$). The model was trained using the AdamW optimizer with a learning rate of 1×10^{-3}.

The results show that the regularization term of the PINN-TSE model converges smoothly, indicating no vanishing or exploding gradients and confirming the effectiveness of the applied physics-based regularization in promoting training stability (see loss curve (1) in Figure 4). While the model achieved stable convergence and minimized PDE residuals over short time horizons, it failed to capture sharp traffic dynamics in the rest of the predicted domain. The results exhibited overly smooth transitions in density and speed (Speed-map (1) and Density-map (1) in Figure 5). We attributed the limitations to a combination of three factors: (1) the smoothing effect of the Tanh activation function and/or the pressure term, (2) the use of equally weighted residuals, and (3) hard-coded boundary conditions that may have restricted the formation of sharper wave features.

Trial 2: Improved Initial and Boundary Condition Handling

To address the smoothing limitations of Trial 1, the second trial introduced several enhancements. Boundary and initial conditions were more robustly enforced using a tolerance-based formulation, with separate loss terms for each. The initial condition was modeled using a Gaussian distribution: $\rho_{\text{init}} = e^{-100(x-0.5)^2}$, representing a localized density peak at $t = 0$ to facilitate catching the propagation of shockwaves. Also, the linear pressure term ($P(\rho) = p_0(1 - \rho/\rho_{\max})$) was retained to reduce artificial over-smoothing effects.

The model architecture was simplified to a four-layer network with 20 neurons per layer, retaining Tanh activations. This change aimed to reduce excessive nonlinearity while preserving representational capacity. Adam optimizer with a learning rate of 0.001 was used and automatic differentiation computed the PDE residuals at uniformly sampled collocation points.

These changes significantly improved the model's ability to capture sharp gradients, with better representation of shockwave fronts in both density and velocity fields (see loss curves (2) in Figure 4 and Speed-map (2) and Density-map (2) in Figure 5). In addition, Table 1 summarizes the modifications and the resulting improvements.

Trial 3: Capturing Realistic Shockwaves via Source Terms and Driver Behavior

The third trial introduced physical extensions to further enhance the model's ability to capture traffic dynamics. A Gaussian-shaped source/sink term was added to the mass conservation equation to simulate vehicle inflow and outflow:

$$\partial_t \rho + \partial_x(\rho v) = \text{source\_term}(x, t) - D\partial_{xx}\rho,$$

where D is a diffusion coefficient (with initial value of $D = 0.01$) to suppress high-frequency noise and improve numerical stability.

In addition, a driver reaction-time term was added to the momentum equation to capture how drivers adjust their speeds toward equilibrium velocity:

$$\partial_t v + v\partial_x v + \frac{v - v_{\text{eq}}}{\tau} + \partial_x P(\rho) = 0,$$

where $v_{\text{eq}} = v_{\max}(1 - \rho/\rho_{\max})$ is the desired velocity and τ is the reaction-time parameter (initially set to $\tau = 0.5$) [43], [44].

The loss function was further refined by introducing adaptive weights for the PDE, boundary and initial condition terms to better balance the training process (as stated in Equations 7 to 11).

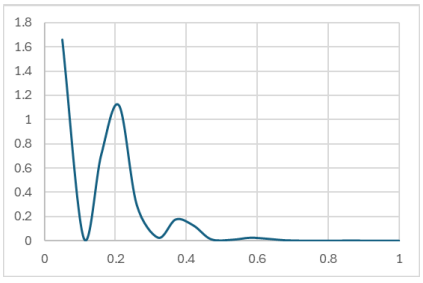
These additions result in a significantly more realistic traffic model. The results showed better handling of shockwave propagation, sharper velocity transitions, and better representation of stop-and-go traffic behavior (see loss curve

TABLE 1. Comparison of model changes and performance improvements between the first and second trials.

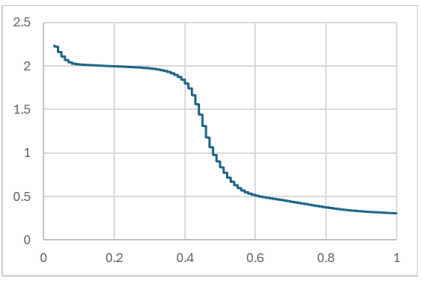
Aspect	First Trial	Second Trial
Objective	Establish baseline model; test simple architecture and PDE enforcement.	Improve shockwave resolution; address smooth transitions caused by default architecture.
Loss Function	Unweighted PDE residuals; fixed IC/BC as hard constraints.	Tolerance-based IC/BC enforcement; separate loss terms for IC, BC, and PDE.
Pressure Term	$P(\rho) = \rho^{\gamma}$; acts as artificial viscosity.	$P(\rho) = p_0(1 - \rho/\rho_{\max})$; improved driver behavior modeling.
Source/Sink Term	Not included.	Not included.
Reaction-Time Term	Not included.	Not included.
Diffusion Term	Not included.	Not included.
Initial Condition	Gaussian ρ; velocity via $v = v_{\max}(1 - \rho/\rho_{\max})$.	Same IC, but explicitly enforced with loss term: $\rho_{\text{init}} = e^{-100(x-0.5)^2}$.
Boundary Conditions	Fixed Dirichlet values: $\rho_{\text{left}} = \rho_{\text{right}} = 0.5$.	Tolerance-based enforcement of Dirichlet conditions.
Neural Network Architecture	3 hidden layers, 64 neurons/layer; `Tanh` activation.	4 hidden layers, 20 neurons/layer; `Tanh` retained.
Optimizer and Training	AdamW with learning rate decay.	Adam with fixed learning rate (0.001).
Shockwave Representation	Overly smooth; failed to capture sharp gradients.	Sharper transitions in density and velocity fields.
Visualization Quality	Diffusive results; lacked realistic transitions.	Clearer wavefronts in space-time plots.

TABLE 2. Comparison of model changes and performance improvements in the third trial.

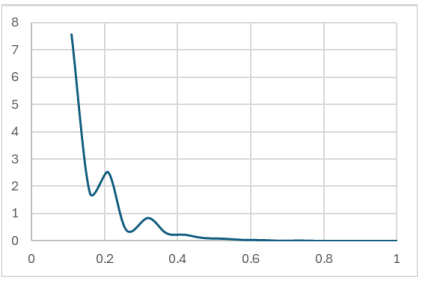
Aspect	Second Trial	Third Trial
Objective	Refine IC/BC handling and enhance shockwave dynamics.	Capture regime transitions; model vehicle inflow/outflow and realistic velocity adjustment.
Loss Function	Explicit loss for IC/BC; unweighted PDE residuals.	Adaptive weighting: $\lambda_{\text{pde}}, \lambda_{\text{ic}}, \lambda_{\text{bc}}$; added regularization.
Pressure Term	$P(\rho) = p_0(1 - \rho/\rho_{\max})$ retained.	Same term; more effective integration into momentum equation.
Source/Sink Term	Not included.	Added Gaussian source_term(x, t) to simulate vehicle entry/exit.
Reaction-Time Term	Not included.	Added $\frac{v - v_{\text{eq}}}{\tau}$ to momentum equation ($\tau = 0.5$).
Diffusion Term	Not included.	Added $D \cdot \partial_{xx}\rho$ for smoothing; $D = 0.01$.
Initial Condition	Explicit Gaussian IC enforced via loss.	Same IC retained.
Boundary Conditions	Tolerance-based enforcement.	Same approach retained.
Neural Network Architecture	4 layers, 20 neurons/layer; `Tanh`.	Same, but final layer uses ReLU for sharper transitions.
Optimizer and Training	Adam (LR = 0.001).	Adam + adaptive loss weighting for balanced convergence.
Shockwave Representation	Clearer wave transitions than Trial 1.	Sharper gradients; captures discontinuities and congestion fronts.
Visualization Quality	Improved over Trial 1.	High-fidelity transitions; realistic space-time dynamics.



(1) Initial modeling loss curve.



(2) Second modeling loss curve.



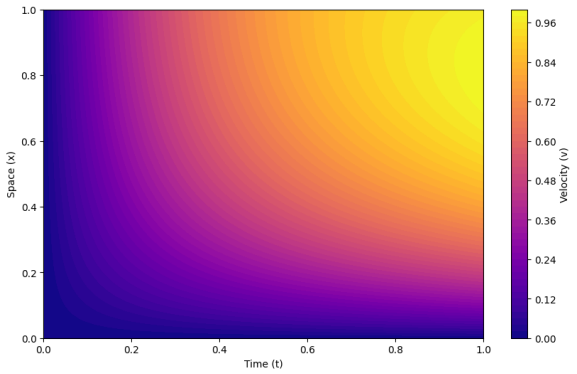
(3) Third modeling loss curve.

FIGURE 4. Training loss curves for the synthetic data modeling trials. *Training epochs are normalized for visualization purposes.

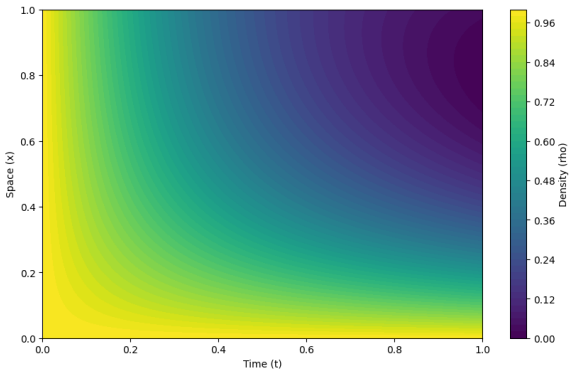
(3) in Figure 4 and Speed-map (3) and Density-map (3) in Figure 5). In addition, detailed comparisons are provided in Table 2.

The iterative approach, starting from a baseline architecture and progressively introducing physical realism and structural refinements, proved effective in addressing

Trial 1: Initial Model — Basic architecture and fixed boundary conditions

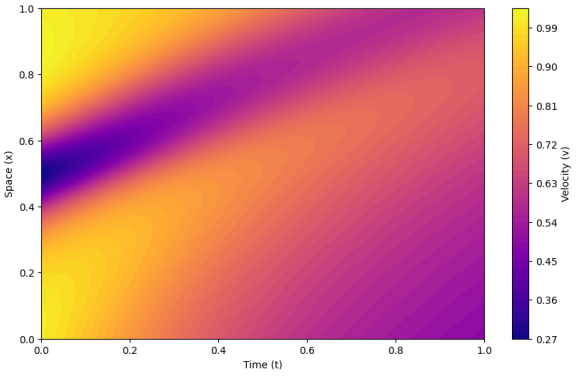


Speed (1): Overly smooth transitions; poor shockwave resolution.

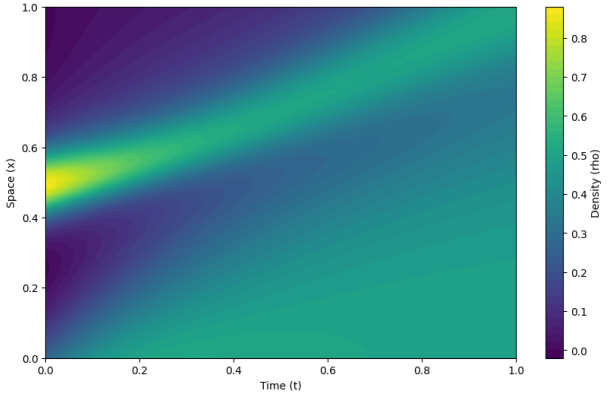


Density (1): Diffused congestion fronts; lacks sharp gradients.

Trial 2: Refined Model — Improved boundary enforcement and loss formulation

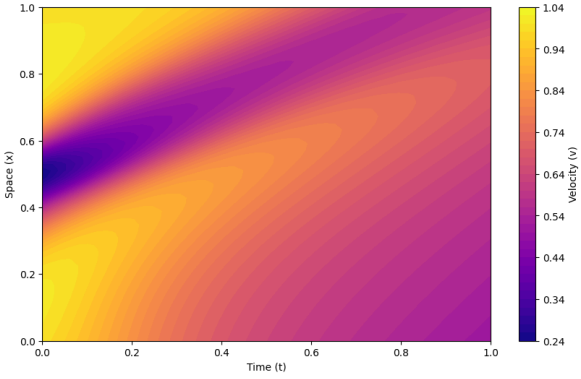


Speed (2): Better wave clarity; improved shock propagation.

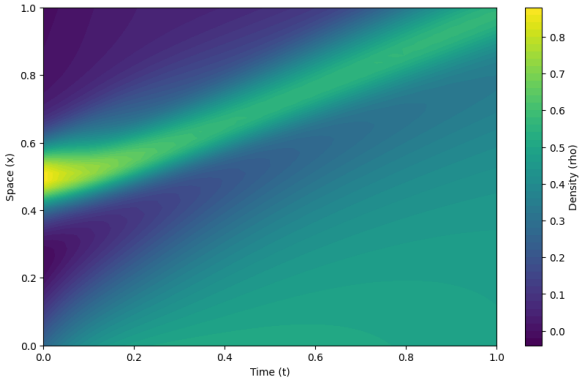


Density (2): Sharper transitions; more realistic congestion profiles.

Trial 3: Optimized Model — Added source/sink and driver reaction dynamics



Speed (3): Sharp congestion fronts; captures wave discontinuities.



Density (3): Realistic spatio-temporal congestion propagation.

FIGURE 5. Visualization of PINN-TSE modeling results across three progressive trials. Each row corresponds to a trial and shows the predicted traffic speed and density over normalized space and time domains.

modeling limitations. Each trial incrementally improved the model's ability to capture key traffic phenomena such as shockwaves, density transitions, and flow discontinuities.

The final version of the PINN-TSE framework, which incorporates adaptive loss weighting, physically informed terms, and robust boundary condition enforcement, is considered for the next real-world application. In the next phase, this optimized model is applied to real traffic data from the US-101 highway to take advantage of its capabilities to provide accurate and interpretable traffic state estimates in practical settings.

C. APPLICATION OF REFINED PINN-TSE MODEL TO NGSIM TRAFFIC DATA

Building on the success of the third PINN-TSE model with synthetic data, we applied the lesson learned to real-world traffic dynamics using the Next Generation

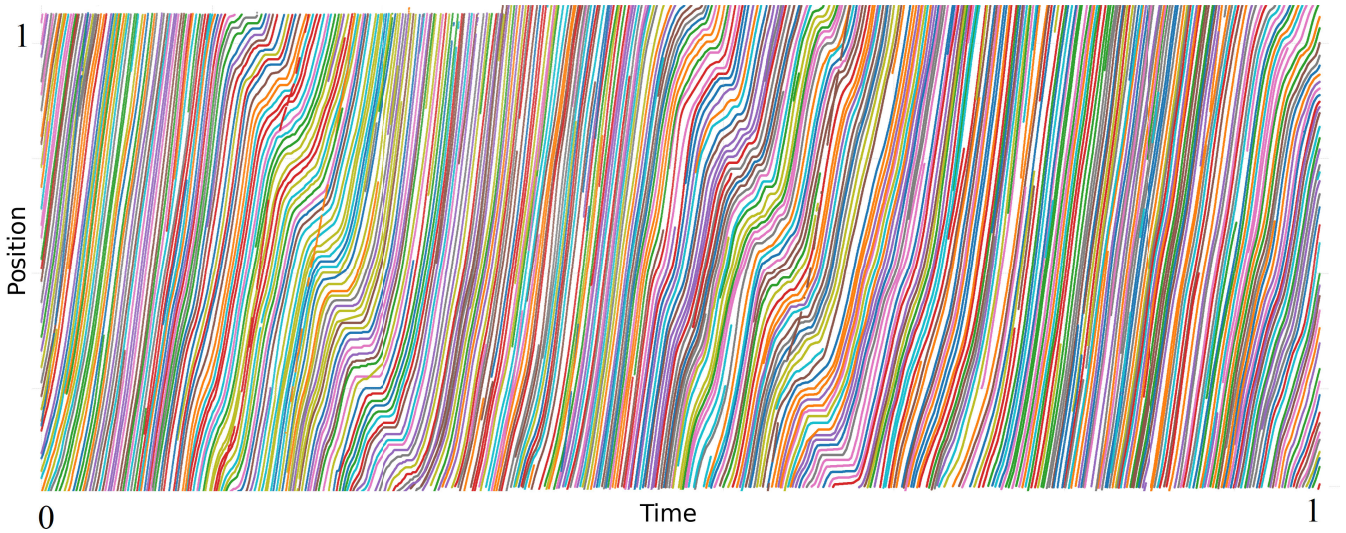


FIGURE 6. Time–space diagram of vehicle trajectories on the NGSIM test segment. Each line represents the path of a single vehicle over time, plotted using normalized time and spatial coordinates. The smoothed trajectories highlight lane-wise movements and reveal key traffic flow patterns such as congestion buildup and dissipation during the study period.

Simulation (NGSIM) dataset. This application involved a hybrid physics-informed learning framework that integrates physical laws with data-driven insights. The overall pipeline begins with rigorous preprocessing of the NGSIM dataset, followed by model configuration tuning, training, and evaluation under realistic traffic conditions.

1) NGSIM DATASET PROCESSING

The NGSIM dataset provides high-resolution vehicle trajectory data collected on a 640-meter (2,100 ft) segment of the US-101 freeway in Los Angeles, California. The data were recorded on April 13, 2005, during the afternoon peak (14:50–15:05), capturing vehicle positions, speeds, accelerations, lane changes, and vehicle types at a sampling rate of 10 Hz. In total, the dataset includes detailed trajectories for more than 2,000 vehicles across five mainline lanes, making it ideal for microscopic traffic flow analysis. For this study, we focused on Lane 2, isolating data within a 15-minute window (15:00–15:15) to analyze shockwave behavior under moderately congested conditions. This segment was selected for its clear transitions between free flow and congestion, which are critical for evaluating the model's ability to capture shockwave dynamics.

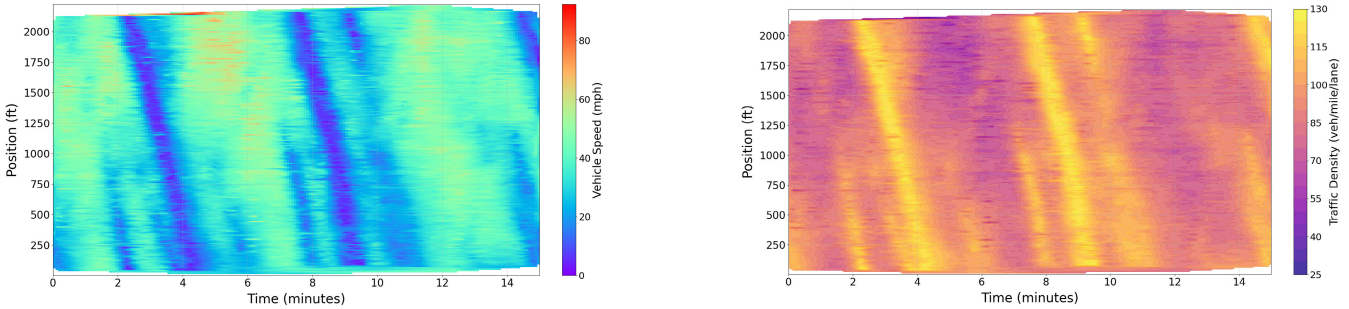
Preprocessing the raw NGSIM data is essential due to measurement noise, sensor artifacts, and inconsistent formatting. The first step involves standardizing time and spatial coordinates, followed by normalization of key numerical attributes (e.g., speed, acceleration, and headway time). To improve data reliability, we employed outlier detection using a Z-score thresholding method. This process flagged and removed approximately 0.07% of speed values, mitigating the impact of extreme or erroneous data points on subsequent analyses.

To further address trajectory noise, particularly in speed and acceleration profiles, we applied the Savitzky-Golay filter. This smoothing technique preserves local structure while reducing fluctuations caused by sensor jitter or detection errors. As illustrated in Figure 6, the smoothed time-space diagram reveals clearer spatio-temporal vehicle dynamics, highlighting phenomena such as bottlenecks, platooning, and congestion waves. These improved visualizations enhance interpretability and aid in the identification of traffic regimes critical for model training.

Following smoothing, traffic state variables such as density and speed were reconstructed to generate heatmaps over the spatio-temporal domain, shown in Figure 7. These visualizations serve as empirical references for evaluating the model's predictive accuracy.

To parameterize the physics-informed loss functions, we derived fundamental traffic relationships from the dataset. Specifically, we fitted Greenshield's model to the speed–density relationship, estimating the free-flow speed (v_f), jam density (k_j), and flow capacity ($q_{\max}$). As shown in Figure 8, the fitted model yielded a free-flow speed of 59 mph and a jam density of 137 vehicles/mile. These parameters were integrated into the PDE residual term of the PINN to reflect realistic traffic dynamics.

For training, the model was not exposed to the entire dataset. Instead, a shockwave-informed sampling strategy was used to extract 8–10% of the preprocessed data concentrated near traffic disturbances. An automated detection algorithm localized shockwave regions using parallelized spatio-temporal binning. Speed drops exceeding 20 mph within 90-foot segments and 0.1-minute intervals were flagged as shockwave events. These locations informed the selection of initial and boundary condition (IC/BC) training points based on high values of the temporal speed



(a) Reconstructed traffic speed over space and time.

(b) Reconstructed traffic density over space and time.

FIGURE 7. **Spatio-temporal heatmaps of reconstructed traffic states from smoothed data: (a) traffic speed and (b) traffic density. These empirical visualizations are used as references for evaluating the predictive accuracy of the trained PINN-TSE model.**

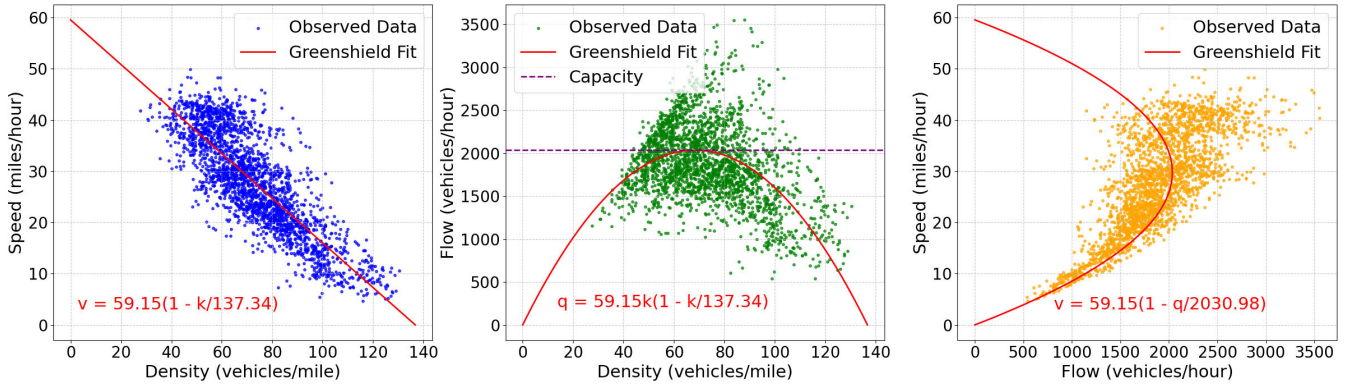


FIGURE 8. **Greenshield curve-fitting results. Estimated parameters are used for constructing the physics regularization term in the PINN loss function.**

gradient ($\partial v/\partial t$). This focused approach enabled the model to prioritize learning from dynamic, high-information regions, improving generalization and physical consistency in its predictions.

With shockwave locations identified, the PINN architecture was initialized using these observation values and IC/BC points, while collocation points covered the full observational domain to enforce physical constraints. Key improvements included hybrid regularization, reaction-time calibration, and diffusion scaling. In addition, the driver reaction time (τ) was set to dynamically adjust during training based on the observed propagation of stop-wave speeds, allowing the model to capture realistic driver behavior.

Furthermore, the diffusion coefficient (D) was made spatially adaptive through a learned mapping ($D(x,t) = f_\theta(x,t)$), where (f_θ) is a neural network that predicts diffusion values based on spatial and temporal inputs. This adaptation allowed the model to handle heterogeneous road conditions more effectively. During training, the neural network learned the mapping (f_θ) by minimizing the loss function, which included terms for physical consistency and data fidelity. Areas with sharp gradients or high traffic density were assigned higher diffusion values to stabilize the numerical solution, while smoother regions retained lower values to preserve detail (see Table 3).

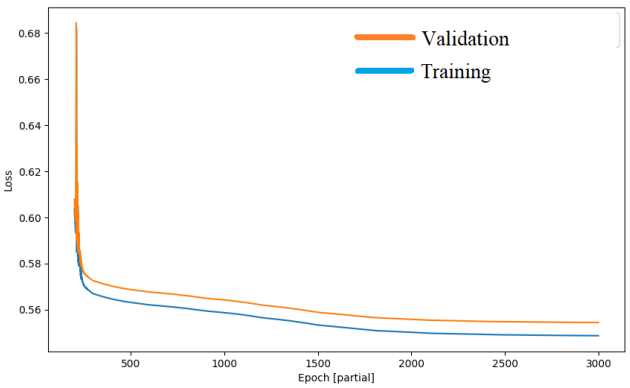


FIGURE 9. **Training and validation loss curves for the PINN-TSE model. The smooth and parallel decline of both curves indicates stable convergence and absence of overfitting.**

2) MODELING PERFORMANCE AND COMPARATIVE ANALYSIS

As shown in Figure 9, both training and validation loss curves exhibit smooth and consistent convergence without signs of divergence or instability. The absence of overfitting is indicated by the close alignment between the two curves, suggesting that the model generalizes well across different data splits.

TABLE 3. Comparison of latest implementation (US-101 Data) with previous modeling (synthetic data).

Aspect	Previous Modeling (Synthetic Data)	Latest Implementation (US-101 Data)
Objective	Improve shockwave gradients and transitions between traffic regimes	Model real-world traffic dynamics on US-101; enhance shockwave resolution and regime transitions
Data Source	Synthetic dataset with predefined initial and boundary conditions	Real-world trajectory data from US-101, processed for shockwave localization
Shockwave Detection	Not applicable (shockwaves predefined in synthetic data)	Optimized detection algorithm with spatio-temporal binning (0.1-min time bins, 20-foot spatial bins)
Initial/Boundary Conditions	Predefined IC/BC from synthetic dataset	IC/BC and shockwave points derived from detected shockwave locations in the US-101 data
Loss Function	Adaptive weights for PDE residuals, IC, and BC; regularization for shockwave gradients	Hybrid loss with data fidelity term; adaptive weights for PDE, IC, BC, and observed shockwave data terms (λ_{data})
reaction-time	Fixed driver reaction-time ($\tau = 0.5$)	Dynamically adjusted τ based on observed stop-wave propagation speeds
Diffusion Term	Constant diffusion coefficient ($D = 0.01$)	Spatially adaptive diffusion ($D(x,t) = f_\theta(x,t)$) for heterogeneous road conditions
Optimization	Adam optimizer with fixed learning rate	Adam optimizer with adaptive learning rate; dual constraints for PDE residuals and data fidelity
Performance	Improved shockwave gradients and transitions in synthetic scenarios	Captured stop-and-go wave speeds within 7%of ground truth; better alignment with observed spacetime diagrams
Visualization	Sharp gradients and defined shockwaves in synthetic data	Improved alignment with real-world velocity fields, especially near shockwave incident locations

To rigorously evaluate the effectiveness of the proposed PINN-TSE model, we performed a comprehensive comparison against a pure data-driven neural network baseline. Multiple performance metrics were employed: mean absolute error (MAE), root mean squared error (RMSE), shockwave speed estimation accuracy, and computational efficiency. These collectively assess predictive accuracy, physical fidelity, and potential for deployment in real-time systems.

To ensure statistical robustness, each experiment was repeated over 10 independent runs with randomized weight initializations and data splits. The reported results represent the mean $\pm$ standard deviation for each metric, as summarized in Table 4. For traffic **density**, the PINN-TSE model achieved a mean MAE of 2.4 ± 0.3 vehicles per mile (vpm), compared to 6.0 ± 0.5 vpm for the baseline. For **velocity**, the MAE improved from 15.0 ± 1.2 mph to 3.98 ± 0.4 mph. RMSE values exhibited similar improvements. Importantly, the shockwave speed estimation error dropped from 25% to 8%, confirming the benefit of embedding physics-informed constraints.

TABLE 4. Performance metrics over 10 runs (Mean ± Std).

Metric	Pure Neural Network	PINN-TSE Model	Improvement
MAE (Density)	6.0 ± 0.5 vpm	2.4 ± 0.3 vpm	60%
MAE (Velocity)	15.0 ± 1.2 mph	3.98 ± 0.4 mph	73%
RMSE (Density)	0.30 ± 0.02	0.15 ± 0.01	50%
RMSE (Velocity)	0.35 ± 0.03	0.17 ± 0.02	51%
Shockwave Speed Error	$25\% \pm 3.1$	$8\% \pm 1.6$	68%
Training Runtime (s)	1.75 ± 0.02	1.85 ± 0.03	–5.7%

Both the baseline and PINN-TSE models share the same underlying MLP architecture. The key distinction lies in the training process: the PINN-TSE model incorporates physics-based constraints via partial differential equation (PDE) residuals, which act as a regularization term. This introduces a modest computational overhead due to additional gradient computations required to enforce the physics constraints.

As shown in Table 4, the average per-batch training runtime for the PINN-TSE model was measured at 1.85 ± 0.03 seconds, compared to 1.75 ± 0.02 seconds for the baseline model. These runtime differences closely align with pre-experimental expectations based on GPU capability and model configuration (MLP with 256 hidden units, 4 layers, batch size $\sim$5000, on RTX A5000).

Importantly, inference speed remains unaffected, as both models rely solely on a forward pass during deployment without PDE-related computations. Both architectures yield approximately 270K trainable parameters and $\sim$30 MFLOPs per forward pass. The increased training cost observed in the PINN-TSE model arises solely from additional backpropagation steps through the physics-informed loss, rather than from architectural complexity.

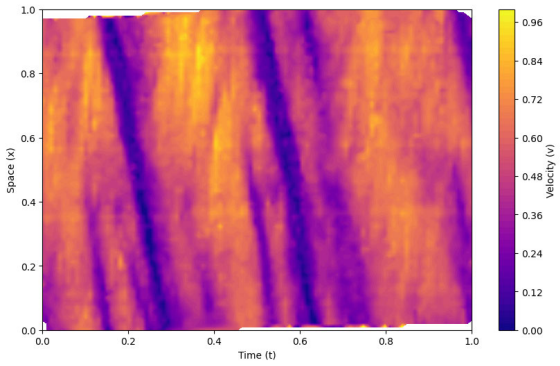
Qualitative comparisons in Figure 10 further highlight the improvements. The pure neural network yields over-smoothed profiles and fails to capture congestion fronts (Figures c–d). In contrast, the PINN-TSE model (Figures e–f) preserves sharp traffic features and reflects realistic wave propagation patterns observed in the ground truth (Figures a–b).

This study focuses primarily on evaluating the impact of embedding physical constraints in data-driven models. Exhaustive benchmarking against broader ML architectures (e.g., LSTM, Transformer) is beyond the current scope but remains a promising direction for future work.

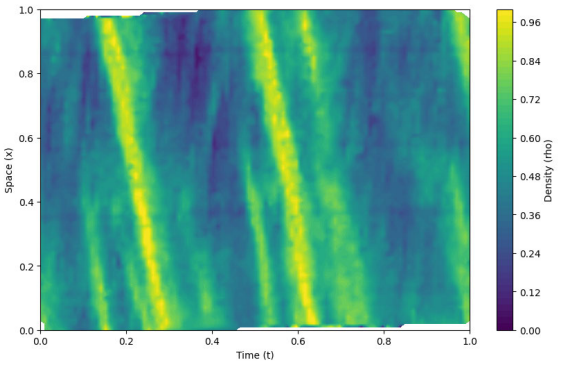
3) ABLATION AND SENSITIVITY ANALYSIS

Table 5 provides a comprehensive view of how various modeling assumptions, architectural decisions, and training

Ground Truth: Observed traffic state from NGSIM dataset

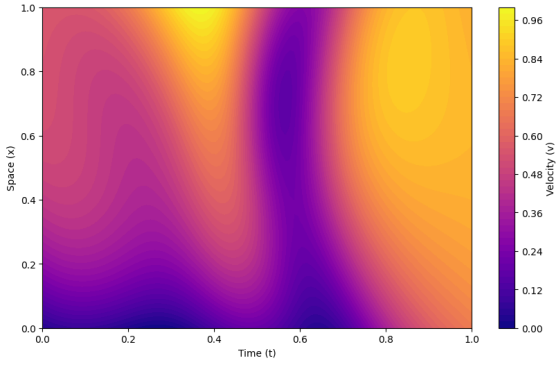


Speed (a): Reference traffic speed profile.

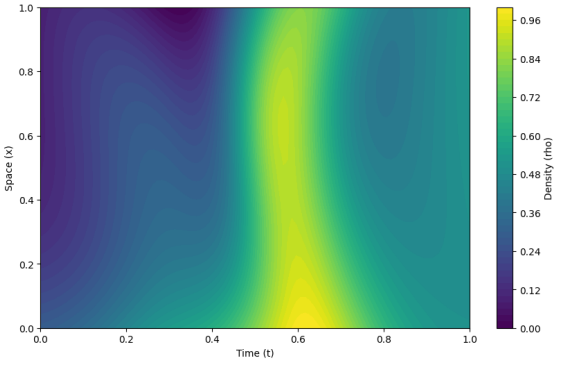


Density (b): Reference traffic density profile.

Pure Neural Network Model: Trained on shockwave point data

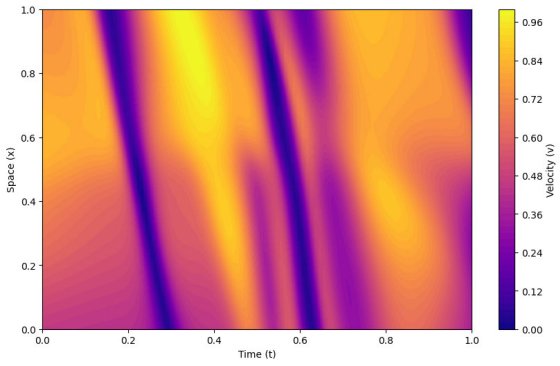


Speed (c): Over-smoothed flow; fails to capture sharp transitions.

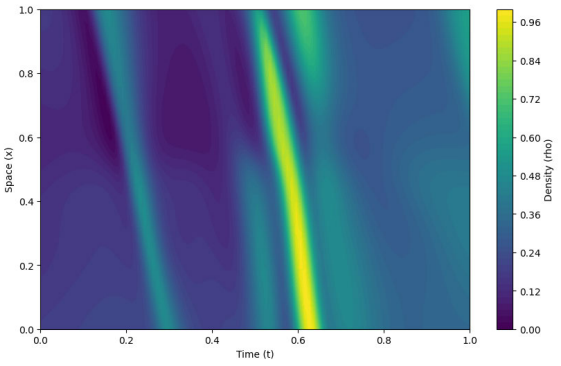


Density (d): Estimation with limited resolution in congested regions.

PINN-TSE Model: Physics-informed learning using shockwave-supervised loss



Speed (e): Captures wave propagation and dynamic transitions.



Density (f): Enhanced structure and congestion front clarity.

FIGURE 10. **Comparison of traffic state estimation methods using NGSIM data. All panels show normalized spatio-temporal traffic speed and density.**

hyperparameters affect the model's predictive accuracy (velocity MAE). Among modeling choices, linear pressure terms and dynamically learned reaction time and diffusion coefficients yielded the best alignment with real-world traffic behavior. Architecturally, a network with 256 hidden units and 3–4 layers performed best. In terms of training, a learning rate of 1×10^{-5}, Adam optimizer, and physics weight $\lambda_{\text{physics}} = 1.0$ led to optimal convergence. Notably, removing physics constraints entirely increased MAE by over 2.5 mph, reaffirming the importance of embedding traffic theory into the training process.

4) DEPLOYMENT CONSIDERATIONS

The trained PINN-TSE model has been successfully integrated into a traffic assistance chatbot application, providing real-time traffic estimates such as speed and density from sparse sensor data. Using its robustness to data noise and its ability to enforce physical constraints, the model is ideally

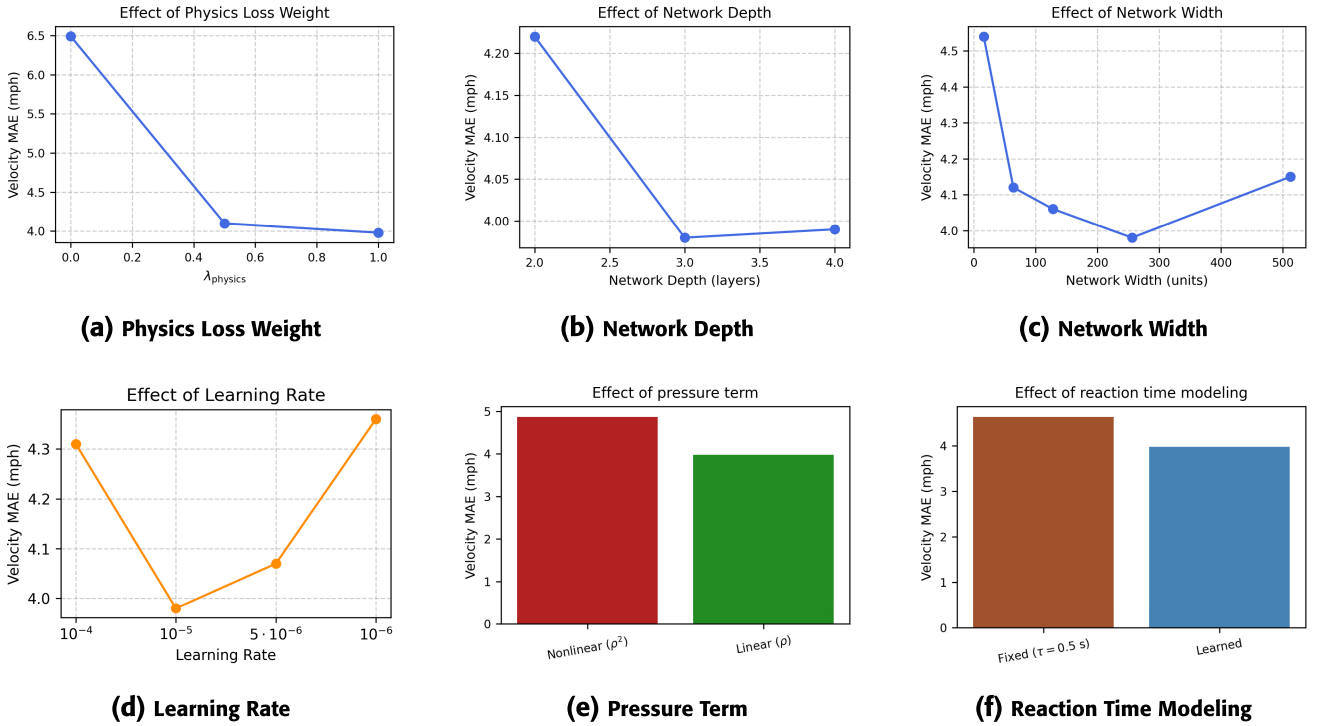


FIGURE 11. Ablation study on modeling and training design choices. Each subplot shows the sensitivity of velocity MAE to:(a) loss weighting for physics constraints,(b) network depth (number of hidden layers),(c) network width (hidden units per layer),(d) learning rate,(e) pressure term formulation, and(f) modeling of driver reaction time. The best-performing configuration in each category is chosen in the final model.

suited for deployment in dynamic, real-time systems such as smart city infrastructure and connected vehicle ecosystems, where reliable traffic predictions are essential.

D. SYSTEM-USER TESTING AND UX INSIGHTS

The traffic information app integrates user query understanding, the PINN-TSE traffic predictor, and a prompt-engineered language model (GPT-3.5-turbo) to provide real-time, conversational traffic insights. To evaluate its functionality, the system was tested with three categories of user queries: full spatio-temporal queries, partial queries (specifying either time or location), and general queries lacking specific context. These evaluations aimed to assess both the system's ability to interpret varied user input and the effectiveness of its user-facing design. These two aspects are discussed in the following subsections: first, the system's robustness in handling diverse query types; second, the rationale behind its conversational user interface.

1) HANDLING OF DIVERSE USER QUERY TYPES

For full spatio-temporal queries, such as **What is the traffic at 3:07 pm and 321 feet?**, the system successfully processes the precise input, delivering a clear and concise response: **At 3:07 PM, the traffic density at 321 feet along I-80 is 52.34 vehicles per mile per lane, and the speed is 38.65 miles per hour. This indicates that the traffic flow is uncongested at this location and time.** This showcases the app's ability to provide localized and precise traffic predictions, which are critical for users seeking specific information about their route.

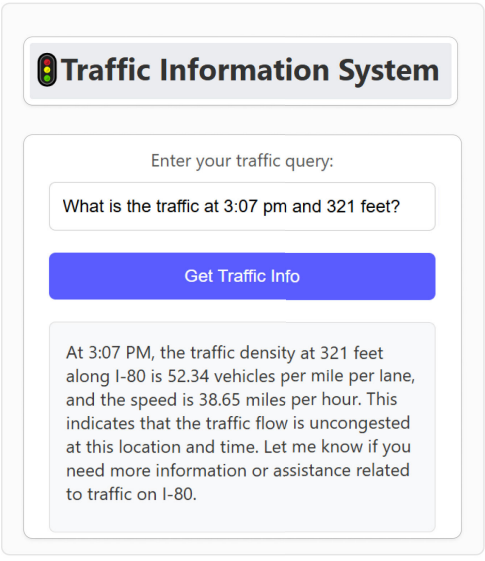


FIGURE 12. System Response: Specific time and location input.

In the case of partial queries, such as **What is the traffic at 3:00?**, the system provides a comprehensive overview of traffic conditions across the spatial domain. The response

TABLE 5. Ablation Study: Impact of modeling, architecture, and training parameters on velocity MAE.

Ablation Category	Setting	Velocity MAE (mph)	Δ MAE vs Best	Comment
Pressure Term	Nonlinear (ρ^2)	4.87	+0.89	Higher error near dense traffic
	Linear (ρ)	**3.98**	—	Best setting; simpler dynamics
Reaction Time τ	Fixed ($\tau = 0.5$ s)	4.63	+0.65	Less adaptive to stop-and-go waves
	Learned	**3.98**	—	Best; dynamically adjusts to flow regimes
Diffusion Term	Constant ($D = 0.01$)	4.79	+0.81	Over-smoothing in localized shockwaves
	Learned function $D(x,t)$	**3.98**	—	Best; captures heterogeneity
Network Width	16	4.54	+0.56	Underfit
	64	4.12	+0.14	Adequate
	128	4.06	+0.08	Slight gain
	256	**3.98**	—	Best setting
	512	4.15	+0.17	Diminishing returns
Network Depth	2 layers	4.22	+0.24	Not expressive enough
	3 layers	**3.98**	—	Best balance
	4 layers	**3.99**	+0.01	Nearly identical, slightly slower
Learning Rate	1e-4	4.31	+0.33	Converges faster, but overshoots
	1e-5	**3.98**	—	Best balance
	5e-6	4.07	+0.09	Slower convergence
	1e-6	4.36	+0.38	Too slow; prone to stalling
Optimizer	SGD (momentum 0.9)	4.66	+0.68	Poor convergence
	Adam	**3.98**	—	Best; stable and adaptive
Loss Weights	$\lambda_{\text{physics}} = 0.0$	6.49	+2.51	No physical constraint
	$\lambda_{\text{physics}} = 0.5$	4.10	+0.12	Balanced guidance
	$\lambda_{physics} = 1.0$	**3.98**	—	Best; physics-consistent

Note: Reaction time τ and diffusion $D(x,t)$ are modeled as learnable parameters during training.

highlights the variability in traffic conditions at the specified time: **At 3:00 PM on I-80 highway, the traffic density varies along the road: Minimum Density: 38.79 vehicles per mile per lane at 1575.00 feet; Maximum Density: 46.12 vehicles per mile per lane at 2100.00 feet. The traffic speeds also vary: Minimum Speed: 28.85 mph at 0.00 feet; Maximum Speed: 46.35 mph at 1575.00 feet. Overall, the traffic flow at 3:00 PM on I-80 is moderate with some fluctuations in density and speed along the highway.** This demonstrates the app's ability to generalize and provide meaningful insights even when spatial details are missing, making it useful for users planning their travel at a specific time.

For location-only queries, such as **What is the traffic at 152 feet?**, the system analyzes traffic conditions over the temporal domain, identifying periods of congestion and uncongested flow: **At 152 feet, the traffic density varies over time: Minimum Density: 45.19 vehicles per mile per lane at 3:03 PM; Maximum Density: 71.53 vehicles per mile per lane at 3:17 PM. Please note that if the traffic density is below 70 vehicles per mile per lane, the flow is uncongested. Otherwise, it is congested.** This highlights the app's ability to adapt to incomplete queries and provide useful insights by leveraging available information, which is particularly valuable for users monitoring traffic at a specific location over time.

For general queries, such as **What is the test road's traffic condition, advise?**, the app provides a high-level summary of traffic conditions across the spatio-temporal domain:

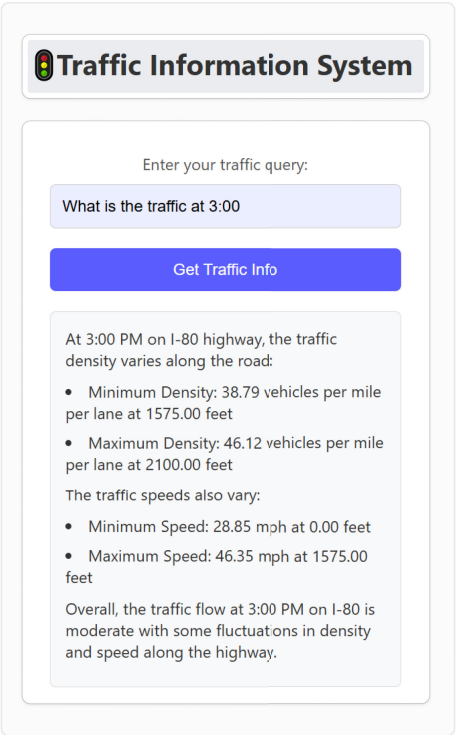


FIGURE 13. System Response: Time specific input, generalization over distance.

Based on the traffic predictions on the test road (I-80 highway) from 3:00 PM to 3:15 PM: Minimum Traffic

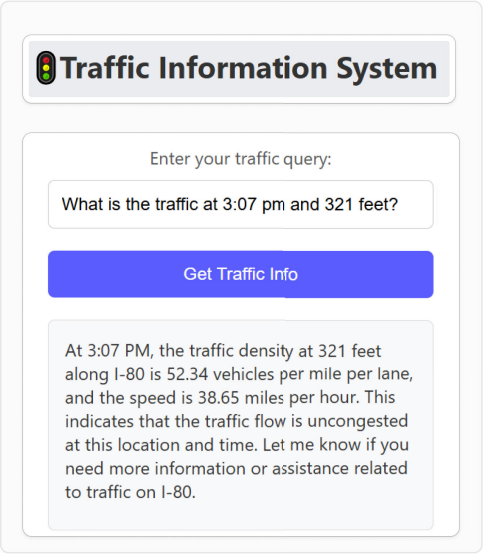


FIGURE 14. System Response: Location specific input, generalization over time.

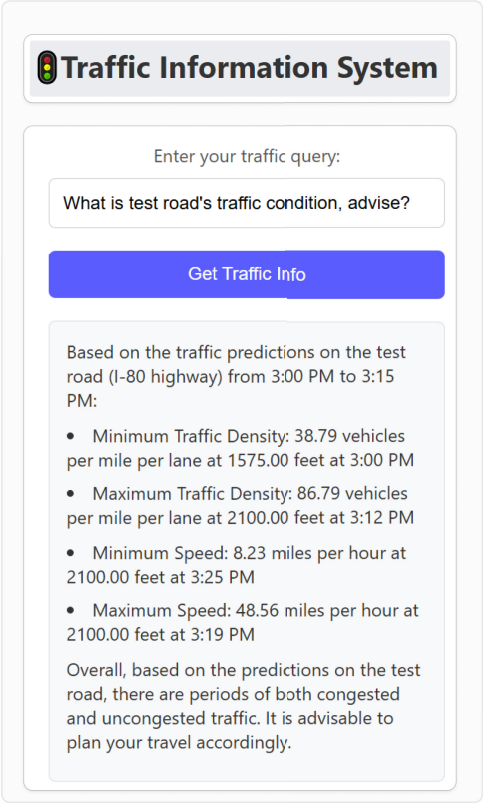


FIGURE 15. System Response: General input, generalization over time and location.

Density: 38.79 vehicles per mile per lane at 1575.00 feet at 3:00 PM; Maximum Traffic Density: 86.79 vehicles per mile per lane at 2100.00 feet at 3:12 PM; Minimum Speed: 8.23 miles per hour at 2100.00 feet at 3:25 PM; Maximum Speed: 48.56 miles per hour at 2100.00 feet at 3:19 PM. Overall, based on the predictions on the test road, there are periods of both congested and uncongested traffic. It is advisable to plan your travel accordingly. This demonstrates the app's ability to handle broad queries and provide actionable insights, even when specific details are lacking.

The system's precision in handling localized queries, generalization for partial queries, temporal analysis for location queries, and high-level summaries for general queries collectively make it a versatile and reliable tool for users seeking traffic information. Its ability to adapt to varying levels of input specificity ensures that users receive tailored and actionable insights, whether they are planning their travel in advance or monitoring real-time traffic conditions.

The conversational interface powered by an LLM (GPT-3.5-turbo) provides an intuitive way for users to interact with the traffic information system. This design is particularly effective for users on the go, such as drivers, who may need to query traffic conditions while keeping their focus on the road. The simple and elegant design ensures that users can easily ask specific or general questions and receive clear, detailed answers without need to navigate through complex menus or interfaces. Additionally, this LLM-based interface is flexible and can be easily adapted for hands-free and voice-based interaction in future iterations, further enhancing accessibility and usability in real-world, mobile environments. This represents a significant step toward a more interactive and dynamic traffic assistance system, capable of handling rich queries with ease.

2) USER INTERFACE DESIGN RATIONALE

The traffic information app adopts a minimal, text-based conversational interface powered by an LLM (GPT-3.5-turbo), designed to support intuitive interaction and dynamic traffic prediction. Users can pose queries ranging from brief questions to detailed spatio-temporal requests, and the system returns informative summaries based on current and predicted traffic conditions. Rather than overwhelming users with raw numerical outputs, the language model generalizes traffic trends into meaningful statistical descriptions, highlighting congestion levels, variability in flow, and significant deviations. This approach aims to simplify the complex model outputs into user-friendly narratives.

A key feature of the interface is its support for iterative, context-aware interaction. Users can request follow-up information or refine their queries to focus on specific time intervals or locations. If the model initially misinterprets a query, users can provide corrective feedback, which the system incorporates by adjusting the underlying inputs used for traffic inference. This feedback loop improves response accuracy and aligns predictions more closely with user intent.

Although the current implementation uses text-based input and output, the interface is designed with future

extension to hands-free, voice-based interaction in mind. Such functionality would be particularly advantageous for drivers and mobile users, enabling them to access real-time and predictive traffic insights without manual input. This conversational interface grounded in predictive modeling and guided by user feedback offers a scalable and accessible framework for delivering actionable traffic information in intelligent transportation systems.

V. CONCLUSION

Traffic modeling remains a central challenge in modern transportation systems, with far-reaching implications for safety, efficiency, and sustainability. Traditional modeling approaches—often reliant on sparse sensor data and rigid mathematical formulations—frequently fall short in capturing complex traffic dynamics or providing interpretable, user-accessible insights. This paper addressed these limitations by introducing PINN-TSE, a hybrid physics-informed neural network framework that integrates the Aw-Rascle model with advanced machine learning techniques and natural language processing.

The proposed system was designed with three core objectives: (i) enhancing physical fidelity by embedding traffic flow theory directly into the learning process, (ii) improving generalization and performance through a carefully balanced loss function informed by observed shockwave data, and (iii) enabling human-centered access to traffic predictions via large language models (LLMs). These design choices allowed PINN-TSE to significantly improve traffic state estimation and user interaction capabilities across a range of real-world conditions.

Empirical evaluation on the US-101 dataset demonstrated strong performance. Using only 8–10% of the original trajectory data, PINN-TSE achieved a mean absolute error (MAE) of 2.4 vehicles per mile (vpm) for traffic density and 3.98 mph for velocity, representing improvements of 60% and 73%, respectively, over a purely data-driven neural network. The shockwave speed estimation error was reduced to 8%, yielding a 68% improvement, which is critical for accurate modeling of stop-and-go wave propagation. These results confirm that incorporating physics constraints into machine learning models leads to more robust and reliable traffic predictions, even in sparse-data regimes.

The complete system implementation extends beyond accurate modeling to support practical deployment. First, the hybrid PINN-TSE architecture delivers physically consistent predictions grounded in established traffic flow theory. Second, the integration of LLMs enables intuitive natural-language interaction, broadening accessibility for both end users and domain experts. Third, the framework's modular design allows for seamless integration with smart infrastructure, connected vehicle platforms, and urban planning tools. This adaptability positions PINN-TSE as a valuable asset for commuters, traffic authorities, and city planners alike.

Looking ahead, future extensions could involve scaling to network-level applications, incorporating contextual data such as weather conditions or incidents, and exploring adaptive PDE formulations to better capture complex or non-standard traffic patterns. These advancements would further enhance the applicability of physics-informed learning in intelligent transportation systems, contributing to safer, more efficient, and more responsive traffic management solutions.

REFERENCES

[1] *The Roadway Safety Problem*, U.S. Department of Transportation, Washington, DC, USA, Oct. 2023.
[2] *Transport: Overview*, World Bank, Washington, DC, USA, Sep. 2023.
[3] L. Hejtmánek, I. Oravcová, J. Motýl, J. Horáček, and I. Fajnerová, "Spatial knowledge impairment after GPS guided navigation: Eye-tracking study in a virtual town," *Int. J. Hum.-Comput. Stud.*, vol. 116, pp. 15–24, Aug. 2018.
[4] L. Yang, X. Meng, and G. E. Karniadakis, "B-PINNs: Bayesian physics-informed neural networks for forward and inverse PDE problems with noisy data," *J. Comput. Phys.*, vol. 425, Jan. 2021, Art. no. 109913.
[5] Office of Operations, Federal Highway Administration. (2023). *Active Traffic Management (ATM)*. Accessed: Oct. 23, 2024. [Online]. Available: https://ops.fhwa.dot.gov/atdm/approaches/atm.htm
[6] T. Syum Gebre, L. Beni, E. Tsehaye Wasehun, and F. Elikem Dorbu, "AI-integrated traffic information system: A synergistic approach of physics informed neural network and GPT-4 for traffic estimation and real-time assistance," *IEEE Access*, vol. 12, pp. 65869–65882, 2024.
[7] T. S. Gebre and L. Hashemi-Beni, "An integrated framework of GPT-4 and PINN for dynamic traffic estimation and support," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2024, pp. 5457–5460.
[8] H. L. Khoo and K. Asitha, "User requirements and route choice response to smart phone traffic applications (apps)," *Travel Behav. Soc.*, vol. 3, pp. 59–70, Jan. 2016.
[9] A. D. Patire, M. Wright, B. Prodhomme, and A. M. Bayen, "How much GPS data do we need?" *Transp. Res. C, Emerg. Technol.*, vol. 58, pp. 325–342, Sep. 2015.
[10] C. Chen, T. H. Luan, X. Guan, N. Lu, and Y. Liu, "Connected vehicular transportation: Data analytics and traffic-dependent networking," *IEEE Veh. Technol. Mag.*, vol. 12, no. 3, pp. 42–54, Sep. 2017.
[11] J. Cao, Z. Fang, G. Qu, H. Sun, and D. Zhang, "An accurate traffic classification model based on support vector machines," *Int. J. Netw. Manage.*, vol. 27, no. 1, p. e1962, Jan. 2017.
[12] C. Deng, F. Wang, H. Shi, and G. Tan, "Real-time freeway traffic state estimation based on cluster analysis and multiclass support vector machine," in *Proc. Int. Workshop Intell. Syst. Appl.*, May 2009, pp. 1–4.
[13] A. B. Habtie, A. Abraham, and D. Midekso, "Artificial neural network based real-time urban road traffic state estimation framework," in *Computational Intelligence in Wireless Sensor Networks* (Studies in Computational Intelligence), vol. 676. Cham, Switzerland: Springer, 2017, pp. 73–97, doi: 10.1007/978-3-319-47715-2_4.
[14] Z. Abbas, A. Al-Shishtawy, S. Girdzijauskas, and V. Vlassov, "Short-term traffic prediction using long short-term memory neural networks," in *Proc. IEEE Int. Congr. Big Data (BigData Congress)*, Jul. 2018, pp. 57–65.
[15] B. N. Passow, D. Elizondo, F. Chiclana, S. Witheridge, and E. Goodyer, "Adapting traffic simulation for traffic management: A neural network approach," in *Proc. 16th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2013, pp. 1402–1407.
[16] G. Dhingra, S. Supreeth, K. Neha, R. Amruthashree, and D. Eshitha, "Traffic management using convolution neural network," *Int. J. Eng. Adv. Technol.*, vol. 8, no. 5S, pp. 146–149, 2019.
[17] D. Xu, Y. Wang, P. Peng, S. Beilun, Z. Deng, and H. Guo, "Real-time road traffic state prediction based on kernel-KNN," *Transportmetrica A: Transp. Sci.*, vol. 16, no. 1, pp. 104–118, Dec. 2020.
[18] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal, "Explaining explanations: An overview of interpretability of machine learning," in *Proc. IEEE 5th Int. Conf. Data Sci. Adv. Anal. (DSAA)*, Oct. 2018, pp. 80–89.
[19] A. Aw and M. Rascle, "Resurrection of 'second order' models of traffic flow," *SIAM J. Appl. Math.*, vol. 60, no. 3, pp. 916–938, 2000.
[20] Y. Wang and M. Papageorgiou, "Real-time freeway traffic state estimation based on extended Kalman filter: A general approach," *Transp. Res. B, Methodol.*, vol. 39, no. 2, pp. 141–167, Feb. 2005.

[21] Y. Yuan, J. W. C. van Lint, R. E. Wilson, F. van Wageningen-Kessels, and S. P. Hoogendoorn, "Real-time Lagrangian traffic state estimator for freeways," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 1, pp. 59–70, Mar. 2012.

[22] T. Seo and A. M. Bayen, "Traffic state estimation method with efficient data fusion based on the Aw-Rascle-Zhang model," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2017, pp. 1–6.

[23] S. C. Vishnoi, S. A. Nugroho, A. F. Taha, and C. G. Claudel, "Traffic state estimation for connected vehicles using the second-order Aw–Rascle–Zhang traffic model," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 11, pp. 16719–16733, Nov. 2024.

[24] H. Yu and M. Krstic, "Traffic congestion control for Aw–Rascle–Zhang model," *Automatica*, vol. 100, pp. 38–51, Feb. 2019.

[25] A. J. Huang and S. Agarwal, "Physics-informed deep learning for traffic state estimation: Illustrations with LWR and CTM models," *IEEE Open J. Intell. Transp. Syst.*, vol. 3, pp. 503–518, 2022.

[26] R. Haberman, *Mathematical Models: Mechanical Vibrations, Population Dynamics, and Traffic Flow*. Philadelphia, PA, USA: SIAM, 1998.

[27] M. Talal, K. N. Ramli, A. Zaidan, B. Zaidan, and F. Jumaa, "Review on car-following sensor based and data-generation mapping for safety and traffic management and road map toward ITS," *Veh. Commun.*, vol. 25, Oct. 2020, Art. no. 100280.

[28] M. Raissi, P. Perdikaris, and G. Em Karniadakis, "Physics informed deep learning (Part I): Data-driven solutions of nonlinear partial differential equations," 2017, *arXiv:1711.10561*.

[29] M. Raissi, P. Perdikaris, and G. Em Karniadakis, "Physics informed deep learning (Part II): Data-driven discovery of nonlinear partial differential equations," 2017, *arXiv:1711.10566*.

[30] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *J. Comput. Phys.*, vol. 378, pp. 686–707, Feb. 2019.

[31] M. Usama, R. Ma, J. Hart, and M. Wojcik, "Physics-informed neural networks (PINNs)-based traffic state estimation: An application to traffic network," *Algorithms*, vol. 15, no. 12, p. 447, Nov. 2022.

[32] A. J. Huang and S. Agarwal, "Physics informed deep learning for traffic state estimation," in *Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst. (ITSC)*, Sep. 2020, pp. 1–6.

[33] J. Lu, C. Li, X. B. Wu, and X. S. Zhou, "Physics-informed neural networks for integrated traffic state and queue profile estimation: A differentiable programming approach on layered computational graphs," *Transp. Res. C, Emerg. Technol.*, vol. 153, Aug. 2023, Art. no. 104224.

[34] X. Di, R. Shi, Z. Mo, and Y. Fu, "Physics-informed deep learning for traffic state estimation: A survey and the outlook," *Algorithms*, vol. 16, no. 6, p. 305, Jun. 2023.

[35] Z. Mao, A. D. Jagtap, and G. E. Karniadakis, "Physics-informed neural networks for high-speed flows," *Comput. Methods Appl. Mech. Eng.*, vol. 360, Mar. 2020, Art. no. 112789.

[36] M. Yin, X. Zheng, J. D. Humphrey, and G. E. Karniadakis, "Non-invasive inference of thrombus material properties with physics-informed neural networks," *Comput. Methods Appl. Mech. Eng.*, vol. 375, Mar. 2021, Art. no. 113603.

[37] L. Kohnke, B. L. Moorhouse, and D. Zou, "ChatGPT for language teaching and learning," *RELC J.*, vol. 54, no. 2, pp. 537–550, Apr. 2023.

[38] X. Lin, "Exploring the role of ChatGPT as a facilitator for motivating self-directed learning among adult learners," *Adult Learn.*, vol. 35, no. 3, pp. 156–166, Aug. 2024.

[39] J. Jeon, S. Lee, and H. Choe, "Beyond ChatGPT: A conceptual framework and systematic review of speech-recognition chatbots for language learning," *Comput. Educ.*, vol. 206, Dec. 2023, Art. no. 104898.

[40] E. Kasneci et al., "ChatGPT for good? On opportunities and challenges of large language models for education," *Learn. Individual Differences*, vol. 103, Mar. 2023, Art. no. 102274.

[41] B. Wang, Z. Cai, M. Monjurul Karim, C. Liu, and Y. Wang, "Traffic performance GPT (TP-GPT): Real-time data informed intelligent ChatBot for transportation surveillance and management," 2024, *arXiv:2405.03076*.

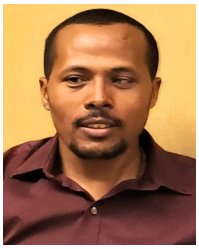
[42] Z.-H. Ou, S.-Q. Dai, P. Zhang, and L.-Y. Dong, "Nonlinear analysis in the Aw–Rascle anticipation model of traffic flow," *SIAM J. Appl. Math.*, vol. 67, no. 3, pp. 605–618, Jan. 2007.

[43] S. Moutari and M. Rascle, "A hybrid Lagrangian model based on the Aw–Rascle traffic flow model," *SIAM J. Appl. Math.*, vol. 68, no. 2, pp. 413–436, Jan. 2007.

[44] N. H. Gartner, C. J. Messer, and A. Rathi, *Traffic Flow Theory: A State-of-the-Art Report: Revised Monograph on Traffic Flow Theory*. McLean, VA, USA: Turner-Fairbank Highway Research Center, 2002.

TEWODROS SYUM GEBRE received the B.S. degree in hydraulic engineering from Arba Minch University, Ethiopia, in 2009, the M.S. degree in civil (road and transport) engineering from Addis Ababa University, Ethiopia, in 2014, and the Ph.D. degree in applied science and technology from North Carolina Agricultural and Technical State University, USA, in 2025.

He is currently a Postdoctoral Researcher with North Carolina Agricultural and Technical State University, in AI and ML AI applications for smart cities and Infrastructure Resilience. He was with Microsoft-funded project developing AI-based traffic monitoring systems using transformer models and UAV imagery. His expertise spans developing machine learning models for automating traffic management, image and both structure and unstructured data analysis, and system development. He is a Former Member of the NC-CAV Center, supported by North Carolina Department of Transportation. He received the G. Herbert Stout Award for Best Student Paper from North Carolina GIS (NCGIS) Conference, the Excellence Scholarship, from 2021 to 2024, and the Microsoft's AFMR Grant, from 2023 to 2024.

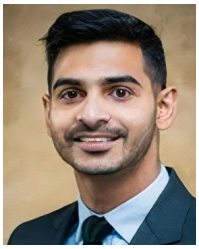


SIMACHEW ENDALE ASHEBIR received the B.S. degree in physics from Haramaya University, Ethiopia, in 2006, the M.S. degree in physics from Addis Ababa University, Ethiopia, in 2010, and the M.S. degree in mathematical sciences from Stellenbosch University, South Africa, in 2013. He is currently pursuing the Ph.D. degree in data science and analytics with North Carolina Agricultural and Technical State University, Greensboro, NC, USA. From January 2020 to May 2021, he was a Research-Based Graduate Fellow in computational data science and food security with the Department of Industrial and Systems Engineering, University of North Carolina, USA. Since 2021, he was a Research Assistant with the Department of Mathematics and Statistics. In the Summer 2023, he participated as a fellow in the Energy Data Analytics Ph.D. Student Fellows Program..

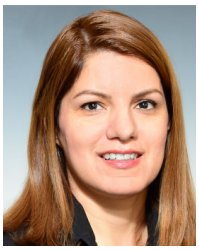
JEFFREY BLAY received the B.A. degree in geography and resource development from the University of Ghana, Ghana, in 2019, and the M.Sc. degree in environmental science (urban geospatial data science) from the Yale School of the Environment, USA. He is currently pursuing the Ph.D. degree in applied science and technology with North Carolina Agricultural and Technical State University, working on a NASA Funded Project on applying computer vision algorithms and advanced geospatial techniques to predict floodwater depth and its impact on settlement areas, from post flood aerial imagery and LiDAR data. His expertise covers applying geospatial data engineering techniques to extracts insights from structured and unstructured data, as well as applying deep learning algorithms to analyze remotely sensed imagery including, drone imagery, aerial photography, satellite imagery and web images. He is a member of American Society for Photogrammetry and Remote Sensing (ASPRS), and the Secretary of North Carolina Agricultural and Technical State University ASPRS Student Chapter. He won the LiDAR Leadership Award from the 2024 Annual ASPRS and the Geo Week Conference in Denver, USA.

MATILDA ANOKYE received the B.S. degree in geomatic engineering from the University of Mines and Technology, Ghana, in 2018, and the M.S. degree in urban forestry and natural resources from Southern University A&M College, Baton Rouge, LA, in 2022. She is currently pursuing the Ph.D. degree in applied science and technology with North Carolina Agricultural and Technical State University, Greensboro, NC, working on NASA and NOAA-funded project that utilizes UAV, LiDAR, SAR, and optical imagery to model flood-induced inundation patterns and assess in wetlands and watersheds.

Her research interests and skills span the area of data fusion, GeoAI, hydrological modeling, LiDAR point cloud processing, spatial analytics, and the application of data science and AI approaches to analyze remote sensing data for environmental monitoring and geospatial data-driven analysis.



VENKTESH PANDEY received the Ph.D. degree from the Department of Civil, Architectural, and Environmental Engineering (CAEE), University of Texas at Austin. He was a Project Associate with Indian Institute of Science, Bangalore. He is currently an Assistant Professor with the Department of Civil, Architectural, and Environmental Engineering, North Carolina Agricultural and Technical State University (N.C. A&T). He is actively involved with various professional organizations such as the Transportation Research Board (TRB), the Institute of Transportation Engineers (ITE), and the Institute for Operations Research and Management Sciences (INFORMS). He also has broad interests in improving Engineering Education systems of the future. His research integrates intelligent transportation systems (ITS) and emerging mobility services in traffic operations, congestion pricing, and transportation planning models with a focus on sustainability. He was a recipient of the 2022-23 Junior Faculty Teaching Excellence Award from N.C. A&T.



LEILA HASHEMI-BENI (Member, IEEE) received the B.S. degree in civil-surveying engineering (geomatics) from the University of Isfahan, the M.S. degree in civil-surveying engineering (photogrammetry/remote sensing) from the University of Tehran, and the Ph.D. degree in geospatial information system from Laval University.

She is currently an Associate Professor and the Director of the NASA-Funded Institute for Harnessing Data Science for Environment Management, North Carolina Agricultural and Technical State University. She is currently a PI of the project "AI-based traffic monitoring systems using generative pre-trained transformer models and high-resolution UAV imagery" supported by Microsoft' Accelerating Foundation Models Research program. She is a PI/Co-PI on many projects supported by NASA, NSF, NOAA, Microsoft, North Carolina Collaboratory, and North Carolina DoT. Her research experience and interests span the areas of geospatial data science, UAV and satellite remote sensing, multi temporal and multisource data fusion and image classification, 3D data modeling, automatic matching and change detection between various datasets and developing GIS and remote sensing methodologies for environmental management. She has served as a proposal panelist and a reviewer for many US and international funding organizations. She has served as the chair/co-chair or as a scientific committee member for many national or international conferences/workshops. She is currently serving as the Co-Chair of LiDAR, Laser Altimetry and Sensor Integration Working Group, International Society of Photogrammetry and Remote Sensing (ISPRS).