Communications in Transportation Research 5 (2025) 100166

Contents lists available at ScienceDirect

Communications in Transportation Research

journal homepage: www.journals.elsevier.com/communications-in-transportation-research

Full Length Article

Physical enhanced residual learning (PERL) framework for vehicle trajectory prediction

Keke Long, Zihao Sheng, Haotian Shi *, Xiaopeng Li **, Sikai Chen, Soyoung Ahn

Department of Civil & Environmental Engineering, University of Wisconsin–Madison, Wisconsin, 53706, USA

ARTICLE INFO

Keywords:
Trajectory prediction
Residual
Car-following model
Neural network

ABSTRACT

While physics models for predicting system states can reveal fundamental insights owing to their parsimonious structure, they may not always yield the most accurate predictions, particularly for complex systems. As an alternative, neural network (NN) models usually yield more accurate predictions; however, they lack interpretable physical insights. To articulate the advantages of both physics and NN models while circumventing their limitations, this study proposes a physics-enhanced residual learning (PERL) framework that adjusts a physics model prediction with a corrective residual predicted from a residual learning NN model. The integration of the physics model preserves interpretability and tremendously reduces the amount of training data compared with pure NN models. We apply PERL to a vehicle trajectory prediction problem with real-world trajectory data of both a human-driven vehicle (HV) and an autonomous vehicle (AV), using an adapted Newell car-following model as the physics model and four kinds of neural networks (Gated Recurrent Unit (GRU), Convolution long short-term memory (CLSTM), Variational Autoencoder (VAE), and the Informer model) as the residual learning model. We compare this PERL model with pure physics models, NN models, and other physics-informed neural network (PINN) models. The results reveal that PERL yields the best prediction when the training data are small. The PERL model converges quickly during training. Moreover, compared with the NN and PINN models, the PERL model requires fewer parameters to achieve similar predictive performance. A sensitivity analysis revealed that the PERL model consistently outperforms the physics models, NN models and PINN models with different physics and residual learning models given a small training dataset. Among these, the PERL model based on CLSTM achieved the most accurate predictions.

1. Introduction

The prediction approach in recent decades has undergone a revolutionary change with the development of computer technology. Traditionally, researchers have used theoretical derivation combined with experimental verification to study natural phenomena and then make predictions on the basis of the theory they have learned. Physics models are constructed from fundamental laws and principles governing physical systems for system state prediction (Fig. 1a). As it intuitively reveals fundamental insights into the system's nature, a physics model's parsimonious structures may not always accurately describe complex nonlinearity, high dimensionality, or latent patterns, thus limiting its predictability.

The parsimony of physics models is sometimes compromised when only limited data and computational resources are available. The advent of data science has led to a prolific accumulation of data across various fields. Concurrently, neural network (NN) models, along with numerous data-driven machine learning methods, have emerged to enhance predictability, as NN models are capable of approximating any continuous function (Fig. 1b) (Hornik et al., 1989). NN models have been extensively utilized in domains such as medical forecasting (Gulshan et al., 2016), weather forecasting (Salman et al., 2015), and road traffic prediction (Bates et al., 2024; Xu et al., 2023; Zhong et al., 2023), the third being the focused domain of this study.

Despite the superior predictability of NN models, they suffer from several limitations in real-world applications, including a lack of interpretability, lack of robustness, and dependence on vast amounts of data (Karniadakis et al., 2021; Li et al., 2022). To overcome these shortcomings, the physics-informed neural network (PINN) model has emerged (Karniadakis et al., 2021; Zhang et al., 2024). To enhance interpretability

* Corresponding author.
** Corresponding author.
E-mail addresses: hshi84@wisc.edu (H. Shi), xli2485@wisc.edu (X. Li).

https://doi.org/10.1016/j.commtr.2025.100166
Received 19 October 2024; Received in revised form 4 December 2024; Accepted 6 December 2024
Available online 14 April 2025
2772-4247/© 2025 The Authors. Published by Elsevier Ltd on behalf of Tsinghua University Press. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

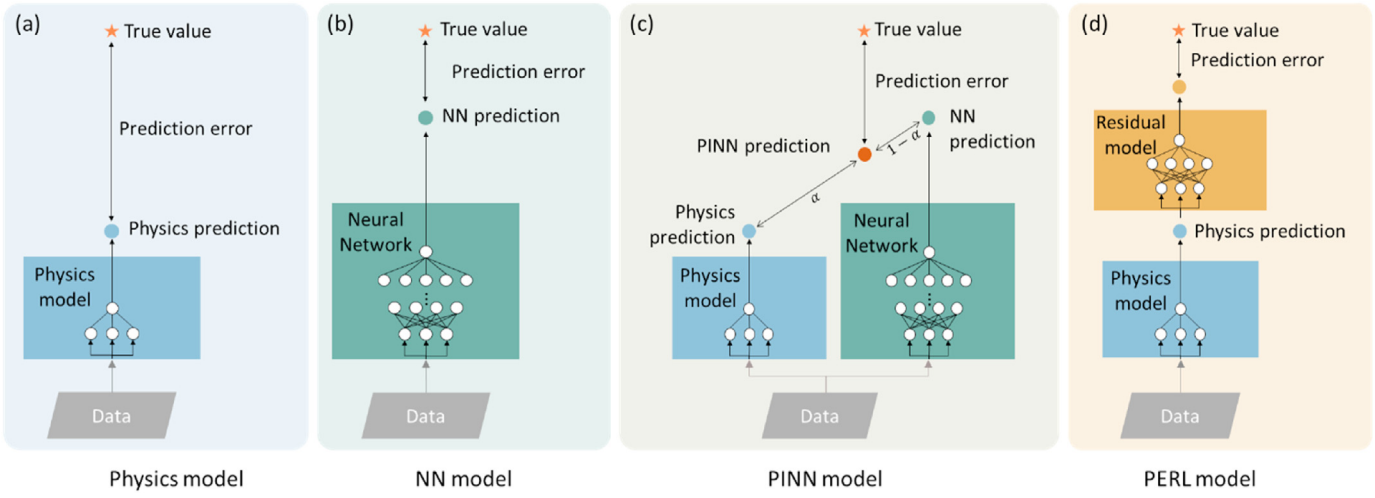


Fig. 1. General structure of the PERL model and three existing models: the physics model, NN model, and PINN model.

and ensure the stability of models beyond the domains of their training data, the PINN model encodes physics prior to the NN. The widely adopted PINN for road traffic prediction, such as vehicle trajectory prediction, integrates physics priors into the NN's loss function (Long et al., 2024b; Mo et al., 2021; Naing et al., 2022; Yao et al., 2023). This approach strikes a balance between the physics model and NN by modulating the weight of the deviation from the physics model prediction and that of the NN's loss function during training, as shown in Fig. 1c. Despite their advantages, deviation correction toward often less accurate physics model predictions may compromise the PINN's predictability. Another concern arises from unstable training due to the different scales and convergence rates between the NN loss and the physics model deviation. Thus, tuning the weights of the physics model within the loss function is proven to be challenging (Krishnapriyan et al., 2021; Wang et al., 2021, 2022). Furthermore, without sufficient data, PINNs may underperform since the efficacy of PINNs heavily leans toward the relatively inferior predictability of the physics model.

To exploit the advantages of all these models while circumventing their limitations, we propose a novel physics-enhanced residual learning (PERL) framework (Fig. 1d) (Long et al., 2024c). Instead of a compromise between the two models, PERL further improves the prediction of the physics model with a residual term learned from the NN model. The physics model component alone can yield a reasonably accurate (albeit not the best) prediction with only a parsimonious structure specified by a few parameters. Then, as opposed to a pure NN model starting from scratch, the residual learning model component only needs to improve an already reasonably accurate prediction to a higher accuracy comparable to the pure NN model, which likely requires much less training data and computational resources and thus only needs fewer parameters and less training data. Moreover, unlike the PINN, which may draw predictions toward relatively less accurate physics model predictions, PERL pushes the prediction away from the physics model prediction to the most accurate value that a data-driven model can ever reach. With this, PERL better integrates the merits of both physics and NN models to achieve the best and uncompromised predictability and the fastest convergence rate with the fewest parameters and inherited interpretability. While the concept of a residual term has been explored in various fields, such as physics systems modeling (Brunton et al., 2016), robotic manipulation tasks (Möllerstedt et al., 2024; Silver et al., 2018), and electromagnetic modeling (Shan et al., 2023), to the best of the authors' knowledge, residual learning has rarely been explored for system state prediction, which motivated us to develop the PERL model.

To demonstrate the effectiveness of the PERL model, we applied the PERL model to the vehicle trajectory prediction problem via a real-world dataset. Vehicle trajectory prediction in the context of road traffic is a procedure that uses the observed motions of road vehicles to predict their near-future trajectory attributes, such as speed and acceleration. It is necessary for an automated vehicle to predict the motion paths of its surrounding vehicles and ensure safety control. Furthermore, with the development of high-resolution traffic sensing technologies, vehicle trajectory prediction can be applied to predict future traffic states and potential traffic conflicts for safer, faster, and greener traffic operations. Road traffic dynamics are a representative example of complex systems with partially known physics. For example, some general physics rules in which the behavior of the preceding vehicle influences the following vehicle's behavior are already considered in widely recognized car-following models (Jiang et al., 2001; Treiber et al., 2000). In this application, we first obtain a calibrated physics model on the basis of the next-generation simulation (NGSIM) dataset (Dong et al., 2021). The residuals obtained from this physics model component are utilized to train a residual learning model component of the PERL model. The final prediction of the PERL model is the summation of the physics prediction and the learned residual. The performance of the PERL model is compared with that of the pure physics model, the pure NN model, and the PINN model. To further validate the generalizability of the PERL model, we conducted a sensitivity analysis to assess the performance of PERL when paired with alternative residual learning models via state-of-the-art NN models and physics car-following models. Additionally, we tested the model on autonomous vehicle (AV) trajectories to emphasize its robustness in diverse vehicular contexts.

This paper is organized as follows. Section 2 describes the prediction problem and the methodology employed in the study, including the PERL model framework and the baseline models. Section 3 presents the use case of vehicle trajectory prediction for the validation of the PERL model. Section 4 provides an example of the application of the PERL model, including details of the physics and residual learning components and the baseline models. Section 5 presents the results of the comparative analysis, including multistep prediction comparisons, convergence comparisons, and sensitivity analysis. Finally, Section 6 concludes the paper with a summary of findings and future research directions.

2. Methodology

This section describes the prediction problem and the PERL model. The input state (e.g., traffic states) space and the output response (e.g., set of longitudinal accelerations of the subject vehicle) space of a physics system are denoted by $\mathscr{S}$ and $\mathscr{Y}$, respectively, where M and N denote the numbers of dimensions of these two spaces and where $\mathbb{R}$ is the set of real numbers. Let $g(\cdot): \mathscr{S} \to \mathscr{Y}$ denote the ground-truth system that has response $g(s)$ for each input state $s \in \mathscr{S}$.

The prediction problem involves constructing a prediction function $f(\cdot|\theta): \mathscr{S} \to \mathscr{Y}$ such that $f(s|\theta)$ is the prediction of $g(s)$, $\forall s \in \mathscr{S}$, where $\theta \in \Theta$ represents a tunable parameter vector in a parameter space Θ. The predictability of function $f(\cdot|\theta)$ is evaluated by the prediction error between $f(s|\theta)$ and $g(s), \forall s \in \mathscr{S}$, e.g., the mean squared error (MSE). Solving a prediction problem usually involves finding the optimal θ to minimize the prediction error or maximize its predictability. The interpretability of $f(\cdot|\theta)$ is characterized by the ability of this function to reflect comprehensible physics rules and relationships, e.g., how elements in θ can be interpreted as certain physics rules and relationships.

2.1. PERL

Built upon the defined prediction problem, the PERL model can be described as follows. PERL prediction is the combination of a physics model component and a residual learning component, formulated as $f^{\text{PERL}}(\cdot|\theta^{\text{PERL}}) := f^{\text{Phy}}(S^{\text{Phy}}(\cdot)|\theta^{\text{Phy}}) + f^{\text{RL}}(\cdot|\theta^{\text{RL}})$. The components of this formulation are specified as follows. $f^{\text{Phy}}(S^{\text{Phy}}(s)|\theta^{\text{Phy}}): \mathscr{S}^{\text{Phy}} \to \mathscr{Y}^{\text{Phy}}$ is the physics model component that makes an initial prediction of an input state $s \in \mathscr{S}$. Here, $S^{\text{Phy}}(s) \in \mathscr{S}^{\text{Phy}}$ denotes the projection of $s \in \mathscr{S}$ to $\mathscr{S}^{\text{Phy}}$ as a subspace of $\mathscr{S}$. Note that, owing to parsimony, the input state to the physics model component f^{Phy} may not include all the dimensions in $\mathscr{S}$ and can be a projection to its subspace $\mathscr{S}^{\text{Phy}}$. θ^{Phy} is the parameter vector for f^{Phy} in parameter space Θ^{Phy}, which is often a low-dimensional space due to the parsimony of the physics model. $\mathscr{Y}^{\text{Phy}}$ is the range of component f^{Phy} with domain $\mathscr{S}^{\text{Phy}}$ and has the same cardinality as $\mathscr{Y}$ since f^{Phy} also predicts g. $f^{\text{RL}}(\cdot|\theta^{\text{RL}}): \mathscr{S} \to \mathscr{Y}^{\text{RL}} := \mathscr{Y} - \mathscr{Y}^{\text{Phy}}$ is the residual learning component such that $f^{\text{RL}}(s|\theta^{\text{RL}})$ predicts the residual from the physical model prediction to the ground truth, i.e., $g(s) - f^{\text{Phy}}(S^{\text{Phy}}(s)|\theta^{\text{Phy}}), \forall s \in \mathscr{S}$. θ^{RL} is the parameter vector for the residual learning component f^{RL} in parameter space Θ^{RL}. With this, the PERL model prediction $f^{\text{PERL}}(s|\theta^{\text{PERL}}) \in \mathscr{Y}$ is the summation of the physics model prediction $f^{\text{Phy}}(S^{\text{Phy}}(s)|\theta^{\text{Phy}})$ and the residual prediction $f^{\text{RL}}(s|\theta^{\text{RL}})$, $\forall s \in \mathscr{S}$, where the parameter vector θ^{PERL} is a simple concatenation of those for the physics and residual learning components, i.e., $\theta^{\text{PERL}} := [\theta^{\text{Phy}}, \theta^{\text{RL}}] \in \Theta^{\text{Phy}} \times \Theta^{\text{RL}}$.

In a real-world application, let $\mathscr{I} := \{1, \cdots, I\}$ denote the index set of observed samples, where $I \in \mathbb{R}^+$ is the total number of samples. $\mathscr{I}$ is randomly divided into three sets, i.e., $\mathscr{I}^{\text{Train}} \cup \mathscr{I}^{\text{Val}} \cup \mathscr{I}^{\text{Test}} = \mathscr{I}$, $\mathscr{I}^{\text{Train}} \cap \mathscr{I}^{\text{Val}} = \emptyset$; $\mathscr{I}^{\text{Train}} \cap \mathscr{I}^{\text{Test}} = \emptyset$;, and $\mathscr{I}^{\text{Val}} \cap \mathscr{I}^{\text{Test}} = \emptyset$. ω^{Train} and ω^{Val} are defined as the proportions of the total dataset allocated for training and validation purposes: $\left|\mathscr{I}^{\text{Train}}\right| = \left\lfloor \omega^{\text{Train}} \times \mathscr{I} \right\rfloor$ and $\left|\mathscr{I}^{\text{Val}}\right| = \left\lfloor \omega^{\text{Val}} \times \mathscr{I} \right\rfloor$, where $\lfloor \cdot \rfloor$ denotes the floor function. The testing set index $\mathscr{I}^{\text{Test}}$ consists of the remaining indices, which are calculated as $\left|\mathscr{I}^{\text{Test}}\right| = |\mathscr{I}| - \left|\mathscr{I}^{\text{Train}}\right| - \left|\mathscr{I}^{\text{Val}}\right|$. θ^{Phy} is obtained via calibration via a set of observed states $\mathscr{S}^{\text{Train}} := \{s_i \in \mathscr{S}\}_{i \in \mathscr{I}^{\text{Train}}}$:

$$\theta^{\text{Phy}} := \underset{\theta^{\text{Phy}} \in \Theta^{\text{Phy}}}{\arg\min} \sum_{s \in \mathscr{S}^{\text{Train}}} \left(f^{\text{Phy}}\left(S^{\text{Phy}}(s)\middle|\theta^{\text{Phy}}\right) - g(s)\right)^2 \tag{1}$$

where θ^{RL} is obtained by training with the training dataset:

$$\theta^{\text{RL}} = \underset{\theta^{\text{RL}} \in \Theta^{\text{RL}}}{\arg\min} \sum_{s \in \mathscr{S}^{\text{Train}}} \left(f^{\text{RL}}\left(s\middle|\theta^{\text{RL}}\right) - \left(g(s) - f^{\text{Phy}}\left(S^{\text{Phy}}(s)\middle|\theta^{\text{Phy}}\right)\right)\right)^2 \tag{2}$$

2.2. Baseline models

For comparative analysis, we also evaluate the performance of a pure neural network (NN) prediction model, a pure physics-based prediction model, and a physics-informed neural network (PINN) model. The model descriptions are as follows.

2.2.1. Physics model

A physics model (e.g., a car-following model) is denoted as $f^{\text{Phy}}(S^{\text{Phy}}(s)|\theta^{\text{Phy}}): \mathscr{S}^{\text{Phy}} \to \mathscr{Y}^{\text{Phy}}$. Here, $S^{\text{Phy}}(s) \in \mathscr{S}^{\text{Phy}}$ denotes the projection of $s \in \mathscr{S}$ to $\mathscr{S}^{\text{Phy}}$ as a subspace of $\mathscr{S}$. $\mathscr{Y}^{\text{Phy}}$ is the range of component f^{Phy} with domain $\mathscr{S}^{\text{Phy}}$ and has the same cardinality as $\mathscr{Y}$. θ^{Phy} is the parameter vector for f^{Phy} in a low-dimensional parameter space Θ^{Phy} and is determined through calibration as specified in Eq. (1).

2.2.2. NN model

An NN model is denoted as $f^{\text{NN}}(s|\theta^{\text{NN}}): \mathscr{S} \to \mathscr{Y}^{\text{NN}}$, which makes a prediction of an input state $s \in \mathscr{S}$. $\mathscr{Y}^{\text{NN}}$ is the range of component f^{NN} with domain $\mathscr{S}$ and has the same cardinality as $\mathscr{Y}$. θ^{NN} is the parameter vector for f^{NN} in parameter space Θ^{NN} and is obtained by training via the training dataset:

$$\theta^{\text{NN}} = \underset{\theta^{\text{NN}} \in \Theta^{\text{NN}}}{\arg\min} \sum_{s \in \mathscr{S}^{\text{Train}}} \left(f^{\text{NN}}\left(s\middle|\theta^{\text{NN}}\right) - g(s)\right)^2 \tag{3}$$

2.2.3. PINN model

A PINN model denoted as $f^{\text{PINN}}(s|\theta^{\text{PINN}}): \mathscr{S} \to \mathscr{Y}^{\text{PINN}}$ that predicts an input state $s \in \mathscr{S}$. $\mathscr{Y}^{\text{PINN}}$ is the range of component f^{PINN} with domain $\mathscr{S}$ and has the same cardinality as $\mathscr{Y}$. During training, the physics model $f^{\text{Phy}}(S^{\text{Phy}}(s)|\theta^{\text{Phy}})$ enhances the loss function of the NN component $f^{\text{NN}}(s|\theta^{\text{NN}})$, aiming to align the NN predictions closely with both the physics model predictions and the ground truth. Thus, the loss function of the NN component contains two parts, LOSS_1 and LOSS_2:

$$\text{LOSS}_1 := \left(f^{\text{NN}}\left(s\middle|\theta^{\text{NN}}\right) - g(s)\right)^2 \tag{4}$$

$$\text{LOSS}_2 := \left(f^{\text{NN}}\left(s\middle|\theta^{\text{NN}}\right) - f^{\text{Phy}}\left(S^{\text{Phy}}(s)\middle|\theta^{\text{Phy}}\right)\right)^2 \tag{5}$$

$$\theta^{\text{PINN}} = \underset{\theta^{\text{PINN}} \in \Theta^{\text{PINN}}}{\text{argmin}} \sum_{s \in \mathscr{S}^{\text{Train}}} \mu \cdot \text{LOSS}_1 + (1-\mu) \cdot \text{LOSS}_2 \tag{6}$$

where μ represents the weight hyperparameters to be tuned. The parameter vector $\theta^{\text{PINN}} \in \Theta^{\text{PINN}}$ is a concatenation of the physics component and NN component, i.e., $\theta^{\text{PINN}} := [\theta^{\text{Phy}}, \theta^{\text{NN}}] \in \Theta^{\text{Phy}} \times \Theta^{\text{NN}}$. During prediction, the NN component $f^{\text{NN}}(s|\theta^{\text{NN}})$ is responsible for the prediction result.

3. PERL use case: vehicle trajectory prediction

Section 2 introduces the PERL model methodology. In this section, we apply the PERL model to a specific prediction problem, vehicle trajectory prediction, as described below. As shown in Fig. 2, an input state stems from a set of K consecutive vehicle trajectories, indexed by $k \in \mathscr{K} := \{1, 2, \cdots, K\}$ from the upstream lead vehicle indexed by 1 through the downstream ego vehicle by K, at T^{b} historical time points $\mathscr{T}^{\text{b}} := \{t_0 - (T^{\text{b}} - 1)\delta, \cdots, t_0 - \delta, t_0\}$ separated by time interval δ. The input state is defined as $s = [a_{kt}, v_{kt}, \Delta d_{kt}]_{\forall k \in \mathscr{K}, t \in \mathscr{T}^{\text{b}}}$, where a_{kt}, v_{kt}, and Δd_{kt} are the acceleration, speed, and spacing (from the preceding vehicle) of vehicle k at time t, respectively. With this, $|\mathscr{S}| = 3|\mathscr{K}|T^{\text{b}}$. The corresponding output response $g(s)$ is the ground-truth trajectory of ego vehicle K at future time points $\mathscr{T}^{\text{f}} := \{t_0 + \delta, t_0 + 2\delta, \cdots, t_0 + T^{\text{f}}\delta\}$, i.e., $g(s) = [a_{Kt}]_{t \in \mathscr{T}^{\text{f}}}$, where a_{Kt} is the ground-truth acceleration of ego vehicle K at a future time point t. With this, $|\mathscr{Y}| = T^{\text{f}}$.

In the physics component, $S^{\text{Phy}}(\cdot)$ projects s to a much lower-dimensional space, i.e., $S^{\text{Phy}}(s) = [a_{kt_k}, v_{kt_k}, \Delta d_{kt_k}]_{k \in \mathscr{K}, t_k \in [t_0 - \tau_k, \cdots, t_0]}$ or its subset, where τ_k is a short reaction time depending on the specific physics model $t_0 - \tau_k \in \mathscr{T}^{\text{b}}$. With this, $\left|\Theta^{\text{Phy}}\right|$ is on the order of $\left|S^{\text{Phy}}(s)\right|$, i.e., $O(\left|S^{\text{Phy}}(s)\right|)$, and thus much smaller than $|\mathscr{S}|$. $\mathscr{Y}^{\text{Phy}}$ is the predicted ac-

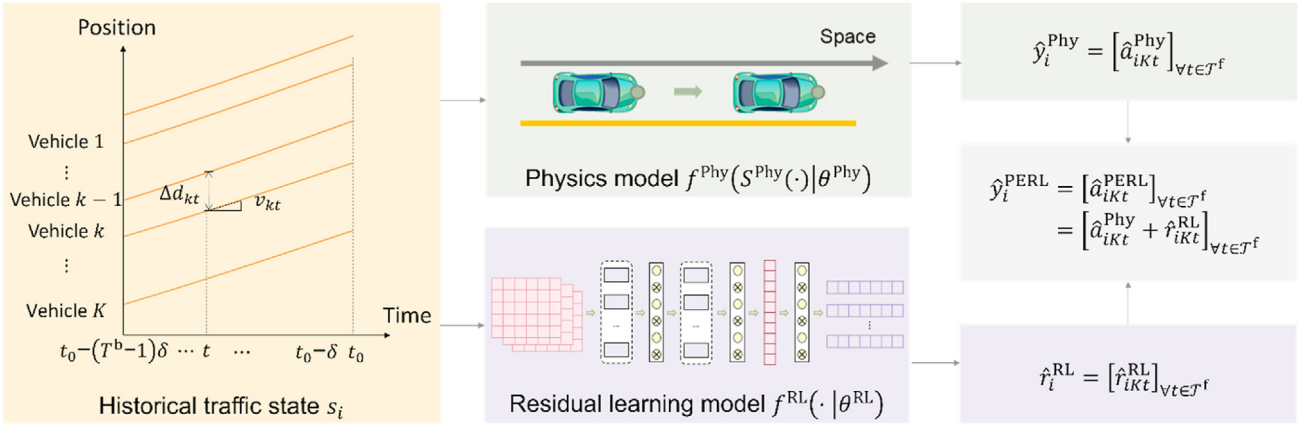


Fig. 2. Workflow of the PERL model (examples of vehicle trajectory prediction).

celeration of vehicle K: $[\hat{a}_{Kt}^{\text{Phy}}(\theta^{\text{Phy}})]_{\forall t\in\mathscr{T}^{\text{f}}} = f^{\text{Phy}}(S^{\text{Phy}}(s)|\theta^{\text{Phy}})$. Therefore, we obtain the residual acceleration predicted by the physics model $r_{Kt}(\theta^{\text{Phy}}) := g(s) - \hat{a}_{Kt}^{\text{Phy}}(\theta^{\text{Phy}}), \forall t \in \mathscr{T}^{\text{f}}$.

In the residual learning component, $|\Theta^{\text{RL}}|$ is generally polynomial in relation to $|s|$ and the architecture of the model; thus, $|\Theta^{\text{RL}}|$ is likely much greater than $|\Theta^{\text{Phy}}|$. $\mathscr{Y}^{\text{RL}}$ is the predicted residual of vehicle K: r_{Kt}, i.e., $[\hat{r}_{Kt}^{\text{RL}}(\theta^{\text{RL}})]_{\forall t\in\mathscr{T}^{\text{f}}} = f^{\text{RL}}(s|\theta^{\text{RL}})$. Therefore, the output of the PERL model is $f^{\text{PERL}}(s|\theta^{\text{PERL}}) = [\hat{a}_{Kt}^{\text{Phy}}(\theta^{\text{Phy}}) + \hat{r}_{Kt}^{\text{RL}}(\theta^{\text{RL}})]_{\forall t\in\mathscr{T}^{\text{f}}}$.

4. PERL application example

On the basis of the above defined PERL vehicle trajectory prediction use case, this section provides a detailed example that demonstrates the application of the PERL model to vehicle trajectory prediction, including data preparation, model component adoption for the physics component and residual learning component, and baseline model adoption.

4.1. Data and metrics

We utilized two types of data in our study. For the validation of the PERL model, we adopted the NGSIM US101 data (Dong et al., 2021), focusing on human-driven vehicles (HVs). In the sensitivity analysis, we employed data on purely autonomous vehicle (AV) platoons from OpenACC (Makridis et al., 2021) to validate the model's generalizability across different vehicle trajectory types.

To ensure sufficient training and comprehensive evaluation, we thoroughly prepared the data and evaluated methods on the basis of different scenarios. A total of $I = 20{,}000$ samples, $\delta = 0.1s$, $\omega^{\text{Train}} = 0.6$, $\omega^{\text{Val}} = 0.2$. Each sample comprises $K = 4$ consecutive vehicles. We evaluated two typical prediction scenarios, one-step and multistep predictions, in this study. For one-step prediction, $T^{\text{h}} = 50$, $T^{\text{f}} = 1$; for multistep prediction, $T^{\text{h}} = T^{\text{f}} = 50$. The predicted speed of vehicle K at time t in sample $i \in \mathscr{I}$ is calculated as $\hat{v}_{iKt} = v_{iKt_0} + \sum_{t'=t_0}^{t} \hat{a}_{iKt'}\delta, \forall t' \in \mathscr{T}^{\text{f}}$. The evaluation for one-step and multistep prediction on the test set $\mathscr{I}^{\text{test}}$ includes two metrics: the MSEs of acceleration and speed prediction: $\text{MSE}_{\text{test}}^{a} = \frac{1}{|\mathscr{I}^{\text{Test}}|T^{\text{f}}} \sum_{i\in\mathscr{I}^{\text{Test}},t\in\mathscr{T}^{\text{f}}} (a_{iKt} - \hat{a}_{iKt})^2$ and $\text{MSE}_{\text{test}}^{v} = \frac{1}{|\mathscr{I}^{\text{Test}}|T^{\text{f}}} \sum_{i\in\mathscr{I}^{\text{Test}},t\in\mathscr{T}^{\text{f}}} (v_{iKt} - \hat{v}_{iKt})^2$. The convergence of the training process is measured by the change in the MSE of acceleration and speed prediction on the validation set: $\text{MSE}_{\text{val}}^{a} = \frac{1}{|\mathscr{I}^{\text{Val}}|T^{\text{f}}} \sum_{i\in\mathscr{I}^{\text{Val}},t\in\mathscr{T}^{\text{f}}} (a_{iKt} - \hat{a}_{iKt})^2$ and $\text{MSE}_{\text{val}}^{v} = \frac{1}{|\mathscr{I}^{\text{Val}}|T^{\text{f}}} \sum_{i\in\mathscr{I}^{\text{Val}},t\in\mathscr{T}^{\text{f}}} (v_{iKt} - \hat{v}_{iKt})^2$.

4.2. PERL physics component

A model adapted from Newell's car-following model is adopted as the physics component of the PERL model. Additionally, for sensitivity analysis, the intelligent driver model (IDM) (Treiber et al., 2000) and the full velocity difference (FVD) model (Yao et al., 2019) serve as alternative physics components for validation.

Initially, we utilized an adapted Newell model characterized by a single parameter w denoting the wave speed, as shown in Fig. 3. The physics model output $\mathscr{Y}^{\text{Phy}} = [\hat{a}_{iKt}^{\text{Phy}}(\theta^{\text{Phy}})]_{\forall t\in\mathscr{T}^{\text{f}},i\in\mathscr{I}}$ is given by

$$\hat{a}_{iKt}^{\text{Phy}} = a_{ik't-\frac{D_{iKk't}}{w}}, \forall k' \in \{1,2,\ldots,K-1\}, t \in \mathscr{T}^{\text{f}}, i \in \mathscr{I} \tag{7}$$

where $D_{iKk't}$ represents the position distance between vehicle K and vehicle k' at time $t \in \mathscr{T}^{\text{f}}$ in sample $i \in \mathscr{I}$.

The IDM and FVD models are adopted for sensitivity analysis as alternative physics models. The IDM model provides a model acceleration function as a continuous function of speed, gap, and speed difference and is expressed as

$$\hat{a}_{iK(t_0+1)}^{\text{Phy}} = \bar{a}\left[1 - \left(\frac{v_{iKt_0}}{v^{\text{f}}}\right)^4 - \left(\frac{S(v_{iKt_0}, \Delta v_{iKt_0})}{\Delta x_{Kt_0}}\right)^2\right], \forall i \in \mathscr{I} \tag{8}$$

$$S(v_{iKt_0}, \Delta v_{iKt_0}) = S_0 + T^{\text{g}} v_{iKt_0} - \frac{v_{iKt_0} \cdot \Delta v_{iKt_0}}{2\sqrt{\bar{a}\bar{b}}}, \forall i \in \mathscr{I} \tag{9}$$

where $\Delta d_{i,n,t}$ is the relative distance between two adjacent vehicles, $S(v_{iKt_0}, \Delta v_{iKt_0})$ is the desired space headway function and is calculated from the speed $v_{n,t}$ and the relative speed $\Delta v_{iKt_0} := v_{iKt_0} - v_{i(K-1)t_0}, \forall i \in \mathscr{I}$, v^{f} is the free flow speed, $\bar{a}$ is the maximum acceleration, $\bar{b}$ is the

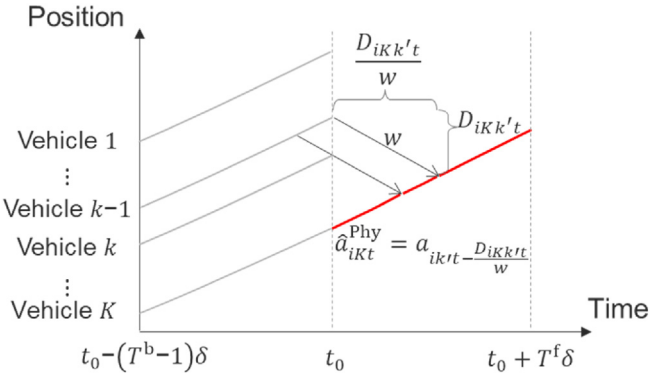


Fig. 3. Adapted Newell model for vehicle trajectory prediction.

comfortable deceleration, and S_0 is the minimum space. T^g is the desired time gap.

The FVD model focuses on the speed difference between vehicles, considering both relative speeds and spatial gaps. This model comprehensively represents car-following dynamics, especially in high-density traffic scenarios where speed differences play a pivotal role.

$$\hat{a}^{\text{Phy}}_{iK(t_0+1)} = \kappa[V(\Delta d_{iKt_0}) - v_{iKt_0}] + \lambda \Delta v_{iKt_0}, \forall i \in \mathcal{I} \tag{10}$$

$$V(\Delta d_{iKt_0}) := V_1 + V_2 \tanh[C_1(\Delta d_{iKt_0} - l_c) - C_2], \forall i \in \mathcal{I} \tag{11}$$

where κ and λ are sensitivity parameters and where $V(\Delta d_{iKt_0})$ is the optimal speed that the drivers prefer. We adopted the calibrated parameters for $V(\Delta d_{iKt_0})$ as $V_1 = 6.75$ m/s, $V_2 = 7.91$ m/s, $C_1 = 0.13$ m^{-1}, and $C_2 = 1.54$.

We employed the Monte Carlo method for physics model calibration, which randomly selects a certain number of samples from the dataset. The calibrated parameters of the physics models are shown in Table 1. As the training data size increased, the calibration results tended to stabilize (Long et al., 2024a), and the calibrated parameter was near the value in the literature (Duret et al., 2011; Laval and Leclercq, 2010). The calibrated parameters of the physics model are used as initial values in the PINN model and PERL model.

4.3. PERL residual learning component

For the residual learning component, we compared four groups of learning-based models—the RNN model, convolution model, generation model, and attention model—to validate the robustness and generalizability of the PERL model.

For the RNN model, we choose the gated recurrent unit (GRU) model (Fig. 4a). GRU models are a type of recurrent NN that allows connections between neurons to form a directed cycle, thus making it possible to exhibit dynamic temporal behavior. The GRU has also been applied in car-following behavior modeling (Wang et al., 2018).

For the convolution model, we selected CLSTM, a model that combines convolutional layers with long short-term memory (LSTM) (Fig. 4b) (Lee et al., 2017; Pereira et al., 2022) and is suitable for spatiotemporal data.

For the generation model, we choose the variational autoencoder (VAE) (Fig. 4c), a generative model used to learn the latent distribution of data (De Miguel et al., 2022; Ivanovic et al., 2021). VAE offers two advantages: First, it exhibits stable training dynamics, and second, its interpretable latent space facilitates understanding of the learned representations by the model.

For the attention model, we opt for the Informer model (Fig. 4d), which is a transformer-based architecture (Zhou et al., 2021). It introduces a masking mechanism in the self-attention mechanism to ensure that the model only attends to information preceding the current time step, preventing leakage of future information when processing sequential data. Additionally, it employs a multistep input representation method to better capture long-term dependencies in time series data.

Fig. 4 illustrates the structures of the four models, which focus only on the primary model layers; the activation layers are omitted for clarity. The detailed model configurations and parameters are listed in Tables 2–5. The input to each model includes T^b time steps of acceleration, speed, and spacing data from K consecutive vehicles (Table 6).

4.4. Baseline model adoption

With respect to the three baseline models, the physics model employs the adapted Newell model, which aligns with the physics component of PERL. The NN model structure aligns with the residual learning component of PERL but with more parameters. Its prediction target is not the residuals but the future acceleration of the subject vehicle over T^f steps. The PINN model combines the adapted Newell model for its physics component and mirrors the NN model structures in its NN component.

5. Results

On the basis of the methodology and application example introduced in Section 2 and 3, this section presents a detailed comparative analysis of the multistep prediction performance of various learning-based models, including the PERL, NN, PINN, and baseline models, under different training data sizes and conditions.

5.1. Multistep prediction comparison

Fig. 5 compares the prediction performance on acceleration and

Table 1
Calibrated parameters of the physics models with different training sample sizes.

Model	Parameter	Training sample size						
		300	500	1000	2000	5000	10,000	12,000
Adapted Newell model	w (m/s)	4.044	4.150	4.122	4.027	4.030	4.010	4.010
	(variance)	0.088	0.017	0.016	0.018	0.001	0.000	0.000
IDM	v^f (m/s)	22.821	22.512	22.084	22.377	22.125	22.125	22.495
	(variance)	0.035	0.095	0.013	0.129	0.003	0.003	0.217
	$\bar{a}$ (m/s^2)	0.682	0.475	0.626	0.540	0.613	0.613	0.911
	(variance)	0.095	0.090	0.175	0.099	0.192	0.192	0.000
	$\bar{b}$ (m/s^2)	2.562	2.271	2.473	2.262	2.478	2.478	2.859
	(variance)	0.205	0.141	0.246	0.087	0.237	0.237	0.003
	S_0 (m)	1.712	1.703	1.627	1.608	1.615	1.615	1.627
	(variance)	0.010	0.021	0.013	0.006	0.010	0.010	0.013
	T^g (s)	1.200	1.216	1.230	1.162	1.266	1.266	1.132
	(variance)	0.005	0.002	0.000	0.020	0.003	0.003	0.017
FVD	k	0.108	0.073	0.088	0.044	0.108	0.076	0.007
	(variance)	0.003	0.003	0.002	0.003	0.005	0.006	0.000
	λ	0.498	0.397	0.478	0.452	0.445	0.284	0.137
	(variance)	0.019	0.041	0.041	0.076	0.029	0.037	0.000
	V_0 (m/s)	19.779	18.814	19.647	23.034	21.542	22.093	24.158
	(variance)	15.982	10.059	19.536	26.672	17.456	14.257	5.000
	b (m/s^2)	10.084	10.746	9.637	8.406	10.327	11.243	7.954
	(variance)	4.935	14.088	5.636	10.330	8.447	11.241	4.340
	β	2.561	3.011	2.824	2.449	2.383	2.133	1.724
	(variance)	1.338	1.600	1.585	0.733	2.388	3.520	0.079

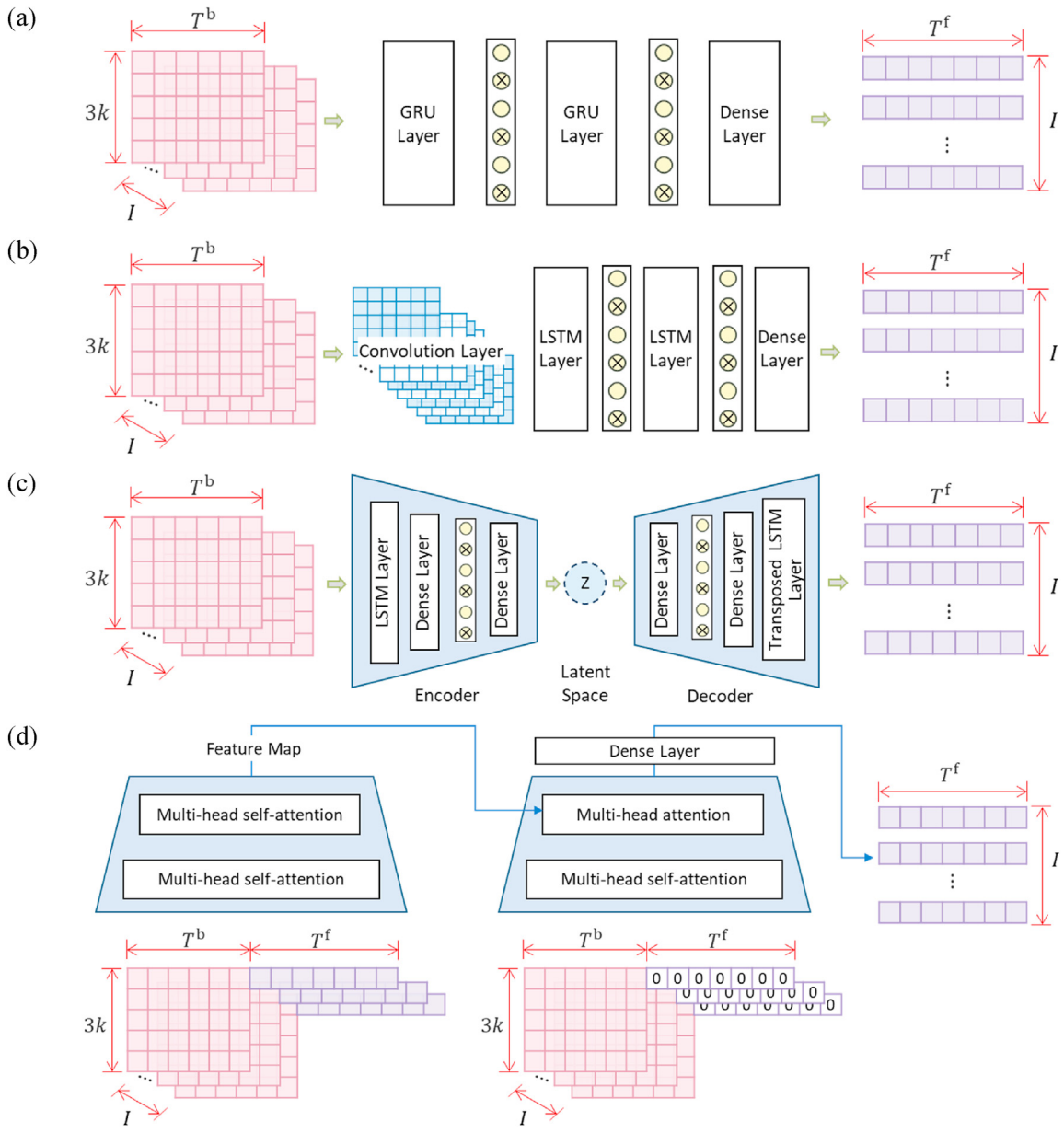


Fig. 4. Model structures of (a) GRU model, (b) CLSTM model, (c) VAE model, and (d) Informer model.

Table 2
Model structure and parameter setting of the GRU-based models.

	NN	PINN	PERL
GRU layer	Units = 128	Units = 128	Units = 96
Dropout layer	0.3	0.3	0.3
GRU layer	Units = 128	units = 128	Units = 64
Dropout layer	0.3	0.3	0.3
Activation layer	Relu	Relu	Relu

Table 3
Model structure and parameter setting in CLSTM-based models.

	NN	PINN	PERL
Convolution layer	Filters = 64, kernel_size = 3, ReLU	Filters = 64, kernel_size = 3, ReLU	Filters = 64, kernel_size = 3, ReLU
LSTM layer	Units = 128	Units = 128	Units = 96
Dropout layer	0.3	0.3	0.3
LSTM layer	Units = 128	Units = 128	Units = 64
Dropout layer	0.3	0.3	0.3
Activation layer	Relu	Relu	Relu

speed of PERL and the baseline using different learning-based models in multistep prediction. For all four groups of learning methods, the results show that the PERL model exhibits superior predictability with less training data. In scenarios characterized by very limited data availability (e.g., 300 training data points), both the PERL and PINN models outperform the NN model in terms of predictability. As the dataset expands beyond 1000 training samples, the performance differential between the PERL model and the baseline models, including the NN model, becomes markedly less pronounced (in Fig. 3). In this context, all the models demonstrate remarkable predictability, achieving acceleration prediction errors as low as $MSE^a_{Test} = 0.07\ m^2/s^4$. Notably, the PINN model slightly outperforms the NN and physics models when the training dataset size is limited to 300. This underperformance is likely because each model underwent 30 independent training sessions with randomly selected samples, potentially leading to instances where the PINN results

Table 4
Model structure and parameter setting in VAE-based models.

		NN	PINN	PERL
Encoder	Convolution layer	Filters = 64, kernel_size = 3, ReLU	Filters = 64, kernel_size = 3, ReLU	Filters = 64, kernel_size = 3, ReLU
	LSTM layer	Units = 128	Units = 128	Units = 96
	Dropout layer	0.3	0.3	0.3
	LSTM layer	Units = 128	Units = 128	Units = 64
Latent Space		Demention = 20	Demention = 20	Demention = 20
Decoder	Activation layer	Relu	Relu	Relu

Table 5
Model structure and parameter setting in VAE-based models.

		NN	PINN	PERL
Encoder	Self-attention layer	*n*_heads = 12	*n*_heads = 8	*n*_heads = 8
	Convolutional layer	Down convolution layer kernel_size = 3 **Activation = ELU**	Down convolution layer kernel_size = 3 **Activation = ELU**	Down convolution layer kernel_size = 3 **Activation = ELU**
	Feed-forward network	Convolutional layer1 kernel_size = 1 Convolutional layer2 kernel_size = 1	Convolutional layer1 kernel_size = 1 Convolutional layer2 kernel_size = 1	Convolutional layer1 kernel_size = 1 Convolutional layer2 kernel_size = 1
Decoder	Self-attention layer	*n*_heads = 12	*n*_heads = 8	*n*_heads = 8
	Cross-attention layer	*n*_heads = 12	*n*_heads = 8	*n*_heads = 8
	Feed-forward network	Convolutional layer1 kernel_size = 1 Convolutional layer2 kernel_size = 1	Convolutional layer1 kernel_size = 1 Convolutional layer2 kernel_size = 1	Convolutional layer1 kernel_size = 1 Convolutional layer2 kernel_size = 1

in exceptionally poor predictive performance and adversely affects its average performance relative to the NN and physics models.

When comparing the four groups of learning methods with a small training dataset size of 300, the models based on the Informer architecture (NN, PINN, and PERL) performed notably well. This can be attributed primarily to two advantages of the Informer model. First, the Informer's outputs are relatively stable, ensuring reliable predictions even with limited data. Second, unlike traditional recurrent neural networks such as CLSTMs and GRUs, which depend on the output and hidden states of previous time steps, the Informer model processes data in a nonsequential manner, reducing the accumulation of prediction errors.

With larger datasets, the models based on CLSTM exhibited the best performance, achieving acceleration prediction errors as low as $MSE^{a}_{Test} = 0.056\ m^2/s^4$ at a dataset size of 15,000. The convolutional layers preprocess the data, and the LSTM is sufficient for managing the relatively straightforward car-following dynamics in our study, whereas the attention mechanism of the Informer is more challenging to train and offers limited benefits for this specific scenario. Moreover, the GRU-based NN model performs worse than the LSTM-based NN model does, primarily because of the simpler structure of the GRU, which struggles with issues such as vanishing or exploding gradients, particularly in longer sequences. This simplicity may prevent the GRU from capturing complex sequence patterns or long-term dependencies as effectively as the LSTM. Additionally, the GRU-based PERL model (GRU + Adapted newell) also underperformed compared with the LSTM-based PERL (LSTM + Adapted newell) model. The VAE model was less suited to this prediction task because data are assumed to be generated from latent variables following a Gaussian distribution, which may restrict the model's ability to fit the trajectory data.

One contributing factor to PERL's superior performance in predicting acceleration residuals, as opposed to the NN model's ability to predict acceleration, is the different distribution patterns of acceleration residuals and acceleration. The variance in acceleration is greater than that in acceleration residuals across all samples, as depicted in Fig. 5, This difference is further highlighted by comparing the distributions of acceleration and acceleration residuals in a single sample, as illustrated in Figs. 6 and 7. The observed lower variance in the acceleration residual suggests that in scenarios with limited data, the distribution of the acceleration residual may more accurately mirror the actual distribution, facilitating better prediction outcomes. On the other hand, the higher variance in acceleration might lead to a less accurate representation of the true distribution in small sample scenarios, thereby potentially diminishing the prediction accuracy. These findings align with the operational principles of the PERL model, which chiefly involves predicting the residual of the physics model, a task seemingly well suited for situations with restricted data availability. This stands in contrast to the PINN model, which aims at the prediction of acceleration, a process that could be adversely affected by the noted higher variance, especially in data-scarce conditions. Hence, this visual representation further underscores the sample efficiency and effectiveness of the PERL model in acceleration prediction, particularly when data resources are limited.

Fig. 5 shows that when the training data size is 12,000, the prediction accuracies of the PERL, PINN, and NN models are comparable, and all of them are superior to those of the physics model. This is validated by the six examples in Fig. 8. The results indicate that both the PINN and NN models can predict the overall trend of acceleration changes, but they fail to fit the local acceleration oscillation. The physics model can account for local acceleration oscillation on the basis of physical principles, but as the physics model is calibrated with multiple vehicle trajectories, its results deviate significantly from real-world values. Only the PERL model can predict the overall trend of acceleration change and keep the local acceleration oscillation results from its physics model component.

5.2. Convergence comparison

To ensure a fair comparison of convergence among different models, we aligned the number of parameters in the NN, PINN, and PERL models. Specifically, we standardized the LSTM architecture within the NN component of the NN and PINN models and the residual learning component of the PERL model, setting both LSTM layers to 128 and 64 units, respectively. The results indicate that PERL converges more rapidly than both the NN and the PINN models do with various training data sizes (Fig. 9). In scenarios with limited data (300 training data points),

Table 6
Comparison of the prediction performance on the HV trajectory and AV trajectory via the CLSTM model.

Training data size	Trajectory data	MSE^{a}_{test} (m^2/s^4)				MSE^{v}_{test} (m^2/s^2)			
		Physics	NN	PINN	PERL	Physics	NN	PINN	PERL
1000	HV data	0.256	0.123	0.130	0.091	0.682	0.530	0.358	0.319
	CAV data	0.195	0.088	0.089	0.086	0.516	0.191	0.217	0.190
15,000	HV data	0.251	0.058	0.059	0.056	0.653	0.201	0.177	0.194
	CAV data	0.194	0.047	0.046	0.047	0.512	0.176	0.176	0.176

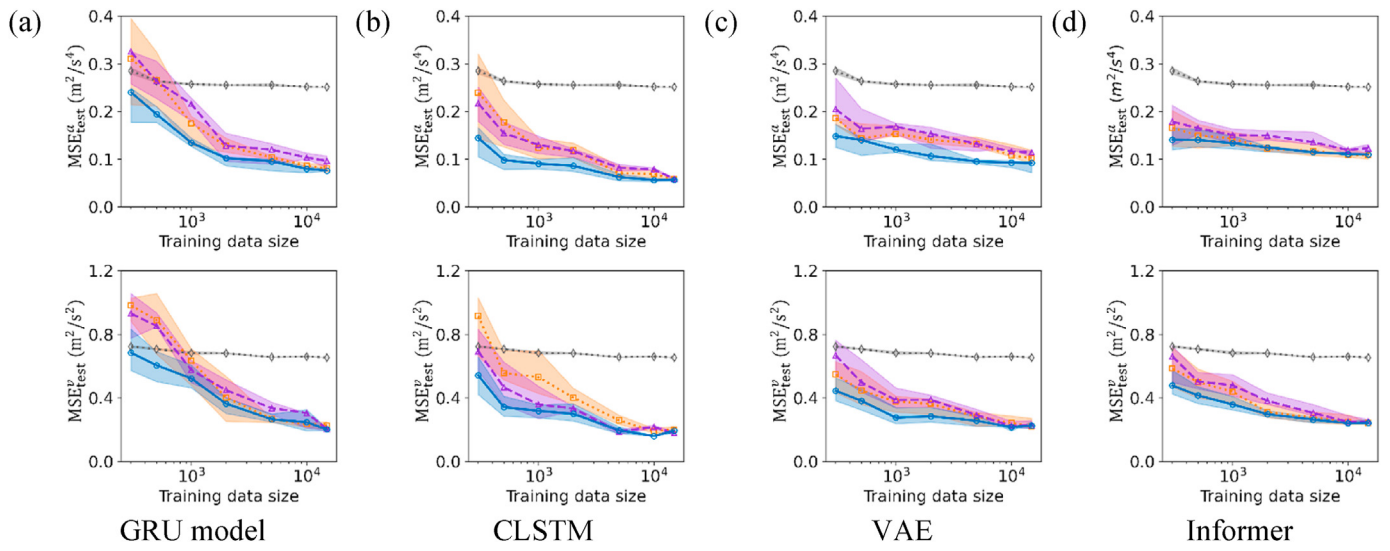


Fig. 5. PERL and baseline prediction performance of different learning-based models.

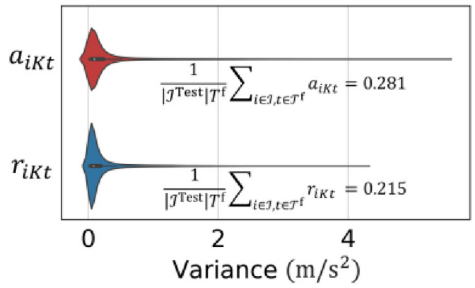


Fig. 6. Differences in the distributions of acceleration a_{iKt} and acceleration residual r_{iKt} of all samples.

the PINN model converges at approximately 80 epochs, and the NN model converges at approximately 140 epochs, whereas the PINN model requires a substantially shorter time, approximately 1000 epochs (Fig. 10). This unstable training behavior of the PINN arises because different parts of the loss function dominate the decrease at various times: Sometimes, it is influenced by the neural network, and other times, it is influenced by the physics rules. This unbalanced back-propagated gradient results in an unstable learning process (Wang et al., 2022). The NN model also converges slower than PERL despite having a nearly identical number of parameters (Fig. 10). This is because the NN model needs to learn all the kinematic rules, whereas PERL only has to learn the features of the residuals. The convergence speed of the PINN is also slower than that of the NN model because its objective function includes terms from both the neural network and the physics model, making the gradient search process more challenging.

This result highlights the advantage of the PERL model, which employs the same LSTM architecture and nearly identical number of parameters as the NN and PINN models but uses a more focused prediction approach that enables faster convergence and more efficient computation using fewer parameters.

5.3. Sensitivity analysis

In the sensitivity analysis, we first examined the impact of different physics models on the results. We subsequently validated the model's generalizability across various vehicle types.

5.3.1. Impact of physics car-following models

In single-step prediction tasks, which are relatively straightforward tasks, it is feasible to employ car-following models that consider only the information of the preceding vehicle, such as the IDM and FVD models. We integrate the PERL model with the IDM and FVD models, utilize them as the physics model, and pair them with an LSTM for residual learning.

PERL (LSTM + IDM) outperforms the physics model, the NN model (LSTM), and the PINN (LSTM + IDM) when the training data size is smaller than 10,000, as shown in Fig. 11. This suggests the versatility of the PERL model, highlighting its compatibility with the IDM physics model. The IDM model effectively captures certain physics laws, and the residuals in acceleration prediction from the IDM model can be proficiently learned by the LSTM. The convergence of the IDM-based PERL (LSTM + IDM) is faster than that of both the NN (LSTM) and the PINN (LSTM + IDM) models, as shown in Fig. 12. This emphasizes the ability of the IDM model to delineate a significant portion of the physics laws. Moreover, the IDM model can learn the acceleration residuals with less data, and its training convergence is faster than directly learning the acceleration residuals via the NN model and PINN model.

The performance of PERL (LSTM + FVD) is less effective than that of

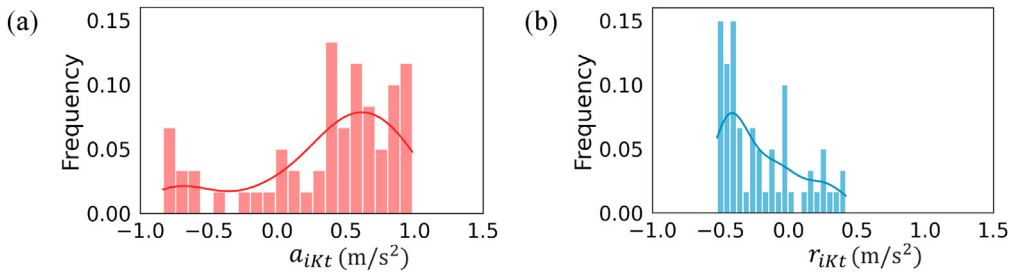


Fig. 7. Differences in the distributions of (a) acceleration a_{iKt} and (b) acceleration residual r_{iKt} in one sample.

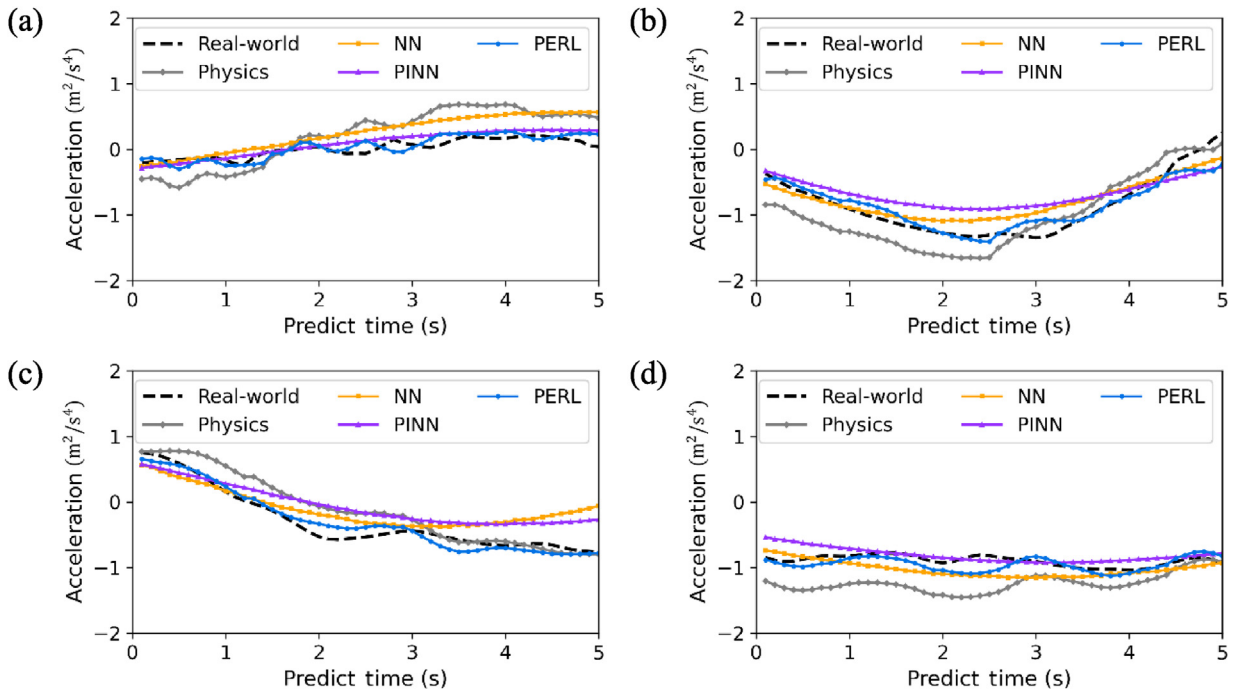


Fig. 8. Comparison of real-world acceleration and predicted results of the PERL (Newell + CLSTM) model and baseline models under four different driving scenario: (a) acceleration, (b) deceleration, (c) cruise, and (d) deceleration.

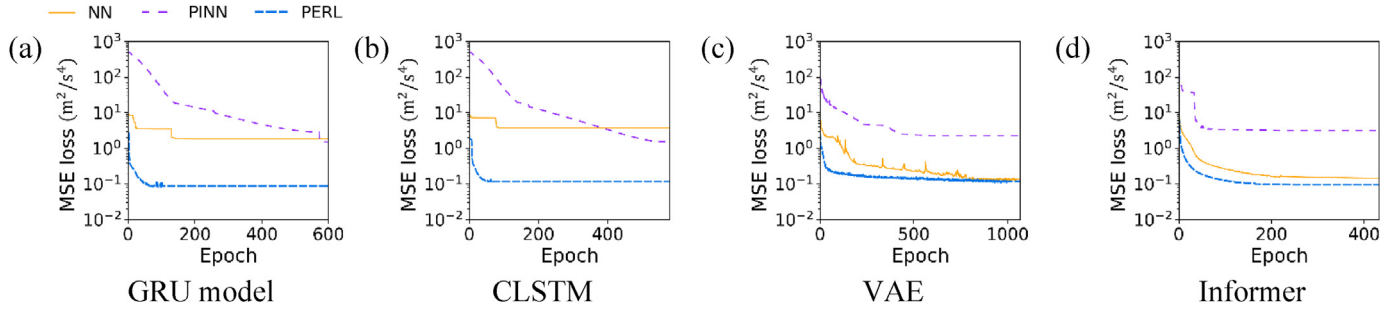


Fig. 9. Training processes of the NN, PINN, and PERL structure using various models with a sample size of 1,000, utilizing various learning-based models: (a) GRU model, (b) CLSTM model, (c) VAE model, and (d) Informer model.

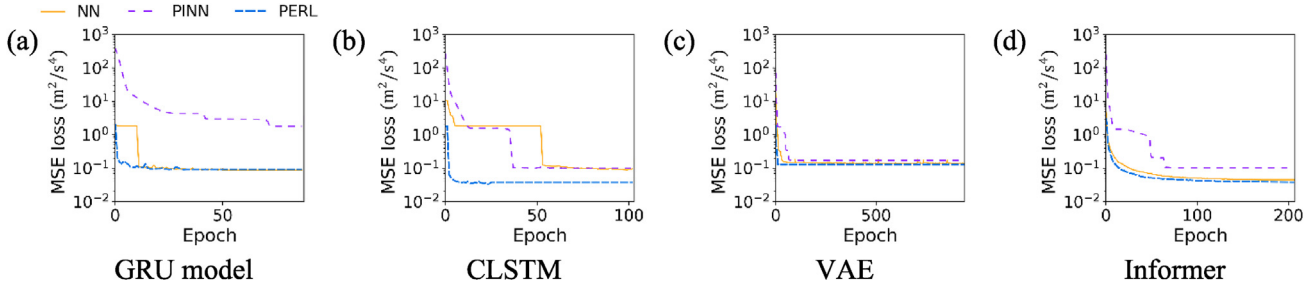


Fig. 10. Training processes of the NN, PINN, and PERL structure using various models with a sample size of 15,000, utilizing various learning-based models: (a) GRU model, (b) CLSTM model, (c) VAE model, and (d) Informer model.

PERL (LSTM + IDM) with small amounts of data, as shown in Fig. 13. This mainly results from the poor performance of the FVD model as a physical model when it is calibrated with small amounts of data. However, even though the physical model is not predictive, PERL (LSTM + FVD) still has a comparative prediction with the NN model and is better than the PINN model. The results also show that the PERL model exhibits quick convergence rates for both small and large training datasets, as shown in Fig. 14.

5.3.2. *Prediction results on AV data*

The proposed method is tested on AV data to validate the model's generalizability. Table 2 compares the prediction results for acceleration and speed across different training data sizes via the physics model, LSTM-based NN, PINN, and PERL models. We observed that the

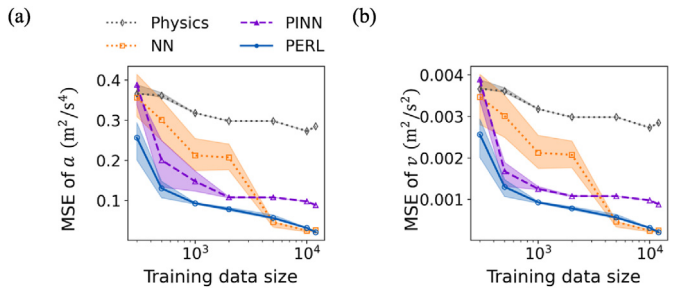


Fig. 11. PERL (LSTM + IDM) and baseline prediction performance: (a) acceleration prediction result and (b) speed prediction result.

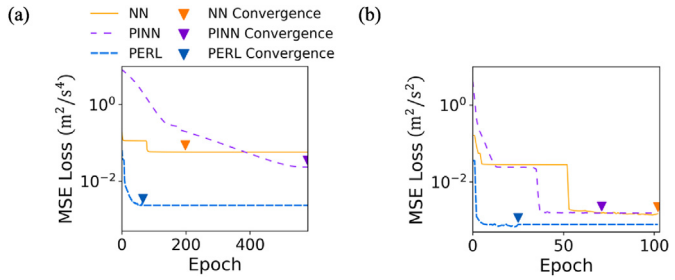


Fig. 12. Training processes of PERL (LSTM + IDM) and baseline models: (a) training data seize = 500 and (b) training data seize = 1,500.

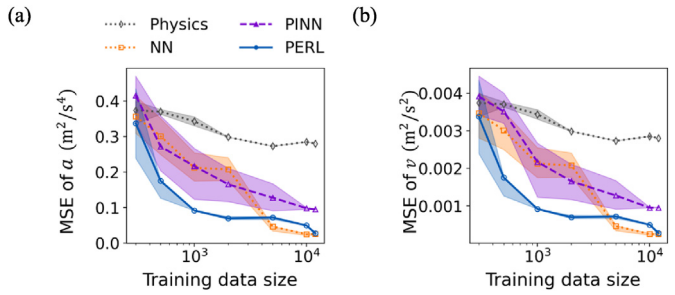


Fig. 13. PERL (LSTM + IDM) and baseline prediction performance: (a) acceleration prediction result and (b) speed prediction result.

predictive performance for AV data is generally better than that for HV data. One reason for this is that AV behaviors are governed by specific control algorithms, making their actions more rational and predictable, whereas HV trajectories exhibit many unstable and random behaviors. Another reason is that the range of acceleration distribution in AVs is much tighter than that in HVs. Fig. 15 compares the distribution of acceleration in the AV and HV trajectory datasets, showing that 50% of accelerations are contained within the range of −0.17–0.17. This

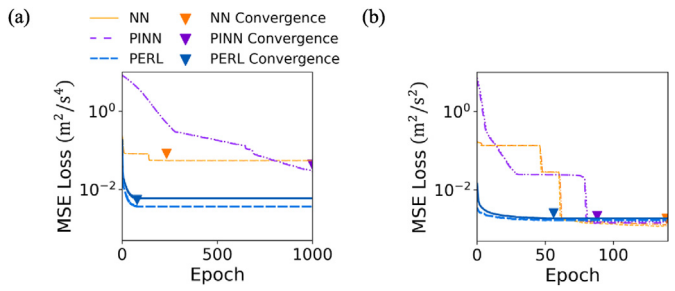


Fig. 14. Training processes of PERL (LSTM + FVD) and baseline models: (a) training data seize = 500 and (b) training data seize = 1,500.

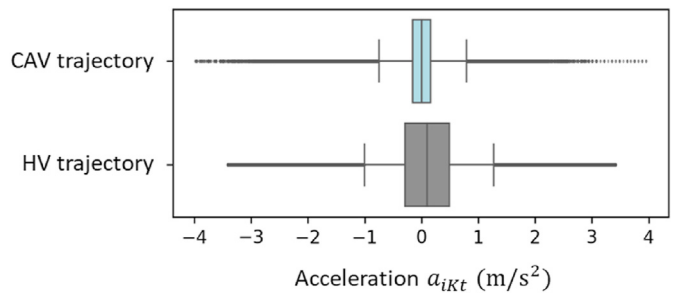


Fig. 15. Comparison of the acceleration distributions in the CAV and HV trajectory datasets.

narrower variance in the prediction target makes the prediction task easier.

6. Conclusions and future work

This study presented the novel PERL model as a potent tool for trajectory prediction, aiming to address inherent limitations found in existing physics models and NN models. The PERL model leverages the power of residual learning, a relatively unexplored concept in the traffic domain, providing the research community with a new perspective on traffic modeling and prediction.

We compared the predictive performance of the PERL model with that of traditional physics and NN models using both HV and AV trajectory data. The results demonstrated the superiority of the PERL model in both one-step and multistep acceleration prediction tasks. Notably, the PERL model consistently outperformed all the other tested models across different data sources and vehicle types. PERL also has faster initial convergence during the training process than the traditional NN model and PINN. For the sensitivity analysis, we assessed the performance of PERL via a diverse array of residual learning models: the GRU, CLSTM, VAE, and transformer models. Each model represents a different neural network approach—GRU and CLSTM for temporal dynamics, VAE for generative capabilities, and Informer for attention mechanisms—highlighting the versatility of PERL across various data processing contexts. Additionally, we integrate various physics-based car-following models, including the IDM and FVD, to further validate the adaptability and robustness of the PERL framework.

Future research could focus on further exploring the advantages of the PERL model structure and addressing potential limitations to enhance its practical applicability. First, the model introduced in this study does not include residual terms in the inputs of the residual learning model. This design choice ensures that the model can be broadly applicable to nontime series scenarios. In fact, in time series forecasting scenarios, incorporating residuals from historical periods of the physics model into the inputs can further enhance the predictive performance of the model. The application of the proposed method in nontime series contexts and the potential benefits of using residual terms as inputs will be further explored in future research.

Second, the robustness of PERL against biased physics models presents an exciting area for exploration. This characteristic was not thoroughly investigated in this study, but it presents a promising avenue for future research. Systematic evaluations of the performance under different bias levels can provide valuable insights, along with developing techniques to mitigate or correct bias in the physics model (Li et al., 2024). Furthermore, the flexibility of the PERL framework allows for the adaptation of its physics model to various types and the extension of its neural network component to include advanced models such as large language models (LLMs) (Qu et al., 2023), thus improving the accuracy and generalizability of PERL across different domains.

Third, it is important to recognize a potential constraint of the PERL model: its robustness against noisy data. Future work should conduct

comprehensive investigations to assess its performance under noisy data conditions. Moreover, we explore and develop novel techniques and methodologies to enhance its ability to handle and mitigate the adverse effects of noisy data effectively.

Fourth, future studies could update the PERL model to capture stochastic driving behaviors by first collecting synthetic data reflecting varying traffic densities, driver behaviors, and environmental conditions (Han et al., 2023). Then, expanding the residual learning component to predict distributions of residuals, rather than single values, could further refine its effectiveness. Techniques such as Bayesian neural networks or Gaussian processes enable this probabilistic modeling, which is crucial for applications such as CAV trajectory planning. By utilizing predictive distributions, PERL can support more robust and informed decision-making under uncertainty, thereby increasing safety and operational efficiency in autonomous driving scenarios.

CRediT authorship contribution statement

Keke Long: Writing – review & editing, Writing – original draft, Visualization, Validation, Methodology. **Zihao Sheng:** Investigation. **Haotian Shi:** Writing – review & editing. **Xiaopeng Li:** Writing – review & editing, Supervision, Project administration, Methodology, Funding acquisition, Conceptualization. **Sikai Chen:** Writing – review & editing. **Soyoung Ahn:** Writing – review & editing.

Replication and data sharing

We utilize publicly available data from the Next-Generation Simulation (NGSIM) dataset and OpenACC for our project. The code for the simulation component is accessible at https://github.com/Keke-Long/PERL.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

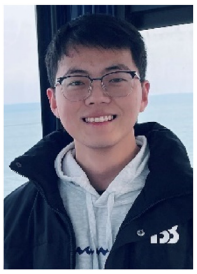
Acknowledgements

This work was supported by the National Science Foundation Cyber-Physical Systems (CPS) program (No. 2343167).

References

Bates, I.W., Karimoddini, A., Karimadini, M., 2024. A learning-based approach for diagnosis and diagnosability of unknown discrete event systems. IEEE Transact. Neural Networks Learn. Syst. 35, 5421–5434.
Brunton, S.L., Proctor, J.L., Kutz, J.N., 2016. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. Proc. Natl. Acad. Sci. USA. 113, 3932–3937.
De Miguel, M.A., Armingol, J.M., Garcia, F., 2022. Vehicles trajectory prediction using recurrent VAE network. IEEE Access 10, 32742–32749.
Dong, S., Zhou, Y., Chen, T., Li, S., Gao, Q., Ran, B., 2021. An integrated Empirical Mode Decomposition and Butterworth filter based vehicle trajectory reconstruction method. Phys. Stat. Mech. Appl. 583, 126295.
Duret, A., Ahn, S., Buisson, C., 2011. Passing rates to measure relaxation and impact of lane-changing in congestion. Comput. Aided Civ. Infrastruct. Eng. 26, 285–297.
Gulshan, V., Peng, L., Coram, M., Stumpe, M.C., Wu, D., Narayanaswamy, A., et al., 2016. Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. JAMA 316, 2402–2410.
Han, Y., Wang, M., Leclercq, L., 2023. Leveraging reinforcement learning for dynamic traffic control: a survey and challenges for field implementation. Commun. Transp. Res. 3, 100104.
Hornik, K., Stinchcombe, M., White, H., 1989. Multilayer feedforward networks are universal approximators. Neural Netw. 2, 359–366.
Ivanovic, B., Leung, K., Schmerling, E., Pavone, M., 2021. Multimodal deep generative models for trajectory prediction: a conditional variational autoencoder approach. IEEE Rob. Autom. Lett. 6, 295–302.
Jiang, R., Wu, Q., Zhu, Z., 2001. Full velocity difference model for a car-following theory. Phys. Rev. E 64, 017101.
Karniadakis, G.E., Kevrekidis, I.G., Lu, L., Perdikaris, P., Wang, S., Yang, L., 2021. Physics-informed machine learning. Nat. Rev. Phys. 3, 422–440.
Krishnapriyan, A., Gholami, A., Zhe, S., Kirby, R.M., Mahoney, M.W., 2021. Characterizing possible failure modes in physics-informed neural networks. In: Advances in Neural Information Processing Systems, pp. 26548–26560.
Laval, J.A., Leclercq, L., 2010. A mechanism to describe the formation and propagation of stop-and-go waves in congested freeway traffic. Philos. Trans. A Math. Phys. Eng. Sci. 368, 4519–4541.
Lee, N., Choi, W., Vernaza, P., Choy, C.B., Torr, P.H.S., Chandraker, M., 2017. DESIRE: distant future prediction in dynamic scenes with interacting agents. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 336–345.
Li, J., Xie, N., Zhang, K., Guo, F., Hu, S., Chen, X., 2022. Network-scale traffic prediction *via* knowledge transfer and regional MFD analysis. Transport. Res. C Emerg. Technol. 141, 103719.
Li, W., Rios-Torres, J., Wang, B., Khattak, Z.H., 2024. Experimental assessment of communication delay's impact on connected automated vehicle speed volatility and energy consumption. Commun. Transp. Res. 4, 100136.
Long, K., Shi, H., Chen, Z., Liang, Z., Li, X., de Souza, F., 2024a. Bi-scale car-following model calibration based on corridor-level trajectory. Transp. Res. Part E Logist. Transp. Rev. 186, 103497.
Long, K., Shi, X., Li, X., 2024b. Physics-informed neural network for cross-dynamics vehicle trajectory stitching. Transp. Res. Part E Logist. Transp. Rev. 192, 103799.
Long, K., Liang, Z., Shi, H., Shi, L., Chen, S., Li, X., 2024c. Traffic oscillation mitigation with physics-enhanced residual learning (PERL)-based predictive control. Commun. Transp. Res. 4, 100154.
Makridis, M., Mattas, K., Anesiadou, A., Ciuffo, B., 2021. An open database of car-following experiments to study the properties of commercial ACC systems. Transport. Res. C Emerg. Technol. 125, 103047.
Mo, Z., Shi, R., Di, X., 2021. A physics-informed deep learning paradigm for car-following models. Transport. Res. C Emerg. Technol. 130, 103240.
Möllerstedt, V.E., Russo, A., Bouton, M., 2024. Model based residual policy learning with applications to antenna control. In: 2024 IEEE International Conference on Machine Learning for Communication and Networking (ICMLCN), pp. 1–21.
Naing, H., Cai, W., Nan, H., Wu, T., Liang, Y., 2022. Dynamic data-driven microscopic traffic simulation using jointly trained physics-guided long short-term memory. ACM Trans. Model Comput. Simulat 32, 1–27.
Pereira, M., Lang, A., Kulcsár, B., 2022. Short-term traffic prediction using physics-aware neural networks. Transport. Res. C Emerg. Technol. 142, 103772.
Qu, X., Lin, H., Liu, Y., 2023. Envisioning the future of transportation: inspiration of ChatGPT and large models. Commun. Transp. Res. 3, 100103.
Salman, A.G., Kanigoro, B., Heryadi, Y., 2015. Weather forecasting using deep learning techniques. In: 2015 International Conference on Advanced Computer Science and Information Systems (ICACSIS), pp. 281–285.
Shan, T., Zeng, J., Song, X., Guo, R., Li, M., Yang, F., et al., 2023. Physics-informed supervised residual learning for electromagnetic modeling. IEEE Trans. Antenn. Propag. 71, 3393–3407.
Silver, T., Allen, K.R., Tenenbaum, J., Kaelbling, L., 2018. Residual Policy Learning arXiv preprint arXiv:1812.06298.
Treiber, M., Hennecke, A., Helbing, D., 2000. Congested traffic states in empirical observations and microscopic simulations. Phys. Rev. E Stat. Phys. Plasmas Fluids Relat. Interdiscip. Topics 62, 1805–1824.
Wang, S., Teng, Y., Perdikaris, P., 2021. Understanding and mitigating gradient flow pathologies in physics-informed neural networks. SIAM J. Sci. Comput. 43, A3055–A3081.
Wang, S., Yu, X., Perdikaris, P., 2022. When and why PINNs fail to train: a neural tangent kernel perspective. J. Comput. Phys. 449, 110768.
Wang, X., Jiang, R., Li, L., Lin, Y., Zheng, X., Wang, F.Y., 2018. Capturing car-following behaviors by deep learning. IEEE Trans. Intell. Transport. Syst. 19, 910–920.
Xu, M., Di, Y., Ding, H., Zhu, Z., Chen, X., Yang, H., 2023. AGNP: network-wide short-term probabilistic traffic speed prediction and imputation. Commun. Transp. Res. 3, 100099.
Yao, H., Li, X., Yang, X., 2023. Physics-aware learning-based vehicle trajectory prediction of congested traffic in a connected vehicle environment. IEEE Trans. Veh. Technol. 72, 102–112.
Yao, Z., Hu, R., Wang, Y., Jiang, Y., Ran, B., Chen, Y., 2019. Stability analysis and the fundamental diagram for mixed connected automated and human-driven vehicles. Phys. Stat. Mech. Appl. 533, 121931.
Zhang, J., Mao, S., Yang, L., Ma, W., Li, S., Gao, Z., 2024. Physics-informed deep learning for traffic state estimation based on the traffic flow model and computational graph method. Inf. Fusion 101, 101971.
Zhong, C., Wu, P., Zhang, Q., Ma, Z., 2023. Online prediction of network-level public transport demand based on principle component analysis. Commun. Transp. Res. 3, 100093.
Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., et al., 2021. Informer: beyond efficient transformer for long sequence time-series forecasting. Proc. AAAI Conf. Artif. Intell. 35, 11106–11115.

Keke Long is a Research Associate in the Department of Civil and Environmental Engineering at the University of Wisconsin–Madison. She received her Ph.D. degree in civil and environmental engineering from the University of Wisconsin–Madison in 2024. She earned her M.S. degree in traffic engineering from Tongji University, Shanghai, China, in 2021, and her B.S. degree in traffic engineering from Chang'an University, Xi'an, China, in 2018. Her primary research interests include the application of Artificial Intelligence (AI) in transportation and the evaluation of automated vehicles.



Zihao Sheng is a Research Assistant in the Department of Civil and Environmental Engineering at the University of Wisconsin–Madison. He obtained his B.S. degree in Xi'an Jiaotong University, in 2019 and his M.S. degree in Shanghai Jiao Tong University in 2022. His main research interests are human-centered AI, autonomous driving, robotics, physics, and intelligent transportation.

Haotian Shi is an incoming Associate Professor at Tongji University. He serves as a Research Associate at University of Wisconsin–Madison. He received his Ph.D. degree in civil and environmental engineering from the University of Wisconsin–Madison in 2023. He also received three M.S. degrees in power and machinery engineering (Tianjin University, 2020), civil and environmental engineering (University of Wisconsin–Madison, 2020), and computer sciences (University of Wisconsin–Madison, 2022). His main research directions are connected and automated vehicles, intelligent transportation systems, and Artificial Intelligence.

Xiaopeng (Shaw) Li is currently a Professor in the Department of Civil and Environmental Engineering and an affiliate in the Department of Electrical and Computer Engineering at the University of Wisconsin–Madison. He served as the Director of National Institute for Congestion Reduction (NICR) before. He is a recipient of a National Science Foundation (NSF) CAREER award. He has published over 110 peer-reviewed journal papers. He has served as the PI or a co-PI for a number of federal, state, and industry grants, with a total budget of around $30 million. His major research interests include automation, connectivity, and sensing in transportation and related systems. He received his B.S. degree (2006) in civil engineering with a minor in computer engineering from Tsinghua University, China, his M.S. degree (2007), and his Ph.D. (2011) degree in civil engineering along with an M.S. degree (2010) in applied mathematics from the University of Illinois at Urban-Champaign, USA.

Sikai (Sky) Chen is an Assistant Professor at the Department of Civil and Environmental Engineering and the Department of Mechanical Engineering (courtesy), University of Wisconsin–Madison. He received his Ph.D. degree in civil engineering with a focus on computational science & engineering from Purdue University in 2019. His research focuses on three major themes: human users, AI, and transportation. He aims to innovate and develop safe, efficient, sustainable, and human-centered transportation systems by cutting-edge methods and technologies. The focus is on incorporating human behaviors, interactive autonomy, digital infrastructure, and intelligent control frameworks. In addition, he is a member of two ASCE national committees: Connected & Autonomous Vehicle Impacts, and Economics & Finance; IEEE Emerging Transportation Technology Testing Technical Committee; and TRB Standing Committee on Statistical Methods (AED60).

Soyoung Ahn is a Professor in the Department of Civil and Environmental Engineering at the University of Wisconsin–Madison. She received her Ph.D. degree (2005) in civil and environmental engineering from the University of California, Berkeley. Her research and teaching interests are in traffic flow theory and operations, Intelligent Transportation Systems applications, and traffic operational impacts on environment and safety. Her research goals are to (1) better understand the fundamental nature of traffic flow through observation, experimentation and the application of quantitative (statistical) methods, (2) examine the environmental and safety impact of traffic flow phenomena, and (3) apply the knowledge in developing traffic theories and control strategies through applications of ITS.