Communications and Letters

IDM-Follower: A Model-Informed Deep Learning Method for Car-Following Trajectory Prediction

Yilin Wang, *Student Member, IEEE*, and Yiheng Feng, *Member, IEEE*

***Abstract*—Model-based and learning-based methods are two main approaches modeling car-following behaviors. To combine advantages from both types of models, this study introduces a novel approach, IDM-Follower, which generates a sequence of the following vehicle's trajectory using a recurrent autoencoder informed by a physical car-following model, the Intelligent Driving Model (IDM). We design an innovative neural network (NN) structure with two independent encoders and an attention-based decoder to predict the trajectory sequence. The loss function accounts for discrepancies from both the physical car-following model and the NN predictions. Numerical experiments are conducted using simulated and real world (i.e., NGSIM) datasets under different data noise levels with varying weights between the learning loss and the model loss. Testing results show the proposed approach outperforms both model-based and learning-based baselines under real and high noise levels. The optimal integrating weight between the model and learning component is significantly influenced by data quality, which affects both prediction accuracy and safety metrics.**

***Index Terms*—Model-informed machine learning, car following model, deep learning, attention.**

I. Introduction

PREDICTING driving behavior is critical for many transportation applications. Car-following is the most common driving behaviors which attracts tremendous research interests [1], [2]. As the edge computing and V2X communication technologies advance, incorporating real-time prediction into intelligent vehicle planning and transportation system management is becoming more popular [3]. Numerous car-following models have been developed to describe human driving behaviors considering the contextual factors related to physics, road geometry, and driving parameters [4].

There are two major methods in modeling car-following behaviors. Model-based methods describe car-following behaviors with explicit mathematical equations while learning-based methods usually mapping leading-following vehicle relations by neural networks (NN). Both model-based and learning-based approaches offer unique advantages and drawbacks. Characterized by their explicit structures and limited parameters, model-based methods align well with aggregated driving patterns. For example, Newell's first-order car following model [5] is one of the most widely used car-following models in transportation literature. However, model-based methods don't consider heterogeneity in driving behaviors, thus may perform poorly in cases with atypical following behavior and/or noisy data. On the other hand, learning-based methods can handle heterogeneous driving behaviors with NN trained by a large amount of data. With more and more sensors being installed on both smart vehicles and infrastructure, acquiring enough training data becomes feasible. However, lacking of interpretability is the major shortcoming of (deep) learning-based methods since the mapping relations within the NN is difficult to explain. Moreover, when the training data set is noisy or deviates from the ground truth, learning-based methods may perform poorly. To leverage the advantages of both methods, physics or model informed machine learning methods have been proposed in the last few years. This new type of methods integrate physical models with NN and outperform both model-based and learning-based methods [6], [7], [8].

In this paper, we propose a model-informed deep learning method, called IDM-Follower, to predict long sequence car-following trajectories. Particularly, we integrate the intelligent driving model (IDM) with a sequence-to-sequence nerual network structure using two encoders and one attention-based decoder. Position and velocity sequences are encoded separately to generate the following vehicle trajectory sequence. We design a hybrid loss function that integrates the calibrated IDM model into the NN. Both simulated and real-world trajectories are tested under multiple levels of GPS noises. Results show that the proposed IDM-Follower has promising performance in term of robustness against GPS noises and outperforms several baseline models. The contributions of this paper are summarized as below: 1) We introduce a sequence-to-sequence model-informed prediction framework that integrates a conventional car-following model with a customized NN. 2) We analyze the optimal weights between the model loss and learning loss under different noise levels. and 3) Our proposed model exhibits robustness against GPS noises, making it suitable for a wide range of field applications.

Manuscript received 4 February 2024; accepted 8 February 2024. Date of publication 20 February 2024; date of current version 9 August 2024. This work was supported in part by the U.S. National Science Foundation under Grant CMMI-2038215 and in part by the United States Department of Transportation (USDOT) under Grant 693JJ32150006. *(Corresponding author: Yiheng Feng.)*

The authors are with the Lyles School of Civil Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: feng333@purdue.edu; wang4517@purdue.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TIV.2024.3367654.

Digital Object Identifier 10.1109/TIV.2024.3367654

II. Literature Review

Early car-following models aim to capture the dynamics of driver behaviors by focusing on how drivers adapt their speeds and headways in response to the actions of preceding vehicles such as General Motors (GM) Model [9], and Gipps Model [10]. Recent notable car-following model, such as Intelligent Driving Model (IDM) [11], exemplifies a time-continuous approach that incorporates a driver's preferred speed, along with their acceleration and deceleration behaviors. Its design is geared towards emulating naturalistic driving patterns and adeptly simulating diverse traffic conditions, including stop-and-go waves. Nevertheless, the IDM's reliance on simplified assumptions and a constrained set of parameters often hampers its ability to accurately reflect the complexities inherent in real-world driving behaviors even though there are a few researches calibrates the model by various methods [12], [13]. This limitation not only leads to prediction errors but also renders the model particularly sensitive to parameter calibration, thereby complicating its validation and fine-tuning processes.

Recent studies have witnessed the emergence of various learning-based methods, attributing to their proficiency in identifying complex patterns within large-scale datasets. Focusing on the application of NNs for predicting car-following driving behaviors, Jia et al. [14] developed a four-layer NN that processes inputs such as relative speed, follower speed, maximum desired speed, and the gap between two vehicles to predict the following vehicle's subsequent acceleration. Zhou et al. [15] employed a basic recurrent neural network (RNN) to model drivers' car-following behavior, showing enhanced performance compared to the IDM [11]. However, conventional RNNs face challenges in handling long-range dependencies due to the gradient vanishing problem, which arises when training gradients become excessively small or large. To overcome these limitations, Huang et al. [16] utilized Long Short-Term Memory (LSTM) networks, which captures realistic driving patterns by integrating data from surrounding vehicles. Zhang et al. [17] implemented an LSTM architecture, named HRC LSTM, that processes 2-D positions (longitudinal and lateral) of nearby vehicles within a detection area. Both LSTM-based models have been validated using the NGSIM dataset. Lately, Ma and Qu [18] proposed a sequence-to-sequence model using a LSTM network for car-following predictions. Zhu et al. [19] proposed a long-sequence trajectory prediction model considering historical context by attention mechanisms. The proposed transformer NN takes the first 4 s car-following trajectories as input and predict the next 12 s. Zhang et al. [20] proposed an attention-based interaction-aware trajectory prediction model with an encoder-decoder structure with graph attention networks.

Model-informed learning approaches integrate traditional car-following models with machine learning techniques to leverage the strengths of both model and learning based approaches. By incorporating physical models into machine learning frameworks, the model-informed methods can outperform both classical and learning-based models. Yuan et al. [8] proposed a stochastic physics regularized Gaussian process (PRGP) model for macroscopic traffic state estimation. Taking Gaussian process as the learning model, this study implemented a mixed structure of learning framework along with a new loss function including both physical and learning-based losses. The proposed PRGP framework was tested on I-15 Highway data in Utah. The results showed that the PRGP could improve the prediction accuracy with a properly selected physics model. On the other hand, performance would get worse if an inappropriate physics model was integrated. Similarly, Shi et al. [7] and Huang [6] proposed physics-informed deep learning (PIDL) frameworks that integrate NNs with the Lighthill Whitham-Richards(LWR) model to construct fundamental diagrams of a highway segment. Similar lost functions are designed that combine data and physical losses. A recent study [21] tried to apply the same framework that integrated NNs with car-following models. Two representative car-following models, namely IDM and optimal velocity model (OVM) are integrated with a NN to predict the acceleration of the following vehicle based on the velocity and gap between the leading and following vehicles. Moreover, a novel structure was developed that could simultaneously calibrate the car-following model parameters and predict the following vehicle's acceleration. However, this framework can only predict step by step, rather than an entire sequence simultaneously. Inspired by previous studies, this paper proposes a deep learning-based sequence-to-sequence vehicle trajectory prediction method integrated with the IDM car-following model, named IDM-Follower.

III. Methodology

A. Problem Statement

Define $X^{(i)}$ and $Y^{(i)}$ representing the trajectories for the leading and following vehicles respectively. The leading vehicle's trajectory consists of two time-series sequence that represent position ($S_X^{(i)}$) and velocity ($V_X^{(i)}$) while the following vehicle's trajectory includes only the position vector ($S_Y^{(i)}$). For each sequence, t is used as the time index and T is the total time steps (i.e., trajectory length).

Further, we define the NN-based car-following model $F_\theta : X^{(i)} \rightarrow \tilde{Y}_\theta^{(i)}$ parameterized by θ. Note that we denote the predicted quantities with a tilde, to distinguish them from observed values. Similarly, we define the model-based model $F_\lambda : X^{(i)} \rightarrow \tilde{Y}_\lambda^{(i)}$ with parameter λ. The function F_θ is represented by a NN, while the function F_λ is a set of physics-based equations. Then IDM-Follower, the model-informed car-following trajectory predictor, can be defined as $F_{\theta,\lambda^*} : X^{(i)} \rightarrow \tilde{Y}_{\theta,\lambda^*}^{(i)}, \forall i = 1, \ldots, N$. The goal is to train an optimal parameter set θ^* for the NN with regularization from the physical car-following model with optimal parameters λ^* which is pre-calibrated from historical observations.

Equations (1) to (3) explain the formulations of the learning component and model component. The learning component, implemented through a NN, produces a time-series position vector to delineate the following trajectory. Conversely, the output of the model component is an acceleration vector at each

time step. This acceleration vector is subsequently converted into a time-series position vector, the same as the output of the learning component, to ensure uniformity in representation.

Learning-based component:

$$F_\theta\left(X^{(i)}\right) = \tilde{S}^{(i)}_{Y,\theta} \quad \forall i = 1, \ldots, N \tag{1}$$

Model-based component:

$$F_\lambda\left(s^i_X(t), v^i_X(t), s^i_Y(t)\right) = \tilde{a}^{(i)}_Y(t) \quad \forall i = 1, \ldots, N \tag{2}$$

$$\tilde{S}^{(i)}_{Y,\lambda} = \left\{ \iint_0^t \tilde{a}^{(i)}_Y(m)\, dm \quad \forall t = 1, \ldots, T \right\} \tag{3}$$

Given the definitions, the model-informed trajectory prediction task can be formulated as Equation (4).

$$\min_\theta \sum_{i=1}^{N} \left\| \tilde{S}^{(i)}_{Y,\theta,\lambda^*} - S^{(i)}_Y \right\|_2 \tag{4}$$

$$\text{s.t.} \quad \tilde{S}^{(i)}_{Y,\theta,\lambda^*} = F_{\theta,\lambda^*}(X^{(i)}), \quad \forall i = 1, \ldots, N$$

B. Neural Network Architecture

Standard RNNs process sequential data by factoring computation processes with symbolic positions from input and output sequences, generating a series of hidden states h_t as a function of the previous hidden state $h_{(t-1)}$ and the input at t. However, this approach poses challenges for long sequential data, such as vehicle trajectories, due to the decay of model memory in lengthy historical contexts. To address the issue, we introduce an encoder-decoder architecture to extract hidden features from the input trajectory, thereby generating a latent vector that encapsulates the complex information within. This architecture allows the decoder to reconstruct the output data based on this latent vector. Within our proposed model, the encoder initially expands the input states, specifically the leading vehicle's position and speed, into a hidden layer. This is a critical step where sequence features are extracted and stored. To account for different scales in position and velocity sequences, two separate encoders are employed to encode the leading vehicle's position and velocity information, respectively. This approach ensures precision, as each encoder independently handles distinct data types. Subsequently, the decoder, integrating two hidden layers from these encoders, reconstructs the following vehicle's trajectory. The attention layer in the decoder leverages outputs from both encoders to generate a context vector. This vector signifies the attention weights between the two encoded latent vectors. Following this, the position sequence decoder processes the latent vector received from the encoder, producing the following trajectory while incorporating the context vector from the attention layer. To provide a comprehensive understanding, the NN's structure is depicted in Fig. 1, with its parameters detailed below.

Encoder: LSTM (input size=1, hidden size=128, layers=2) key network: Linear (256, 128) value network: Linear (256, 128)

Decoder: LSTM (input size=128, hidden size=128, layers=1) Attention (key size = 128, value size = 128) output layer: Linear (256, 1)

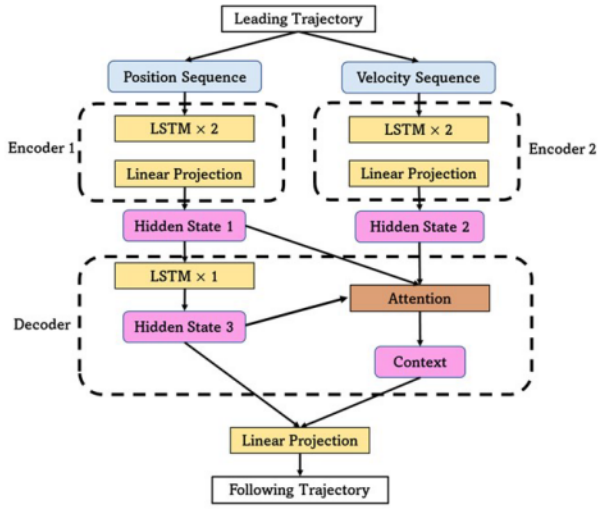


Fig. 1. Architecture of IDM-follower.

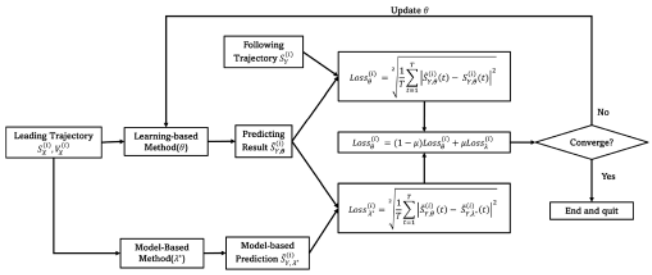


Fig. 2. Model-informed training workflow.

C. Model-Informed Car-Following Trajectory Prediction

Fig. 2 shows the work flow for the model-informed learning framework. The inputs include the leading trajectory as well as a calibrated IDM model with optimal parameter set λ^* as the informed car-following model. A loss function with two components is utilized to compute gradient during the training process. The loss function is shown in Equation (5). The first component, $\text{Loss}^{(i)}_\theta$, evaluates the discrepancies between the NN predictions and the observed following vehicle trajectory, while the second component, $\text{Loss}^{(i)}_{\lambda^*}$, assesses the differences between the NN predictions and IDM model predictions. These two components are linearly combined with a parameter μ. The combined loss function's gradient is then calculated and employed to update the NN parameters θ.

$$\text{Loss}^{(i)}_{\theta,\lambda^*} = (1-\mu)\text{Loss}^{(i)}_\theta + \mu\text{Loss}^{(i)}_{\lambda^*} \tag{5}$$

$$\text{Loss}^{(i)}_\theta = \sqrt{\frac{1}{T}\sum\nolimits_{t=1}^{T} |\tilde{S}^{(i)}_{Y,\theta,\lambda^*}(t) - S^{(i)}_Y(t)|^2} \tag{6}$$

$$\text{Loss}^{(i)}_{\lambda^*} = \sqrt{\frac{1}{T}\sum\nolimits_{t=1}^{T} |\tilde{S}^{(i)}_{Y,\theta,\lambda^*}(t) - \tilde{S}^{(i)}_{Y,\lambda^*}(t)|^2} \tag{7}$$

During the training stage, the following trajectories of the model component (denoted as $\tilde{S}^{(i)}_{Y,\lambda^*}$) is generated by a calibrated IDM model using the ground truth leading vehicle trajectories.

TABLE I
AVERAGE DISPLACEMENT FOR THREE LEVELS OF GPS NOISE(M)

Noise level	NGSIM	SUMO
None	0	0
Real	1.79	1.82
High	5.63	5.66

For the learning component, noisy leading vehicle trajectories are used as training data. When the training is complete, only noisy leading vehicle trajectories are needed in the execution/testing step. This framework is versatile and can be applied to any NN structures and car-following models.

IV. CASE STUDY

To evaluate the efficacy of the model-informed framework, we conduct a series of numerical experiments using both simulation and real-world datasets. This section outlines the experimental setup, including data processing, noise generation, and results analysis. The simulated trajectories are generated by Simulation of Urban MObility (SUMO), whereas the real-world trajectory data is extracted from the Next Generation Simulation (NGSIM) Lankershim Blvd dataset. A few benchmark models are also implemented for performance comparison. Furthermore, various levels of GPS noises are incorporated in the training data, with the intention to mimic real-world GPS data qualities.

A. GPS Noise Generation

GPS inaccuracies arise from various factors such as signal arrival, ionospheric effects, ephemeris impacts, and multipath distortions [22]. It is important that the proposed method is robust against GPS noises. We adopted the GPS error model developed by [23] to add offsets to both simulation and NGSIM data. In the model, latitude and longitude GPS points are independently calibrated with two univariate autoregressive integrated moving average (ARIMA) models respectively. Since this study mainly considers one dimensional driving behavior, only longitudinal GPS error is added. Three levels of GPS noises are considered in the case study. The first level represents no GPS noise (i.e., none). The second level represents real-world GPS noises collected from [23] (i.e., real), while the third level considers high noises (i.e., high.) The average noises under different levels for the two datasets are shown in Table I.

B. Dataset Processing and Training

A single-lane road with constant flow rate is created in SUMO to generate the simulation dataset. A fixed-time traffic signal is placed to create varying driving behaviors. Since SUMO vehicle trajectories are generated by the IDM car-following model, the optimal model parameters can be directly obtained from the simulation environment. The obtained parameters are displayed in the first row of Table II. As a result, the IDM model used for SUMO data can be considered as "perfect". On the other hand, the NGSIM dataset includes bidirectional trajectories of three to four lanes in a roughly 1600-foot street, encompassing three signalized intersections. The dataset contains 30 minutes of raw data. A Kalman Filter [24] is first applied to remove noise caused by video processing. The output data is treated as the ground truth trajectories. Unlike simulation data where the optimal IDM model parameters can be obtained directly from SUMO, the IDM parameters used for the NGSIM dataset require calibration. We utilize the calibration results from an existing study [12], shown in the second row of Table II. We further validate the parameters by calculating the average final displacement error (FDE) in the processed trajectory pairs. The average FDE is 3.01 m, which is deemed within an acceptable error range.

TABLE II
OPTIMAL IDM PARAMETERS FOR DIFFERENT DATASETS (λ^*)

Dataset	v_0	T (s)	s_0	a	b	δ	MAE (m)
SUMO	16.7	1	2.5	3	4.5	4	1.98
NGSIM[12]	15.97	1.3	1.57	2.49	2.39	4	3.01

TABLE III
TRAINING PARAMETERS AND DETAILED SETTINGS

Hyperparameter Name	SUMO	NGSIM
Max Epoch	200	300
Optimizer	Adam	Adam
Learning Rate	0.001	0.001
Learning Rate Weight Decay	1.0×10^{-5}	1.0×10^{-5}
Batch Size	128	64
Train, Test and Validate	(4471, 1788, 2682)	(1730, 494, 741)

Considering 50 m as the maximum car-following distance, we extract car-following trajectory pair with a resolution of 0.1 s with a 8 s length. The leading vehicle's trajectory is recorded as a position-time and velocity-time sequence while the following vehicle's trajectory is processed to be a position-time sequence. Three levels of GPS noises are added to the position sequences. The SUMO dataset and NGSIM dataset have 8942 and 2972 data points respectively. The datasets are divided into training, validation, and testing sets with ratios of 0.5, 0.2, and 0.3, respectively. Due to the differences in sizes and data patterns, different training parameters are applied fodr different datasets. The details are shown in Table III.

C. Results and Analysis

1) Prediction Performance Testing: Three baseline models are used for comparison: LSTM_AE, Transformer, and a calibrated IDM. Fig. 3 illustrates the architectures of the first two baseline models. The LSTM_AE model employs identical NN layers for both the encoder and decoder. The Transformer model seperately encode leading position and velocity as source and target sequence with identical encoder structure resepectively and output the following trajectory. Mean Average Error (MAE) is used to assess the prediction performance.

Table IV shows the prediction performance from these models under the three GPS noises levels. Interestingly, the LSTM_AE model exhibits superior performance in both datasets when there is no GPS noise. It means classic seq-to-seq model is sufficient to map the leading-following trajectory relation with clean data (i.e., The NGSIM data after Kalman filter), and simplified driving patterns (i.e., the same car-following behavior

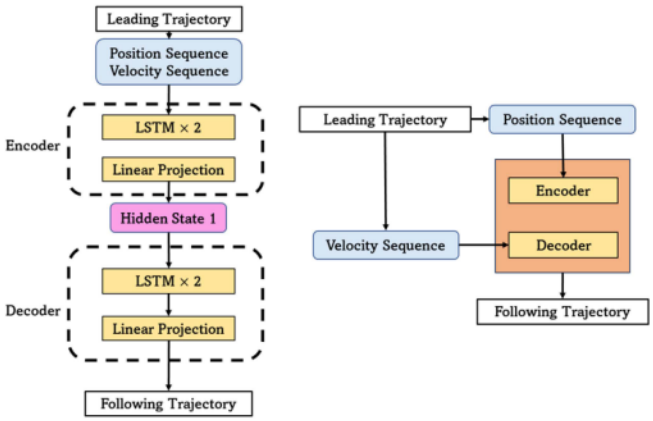


Fig. 3. Baseline model architecture: LSTM_AE (left) Transformer_baseline (right).

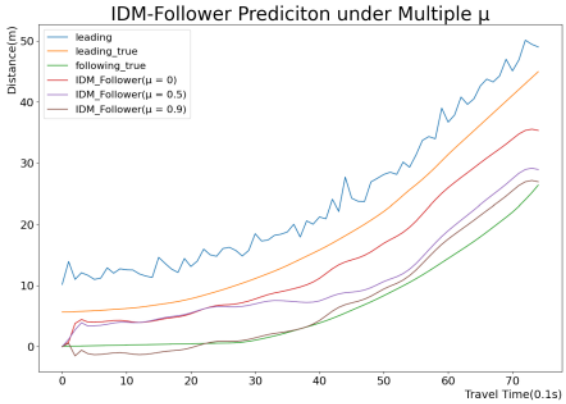


Fig. 4. Predictions from IDM-follower different integration weight (μ) in NGSIM dataset (high noise).

TABLE IV
PREDICTION PERFORMANCE FOR TWO PROPOSED DATASET (MAE: M)

	Testing Model	None	Real	High
SUMO	IDM	1.98	4.41	5.40
	LSTM_AE	**1.60**	3.00	6.52
	Transformer	3.76	4.17	5.67
	IDM-Follower	1.72	**2.17**	**2.37**
NGSIM	IDM	2.29	3.99	6.63
	LSTM_AE	**2.27**	6.62	5.94
	Transformer	4.15	4.26	6.44
	IDM-Follower	2.62	**3.08**	**3.44**

generated from SUMO). However, when GPS noises are introduced as in real-world situations, the proposed IDM-Follower surpasses all baseline models, particularly under high noise levels. This is mainly because the additional GPS displacements influence the data and disrupt the patterns between the gap and velocity. Additionally, the Transformer baseline is more sensitive to GPS noises. Therefore, the baseline models may struggle to accurately capture the coupling errors patterns from both heterogeneous driving behaviors and GPS errors without the assistance from the IDM model. With a properly designed model-informed framework, the performance becomes more robust and can better handle noisy inputs. To further evaluate the influence of the model component in the model-informed framework, we analyze the optimal weight (μ^*) under different noise levels.

2) Analysis on Optimal Integration Ratio: In the loss function (5), μ dictates how much information from the physical model is informed. To analyze the sensitivity on μ, we vary the value of μ from 0 to 0.9 with a step of 0.1 with ten experiments using the same datasets. The results from SUMO and NGSIM datasets are presented in Tables V and VI respectively.

Results indicate that the optimal weight varies with different noise levels. Under none-noise level, the IDM-Follower tends to rely more on the observed data (i.e., lower μ value) since the observed data matches the patterns of real driving behaviors pretty well. However, the optimal μ value increases as the increase of the noise level, which shows a consistent pattern in both datasets. This is because higher noise will downgrade the quality of the data by adding random fluctuation in the training dataset, which deviates from normal car-following behaviors. As a result, more weights from the model component (i.e., higher μ value) can help regularize the training process and improve the prediction results.

Fig. 4 illustrates an example of the IDM-Follower predictions with different weights μ under high GPS noise levels, using the NGSIM dataset. The blue curve represents the leading vehicle trajectory with GPS noises while the orange curve and the green curve represent the "clean" trajectories of the leading and following vehicle respectively (i.e., ground truth). The prediction results of different weights (μ) are shown in the red, purple and brown curves. When $\mu = 0$, only data loss is considered in the training, and the IDM-follower becomes a pure data-driven approach. The predicted trajectory exhibits a constant displacement to the ground truth following vehicle trajectory, similar as the constant displacement between the noisy and clean leading vehicle trajectories. This case indicates that the pure data-driven method may carry the data noise to the prediction results. Moreover, at the beginning of the prediction horizon, the predicted trajectory even intersects with the leading vehicle trajectory, which violates basic car-following rules and physical laws. With the increase values of μ, the predicted following vehicle trajectories are "dragged" towards the ground truth following trajectory due to the integration of the model loss. Combining optimal weight values from Table VI under different noise levels, the results provide insights on selecting the best hyperparameter μ: the more noises exist in the data, the higher weights of the model component should be considered in the training. Note that this conclusion is only true when the integrated model can accurately represent the driving behavior. If an incorrect or poor performance model is integrated, the performance of the model-informed framework may even get worse, as shown in the previous study [8].

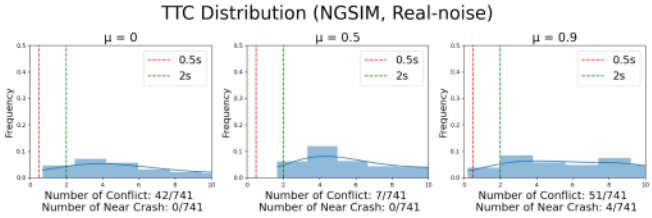
A further investigation on Time to Collision (TTC) is conduct to evaluate the safety performance of the IDM-Follower. The minimum TTC of the predicted following trajectory is used as the metric, as it indicates the most dangerous moment of the entire trajectory. Further, we only consider the trajectories that have minimum TTC smaller than 10 s as shown in Equation (8). The minimum TTC distributions under varying integration

TABLE V
PREDICTION ERROR UNDER DIFFERENT WEIGHTS (μ) IN SUMO DATASET

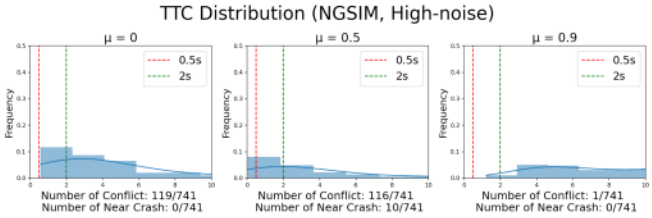
Noise level	μ=0	μ=0.1	μ=0.2	μ=0.3	μ=0.4	μ=0.5	μ=0.6	μ=0.7	μ=0.8	μ=0.9
None	1.80	1.78	**1.72**	1.80	1.84	1.80	1.89	2.21	2.04	2.16
Real	3.37	3.04	2.84	2.81	2.32	2.28	**2.17**	2.22	2.25	2.26
High	6.62	6.22	5.69	4.92	4.28	3.70	3.16	2.59	2.46	**2.37**

TABLE VI
PREDICTION ERROR UNDER DIFFERENT WEIGHTS (μ) IN NGSIM DATASET

Noise level	μ=0	μ=0.1	μ=0.2	μ=0.3	μ=0.4	μ=0.5	μ=0.6	μ=0.7	μ=0.8	μ=0.9
None	2.89	**2.62**	2.74	2.83	2.80	2.98	3.18	3.60	3.48	3.76
Real	3.35	3.42	3.36	3.39	3.31	**3.08**	3.19	3.56	3.44	3.17
High	6.77	5.43	5.29	5.11	4.57	4.20	4.29	3.82	3.68	**3.44**



(a) TTC Distribution for NGSIM Dataset under Real GPS Noise



(b) TTC Distribution for NGSIM Dataset under High GPS Noise

Fig. 5. Time-to-collision (TTC) distributions for NGSIM dataset with various GPS noise.

weights and noise levels from the NGSIM dataset are shown in Fig. 5. Two thresholds are defined. One threshold of two seconds (the green dotted line) is employed to identify conflicts, while a more stringent criterion of 0.5 seconds is used as the threshold for near-crash scenarios (the red dotted line). The number of conflicts and number of near crash scenarios are calculated and shown in Fig. 5. It can be seen that under the real noise level, $\mu = 0.5$ results in the lowest number of conflicts and near crash scenarios. Under the high noise level, $\mu = 0.9$ results in the lowest number. This result is consistent with the prediction accuracy and suggests that an optimal integration weight can also enhance safety performance of the predicted trajectories by mitigating unsafe driving behaviors.

$$\mathrm{TTC}_{min}^{(i)} = \min\left\{ \min_{t} \frac{\left|S_X^{(i)}(t) - S_Y^{(i)}(t)\right|}{\left|v_X^{(i)}(t) - v_Y^{(i)}(t)\right|}, \quad 10 \right\} \tag{8}$$

V. CONCLUSION AND DISCUSSION

In this study, we introduced a model-informed deep learning framework called IDM-Follower to predict sequential car-following trajectories with noisy observations. Employing a sequence-to-sequence model architecture that comprises multiple encoders, which take position and velocity information as inputs, along with an attention-based decoder, the IDM-Follower was trained with a composite loss function that integrated both data-driven and classical IDM information. Simulation and real-world datasets under varying noise levels were utilized for testing. Results indicated that the IDM-Follower outperformed two baseline learning models and pure IDM under noisy data. The study shows that choosing a proper integration weight on physical models could improved the model performance from both accuracy and safety perspective. Moreover, this study also demonstrates the potential for incorporating physical models into a sequence-to-sequence prediction model. One limitation of this study is that it does not consider lane changing behaviors. Our future work includes building a new prediction framework that accommodates multi-lane scenarios as well as theoretical analysis on choosing optimal weights between the model-based component and learning-based component.

REFERENCES

[1] S. Mozaffari, E. Arnold, M. Dianati, and S. Fallah, "Early lane change prediction for automated driving systems using multi-task attention-based convolutional neural networks," *IEEE Trans. Intell. Veh.*, vol. 7, no. 3, pp. 758–770, Sep. 2022.

[2] R. Wang, L. Tang, Y. Yang, S. Wang, M. Tan, and C.-Z. Xu, "Adaptive trajectory tracking control with novel heading angle and velocity compensation for autonomous underwater vehicles," *IEEE Trans. Intell. Veh.*, vol. 8, no. 3, pp. 2135–2147, Mar. 2023.

[3] Z. Hu, S. Lou, Y. Xing, X. Wang, D. Cao, and C. Lv, "Review and perspectives on driver digital twin and its enabling technologies for intelligent vehicles," *IEEE Trans. Intell. Veh.*, vol. 7, no. 3, pp. 417–440, Sep. 2022.

[4] Y. Huang, J. Du, Z. Yang, Z. Zhou, L. Zhang, and H. Chen, "A survey on trajectory-prediction methods for autonomous driving," *IEEE Trans. Intell. Veh.*, vol. 7, no. 3, pp. 652–674, Sep. 2022.

[5] G. F. Newell, "A simplified car-following theory: A lower order model," *Transp. Res. Part B: Methodological*, vol. 36, no. 3, pp. 195–205, Mar. 2002, doi: 10.1016/S0191-2615(00)00044-8.

[6] J. Huang and S. Agarwal, "Physics informed deep learning for traffic state estimation," in *Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst.*, 2020, pp. 1–6, doi: 10.1109/ITSC45102.2020.9294236.

[7] R. Shi, Z. Mo, K. Huang, X. Di, and Q. Du, "A physics-informed deep learning paradigm for traffic state and fundamental diagram estimation," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 11688–11698, Aug. 2021, doi: 10.1109/TITS.2021.3106259.

[8] Y. Yuan, Z. Zhang, X. T. Yang, and S. Zhe, "Macroscopic traffic flow modeling with physics regularized gaussian process: A new insight into machine learning applications in transportation," *Transp. Res. Part B: Methodological*, vol. 146, pp. 88–110, Apr. 2021, doi: 10.1016/j.trb.2021.02.007.

[9] R. E. Chandler, R. Herman, and E. W. Montroll, "Traffic dynamics: Studies in car following," *Operations Res.*, vol. 6, no. 2, pp. 165–184, 1958.

[10] P. Gipps, "A behavioural car-following model for computer simulation," *Transp. Res. Part B: Methodological*, vol. 15, no. 2, pp. 105–111, Apr. 1981, doi: 10.1016/0191-2615(81)90037-0.

[11] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Phys. Rev. E*, vol. 62, no. 2, pp. 1805–1824, Aug. 2000, doi: 10.1103/PhysRevE.62.1805.

[12] J. Wang, Z. Zhang, F. Liu, and G. Lu, "Investigating heterogeneous car-following behaviors of different vehicle types, traffic densities and road types," *Transp. Res. Interdiscipl. Perspectives*, vol. 9, Mar. 2021, Art. no. 100315, doi: 10.1016/j.trip.2021.100315.

[13] C. Zhang and L. Sun, "Bayesian calibration of the intelligent driver model," *IEEE Trans. Intell. Transp. Syst.*, early access, Jan. 26, 2024, doi: 10.1109/TITS.2024.3354102.

[14] X. Jia et al., "Physics-guided machine learning from simulation data: An application in modeling lake and river systems," in *Proc. IEEE Int. Conf. Data Mining*, 2021, pp. 270–279, doi: 10.1109/ICDM51629.2021.00037.

[15] M. Zhou, X. Qu, and X. Li, "A recurrent neural network based microscopic car following model to predict traffic oscillation," *Transp. Res. Part C: Emerg. Technol.*, vol. 84, pp. 245–264, Nov. 2017, doi: 10.1016/j.trc.2017.08.027.

[16] X. Huang, J. Sun, and J. Sun, "A car-following model considering asymmetric driving behavior based on long short-term memory neural networks," *Transp. Res. Part C: Emerg. Technol.*, vol. 95, pp. 346–362, Oct. 2018, doi: 10.1016/j.trc.2018.07.022.

[17] X. Zhang, J. Sun, X. Qi, and J. Sun, "Simultaneous modeling of car-following and lane-changing behaviors using deep learning," *Transp. Res. Part C: Emerg. Technol.*, vol. 104, pp. 287–304, Jul. 2019, doi: 10.1016/j.trc.2019.05.021.

[18] L. Ma and S. Qu, "A sequence to sequence learning based car-following model for multi-step predictions considering reaction delay," *Transp. Res. Part C: Emerg. Technol.*, vol. 120, Nov. 2020, Art. no. 102785, doi: 10.1016/j.trc.2020.102785.

[19] M. Zhu, S. S. Du, X. Wang, Y. Hao, Z. Pu, and Y. Wang, "TransFollower: Long-sequence car-following trajectory prediction through transformer," 2022, *arXiv:2202.03183*.

[20] K. Zhang, L. Zhao, C. Dong, L. Wu, and L. Zheng, "AI-TP: Attention-based interaction-aware trajectory prediction for autonomous driving," *IEEE Trans. Intell. Veh.*, vol. 8, no. 1, pp. 73–83, Jan. 2023.

[21] Z. Mo, R. Shi, and X. Di, "A physics-informed deep learning paradigm for car-following models," *Transp. Res. Part C: Emerg. Technol.*, vol. 130, Sep. 2021, Art. no. 103240, doi: 10.1016/j.trc.2021.103240.

[22] J. Farrell, T. Givargis, and M. Barth, "Differential carrier phase GPS-aided INS for automotive applications," in *Proc. Amer. Control Conf.*, 1999, pp. 3660–3664, doi: 10.1109/ACC.1999.782449.

[23] Y. Feng, "Intelligent traffic control in a connected vehicle environment," Ph.D. dissertation, The University of Arizona, Tucson, AZ, USA, 2015.

[24] V. Punzo, M. T. Borzacchiello, and B. Ciuffo, "On the assessment of vehicle trajectory data accuracy and application to the next generation SIMulation (NGSIM) program data," *Transp. Res. Part C: Emerg. Technol.*, vol. 19, no. 6, pp. 1243–1262, Dec. 2011, doi: 10.1016/j.trc.2010.12.007.