

How Does Traffic Environment Quantitatively Affect the Autonomous Driving Prediction?

Wenbo Shao¹, Yanchao Xu, Jun Li, Chen Lv², Senior Member, IEEE, Weida Wang³, Senior Member, IEEE, and Hong Wang⁴, Senior Member, IEEE

Abstract—Accurate trajectory prediction is essential for safe and efficient autonomous driving in complex traffic environments. While artificial intelligence has shown great promise in improving prediction accuracy, its inherent uncertainty and lack of explainability may lead to unpredictable failures, creating challenges for safety-critical decision-making. This study aims to address these challenges by exploring the impact of traffic environment on prediction algorithms. The study proposes a trajectory prediction framework with epistemic uncertainty estimation ability to output high uncertainty when facing unforeseeable or unknown scenarios. The framework analyzes the environmental effect on the trajectory prediction by considering scenario features and shifts. Features are divided into kinematic features of a target agent, features of surrounding traffic participants, and other scenario features. Feature correlation and importance analyses are performed to study their influence on prediction error and epistemic uncertainty. The impact of unavoidable distributional shifts in the real world on trajectory predictions is investigated using multiple intersection datasets. The results indicate that deep ensemble-based methods have advantages in improving robustness while estimating epistemic uncertainty. Consistent conclusions were obtained from the correlation and importance analyses, indicating that kinematic features of the target agent have relatively strong effects on both prediction error and epistemic uncertainty. Finally, the study analyzes the accuracy deterioration caused by distributional shifts and the potential of the deep ensemble-based method. Through deep ensemble, the errors of the prediction methods based on GRIP++ and Trajectron++ have been improved by 6.4% and 10.8% in the same-dataset test, and 6.3% and 10.8% in the cross-dataset test.

Index Terms—Artificial intelligence, autonomous driving, distributional shift, epistemic uncertainty, traffic environment, trajectory prediction.

Manuscript received 11 October 2022; revised 9 April 2023; accepted 8 May 2023. Date of publication 2 June 2023; date of current version 4 October 2023. This work was supported in part by the National Science Foundation of China under Project 52072215 and Project U1964203 and in part by the National Key Research and Development Program of China under Grant 2022YFB2503003. The Associate Editor for this article was M. Shojafar. (Corresponding author: Hong Wang.)

Wenbo Shao, Jun Li, and Hong Wang are with the Tsinghua Intelligent Vehicle Design and Safety Research Institute, School of Vehicle and Mobility, Tsinghua University, Beijing 100084, China (e-mail: swb19@mails.tsinghua.edu.cn; lj19580324@126.com; hong_wang@mail.tsinghua.edu.cn).

Yanchao Xu and Weida Wang are with the School of Mechanical Engineering, Beijing Institute of Technology, Beijing 100081, China (e-mail: 3120200410@bit.edu.cn; wangwd0430@163.com).

Chen Lv is with the School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore 639798 (e-mail: lyuchen@ntu.edu.sg).

Digital Object Identifier 10.1109/TITS.2023.3278695

1558-0016 © 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See <https://www.ieee.org/publications/rights/index.html> for more information.

I. INTRODUCTION

A. Motivation

TRAJECTORY prediction is a crucial component of the autonomous driving pipeline [1]. To ensure safe and efficient navigation in complex traffic environments, autonomous vehicles (AVs) must accurately and reliably predict the future motion of surrounding traffic participants (TPs), such as pedestrians and vehicles. In recent years, artificial intelligence (AI) has been rapidly developed and widely applied in AV trajectory prediction [2], [3]. This is achieved through the accumulation of large-scale driving data and the development of complex advanced algorithms, resulting in promising results. Although AI-based prediction shows great promise, it also faces several significant challenges. AI models are highly complex and difficult to interpret, which can lead to unexpected and unexplainable failures, significantly reducing the reliability of prediction models. Despite some attempts to improve trajectory prediction algorithms considering multiple factors simultaneously [4], [5], [6], performance degradation may still occur in complex traffic environments.

AI-based trajectory prediction algorithms are essentially data-driven and designed to learn the most efficient prediction model possible from the given data. The model predicts the future trajectory of target agents (TAs) in a scenario, and their performance is mainly dependent on the availability of training data, training processes, and model design. Consequently, in scenarios where training data is insufficient or where environmental factors are not effectively considered during the modeling process, the prediction model may exhibit severe epistemic deficiencies, leading to high levels of uncertainty or poor prediction accuracy. Thus, the performance of the prediction model is highly dependent on environmental factors. Specifically, various traffic environment features may have different impacts on trajectory prediction. Additionally, significant changes in the environment may cause distributional shifts of the test data relative to the training data, leading to performance degradation of prediction models based on the independent and identically distributed (IID) assumption between training and test data [7]. Unfortunately, there has been no systematic research investigating the impact of traffic environment on trajectory prediction errors and epistemic uncertainty. However, such research is critical in identifying the main limitations of prediction models, extracting common

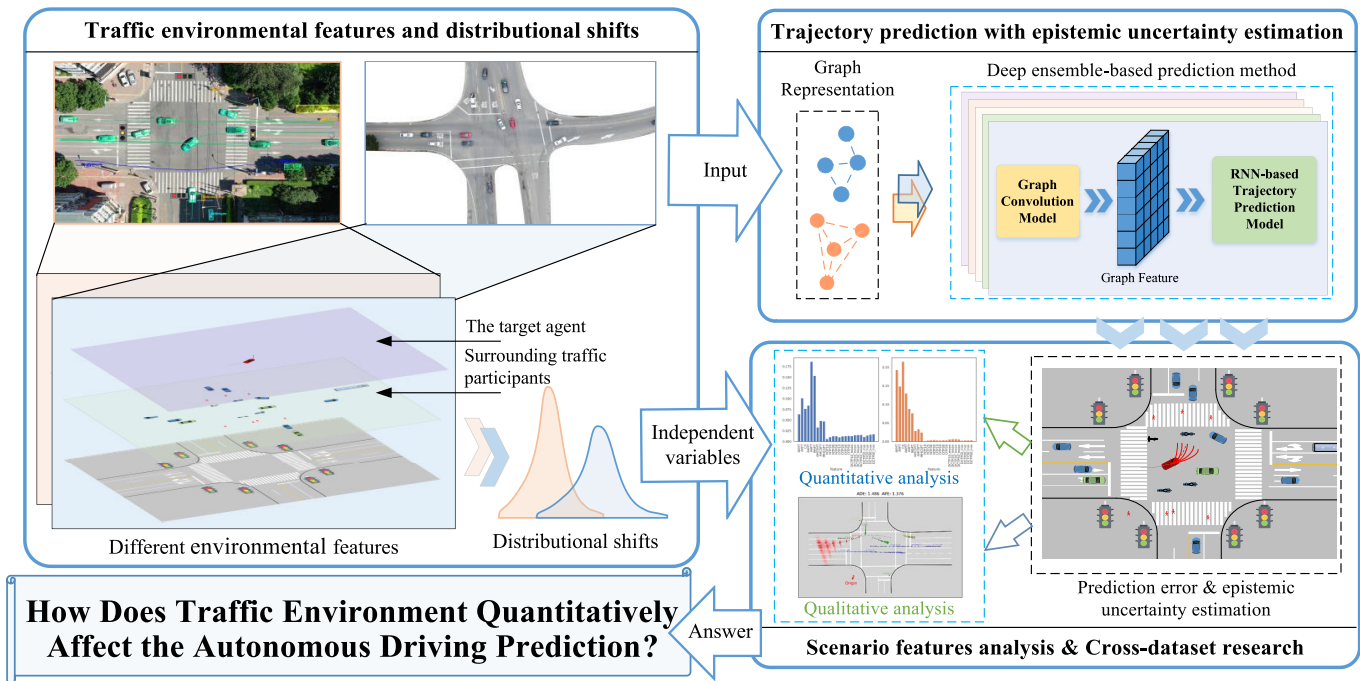


Fig. 1. Illustration of the traffic environment's effect on the trajectory prediction. Traffic environmental data include various TA states, their surrounding TP states, and other contextual information, which may affect the prediction differently. In addition, variations in time and place may lead to distributional shifts, which may further degrade the prediction performance. This study focuses on extracting these factors and analyzing their influence on prediction performance.

influencing factors, and improving and monitoring these prediction models.

B. Contribution

The focus of this work (as shown in Fig. 1) is on exploring the effects of both specific scenario features and distributional shifts on the prediction algorithm. Previous studies have commonly employed prediction error as a kind of evaluation metric for trajectory prediction. This study proposes to expand on this metric by incorporating epistemic uncertainty [8], [9] as a means of quantifying the level of cognition of the prediction algorithm regarding the current environment. When a prediction algorithm is unfamiliar with a certain scenario or lacks sufficient confidence in its understanding, it tends to output higher levels of epistemic uncertainty. These two types of metrics collectively constitute the evaluation system for trajectory prediction in this study. Representative features for traffic environments are defined and quantified, and two quantitative analysis methods are proposed for a comprehensive analysis. Furthermore, cross-dataset experiments are conducted to investigate changes in prediction performance under environmental shifts. The main contributions of this work can be summarized as follows:

- 1) **A trajectory prediction framework that integrates epistemic uncertainty estimation is proposed.** This framework enables simultaneous prediction of the TA's future state and estimation of epistemic uncertainty. It is implemented and evaluated based on various representative trajectory prediction methods;
- 2) **The potential of the proposed deep ensemble (DE)-based trajectory prediction framework for improving the prediction accuracy and estimating**

epistemic uncertainty is demonstrated. The errors of synthetic trajectories from DE-based methods are improved by 3.7% - 17.1% compared to single-model-based prediction errors, and are more robust;

- 3) **A general analysis framework is proposed to study the impact of scenario features on the prediction error and epistemic uncertainty.** Key scenario features are extracted to cover different environmental elements, and methods including feature correlation analysis and feature importance analysis are designed for analysis;
- 4) **The comprehensive experiments and systematic analysis are set up to investigate the distributional shifts between various intersection datasets and their impact on prediction performance.** The shifts of feature distribution among the 6 datasets and the generalization challenge to the prediction algorithm are analyzed. In addition, the experimental results demonstrate that DE helps to improve the robustness of trajectory prediction against distributional shifts.

This paper is structured as follows. Section II reviews the existing literature related to this topic. Section III introduces the proposed method. Section IV provides details on the datasets, evaluation metrics, and implementation used in this work. Section V analyzes and discusses the experimental results. Section VI concludes the paper.

II. RELATED WORK

A. Trajectory Prediction

Numerous studies have focused on improving trajectory prediction algorithms, which can be divided into physics-based, maneuver-based, and interaction-aware methods [10].

Physics-based and maneuver-based methods have shown poor performance in long-term prediction tasks due to the lack of modeling of interactions between TAs and the environment, and recent works have seen a shift towards interaction-aware methods. For example, social pooling (S-pooling) [11] and convolutional social pooling [12] are used to model inter-agent interactions within a certain spatial distance. Graph-based models, such as graph convolutional networks and graph attention networks [2], [13], have shown promising results in modeling interaction while also handling the heterogeneity of agents. In addition, time series processing has been a crucial requirement for trajectory prediction. In addition, time series processing has been a crucial requirement for trajectory prediction. Recurrent neural networks (RNNs), such as long-short-term memory (LSTM) and gated recurrent unit (GRU) models, have been widely used as submodules in trajectory prediction algorithms [6]. These recent advances in interaction modeling and time-series processing have shown significant improvements in the trajectory prediction accuracy.

Neural networks have shown high efficiency in predicting trajectories for different types of TP. Research on modeling pedestrian intentions and predicting their movement has been conducted for decades. Social-LSTM [11] is a successful example in early research, which combines S-pooling and LSTM to predict the future trajectory of pedestrians in crowded scenarios. Social-GAN [14] uses generative adversarial networks (GANs), sequence-to-sequence models, and pooling mechanisms to predict pedestrians' socially feasible future and employs corresponding generators and recursive discriminators. However, training the GAN model is challenging and may not converge, leading to mode collapsing and dropping. Therefore, the Social-Ways uses the Info-GAN, which adds another item to consider mutual information instead of applying the mean square error loss (L2 loss) to force generated samples to be close to real data, thus mitigating the above-mentioned issues.

Since vehicles have higher velocity and need to obey more road constraints than pedestrians, predicting their future movements is a prerequisite for realizing safe and efficient autonomous driving. Several studies have designed specialized networks for vehicle trajectory prediction [15]. For example, vehicle trajectory prediction in highway scenarios, which are relatively simple and where the motion pattern of a vehicle is relatively fixed, has received early attention [12], [16], [17]. With the collection of large-scale datasets [18], [19] and the development of autonomous driving in urban scenes, much research has focused on motion prediction in complex urban environments [4], [20], [21], [22]. TrafficPredict [23] adopts a four-dimensional graph to model the interaction in the instance and category layers, thus predicting the trajectories of heterogeneous traffic agents. GRIP++ achieves joint trajectory prediction of all observed objects while considering multiple classes of TPs, thus greatly improving real-time prediction performance. Trajectron++ [6] achieves an effective prediction of TPs by encoding agent interactions, incorporating heterogeneous data and modeling dynamically feasible trajectories, while using a Conditional Variational Autoencoder (CVAE) to model multimodality. However, previous work focus on

improving the accuracy at the data set level while ignoring the sensitivity of the prediction algorithm to environmental factors, which is the focus of this work.

B. Epistemic Uncertainty Estimation

The original neural network cannot provide an estimate of its epistemic uncertainty. To address this limitation, several studies have quantified the epistemic uncertainty of neural networks [8], [24], [25], which indicates the confidence level of the network in its prediction results. The prominent methods include Bayesian neural networks (BNN), single-pass uncertainty estimation, and DE-based methods.

BNN quantifies the epistemic uncertainty of a neural network by introducing uncertainty into its parameters. The key challenge of these methods is to solve the posterior distribution of network parameters. Variational inference (VI) [26], which uses a prespecified family of distributions [27], [28], was widely used in early research due to its strong theoretical basis. However, VI has faced challenges in solving difficulty and computational complexity with the rapid growth in neural network structure complexity. To address these limitations, Monte Carlo dropout (MCD) [29], [30] was proposed to approximate the results obtained by sampling, assuming that the network weights are in accordance with a Bernoulli distribution. Theoretical demonstrations have shown that MCD has the ability to approximate epistemic uncertainty.

In single-pass uncertainty estimation, the uncertainty is obtained through one forward propagation, which has clear advantages in computational complexity. The deep evidence theory is a representative method and has been widely used in classification [31] and regression [32] tasks. However, these methods require that the original network output has a specific form, which limits their scalability. In addition, these methods do not consider the uncertainty of network weights. Some studies [33] position the uncertainty they extract as a distributional uncertainty, different from epistemic uncertainty.

In DE-based methods, the training process is adjusted to obtain multiple different models, and epistemic uncertainty is estimated by synthesizing the prediction results of the models. DE [34] is a simple, parallelizable and scalable uncertainty estimation method that has received extensive attention due to its excellent performance in estimating epistemic uncertainty [35]. Currently, this method has become a mainstream paradigm. To reduce the storage and computational costs of DE's practical application, many improved methods have been proposed [36], [37]. For example, Batch-Ensemble [36] reduces training and testing costs by defining each weight matrix as the Hadamard product of the shared weights of all ensemble members and the rank-one matrix of each member, but the uncertainty estimation performance is slightly reduced.

In summary, the VI-based method suffers from significant computational complexity, and both it and the single-pass uncertainty estimation method require substantial modifications to the original prediction network, which may result in decreased model performance and lack of scalability. In contrast, the MCD-based and DE-based methods exhibit stronger advantages in terms of versatility and scalability. In particular, DE-based methods have demonstrated good performance

in previous studies. Therefore, this work proposes a trajectory prediction method with epistemic uncertainty estimation, where DE and MCD are used separately to estimate epistemic uncertainty and are compared on a real intersection dataset.

C. Relationship Between Prediction Performance and Traffic Environment

Previous studies focused on improving trajectory prediction accuracy at the dataset level, but traffic environment can significantly affect prediction performance. It is crucial to establish a correlation between the environment and the prediction model to enhance the interpretability of the prediction algorithms and identify their limitations. In fact, the prediction model relies directly on information from perception and V2X modules. With the advancement of connected and autonomous vehicles, there is an opportunity to enhance the quality of this information, such as object detection for edge assisted autonomous mobile vision [38] and high energy-efficient virtual machine placement [39]. Therefore, this work assumes that the prediction model can directly access environmental information and analyze its impact on the prediction model.

Several works focused on modeling and complexity calculation of a traffic environment using different methods, such as five- and six-layer scene models [40], [41], where layer elements can have a strong correlation with the prediction algorithm. Wang et al. [42] proposed a method to quantify scenario complexity in traffic but did not explore its relationship with the autonomous driving algorithm performance. The Shapley value is a feature attribution method that helps to measure the contribution of input variables to model performance. Makansi et al. [43] proposed a variant of Shapley value and analyzed the problems that some of the existing trajectory prediction models consider only the past trajectory of a TA and are difficult to reason about interactions. In addition, recent studies have gradually paid attention to the cross-dataset performance of AI algorithms in object detection and prediction applications [44], [45]. Gesnoui et al. [46] evaluated the impact of differences in pedestrian poses and detection box heights in different datasets on the prediction of pedestrian crossings. Gilles et al. [7] compared the accuracy of vehicle trajectory prediction algorithms on several datasets that contain mixed scenarios. However, there has still been a lack of comprehensive analysis of traffic environmental factors and their changes and quantitative research on their impact on prediction algorithms.

In this work, the research scenario is the intersection, which is a typical and challenging urban scenario. Distributional shifts between different intersection datasets and their effect on trajectory prediction performance are analyzed, considering both error and epistemic uncertainty.

III. PROPOSED METHOD

A. Trajectory Prediction With Epistemic Uncertainty Estimation

1) *Trajectory Prediction*: Trajectory prediction is a task that estimates a TA's future position based on its historical state \mathbf{X} and context \mathbf{C} in a scenario. In particular, at time $t = 0$, the

TABLE I
NOTATION TABLE

Symbol	Description
$s^{(t)}, \hat{s}^{(t)}$	The real and estimated state at time t
\mathbf{X}	TA's Historical state
\mathbf{C}	Scene context for trajectory prediction
$\mathbf{Y}, \hat{\mathbf{Y}}$	Ground truth and estimates of future trajectories
$\theta, \hat{\theta}$	The optimal and estimated parameters of predictor
\mathcal{D}	Training dataset
$P(\bullet \mathcal{D})$	Posterior distribution
$q(\theta)$	Approximate distribution of $P(\bullet \mathcal{D})$
$f(\bullet)$	Model prediction
$\mathcal{H}[\bullet]$	Entropy calculation
$[x^{(t)}, y^{(t)}], [\hat{x}^{(t)}, \hat{y}^{(t)}]$	The real and estimated positions of TAs at time t
$\sigma^2(\bullet)$	Variance calculation
ℓ	Loss function
ρ	Correlation coefficient
$R(x)$	The ranking of x in a group of data
$\overline{R(x)}$	The mean of the ranking of the groups of data

historical input state of TA \mathbf{X} over previous t_h time steps is $\mathbf{X} = [s^{(-t_h+1)}, s^{(-t_h+2)}, \dots, s^{(0)}]$, which contains information such as TA position. In addition, the states of TPs' around the TA and other environmental information are modeled as the scene context \mathbf{C} .

A trajectory prediction model is trained on the dataset \mathcal{D} . Based on input $[\mathbf{X}, \mathbf{C}]$, the trained prediction model produces an estimate $\hat{\mathbf{Y}}$ of the real future trajectory \mathbf{Y} of the TA as follows:

$$\hat{\mathbf{Y}} = f(\mathbf{X}, \mathbf{C}) = f(\mathbf{X}, \mathbf{C}, \mathcal{D}) = f(\mathbf{X}, \mathbf{C}, \hat{\theta}), \quad (1)$$

where $\hat{\mathbf{Y}} = [\hat{s}^{(1)}, \hat{s}^{(2)}, \dots, \hat{s}^{(t_f)}]$, t_f is the predicted horizon, and $\hat{\theta}$ represents the trained model parameters.

To demonstrate the generalizability of the proposed approach, this study employs two representative prediction algorithms, GRIP++ and Trajectron++, as the base models. Among these, GRIP++ is an enhanced graph-based interaction-aware trajectory prediction method that utilizes both fixed and dynamic undirected graphs to model the relationships between TPs, taking into account the impact of inter-agent interactions on a TA's motion. Moreover, GRIP++ employs a submodule with a GRU-based encoder-decoder architecture to facilitate joint trajectory predictions for multiple agents, achieving superior performance in terms of prediction accuracy and speed.

Trajectron++ employs a spatio-temporal graph representation by using a directed graph to model scenes, which allows for a wider range of interaction types and simultaneous modeling of agents with different perception ranges. To enhance prediction accuracy, Trajectron++ adopts a simplified model of single integrators for pedestrians and dynamically-extended unicycles for wheeled vehicles. The model represents neighboring agents' influence using an LSTM and additive attention, resulting in an "influence" representation vector that is concatenated with node history to form a single node representation vector. Unlike GRIP++, which models the output for a single deterministic trajectory, Trajectron++ utilizes a CVAE with a discrete latent variable to handle multimodality and a bidirectional LSTM to encode a node's ground-truth

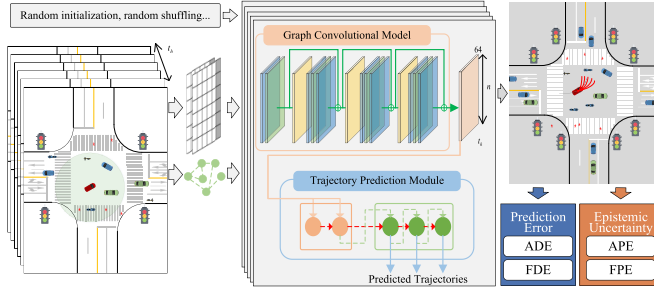


Fig. 2. The trajectory prediction framework with epistemic uncertainty estimation (DE-based method).

future trajectory during training. This study adopts two types of output forms for analysis: the first outputs only the most likely trajectories (ml), while the second generates multiple trajectories through a sampling method (mul).

2) *Epistemic Uncertainty Estimation*: The original prediction network can only output the predicted trajectories and lacks the estimation of epistemic uncertainty. Teal-world traffic scenarios are complex and variable, and it is not possible to construct a training set that will effectively cover all scenarios, so the predictions of an AI-based model may not be reliable enough when confronted with unknown scenarios. In this work, an epistemic uncertainty estimation framework for trajectory prediction networks is proposed to represent the model's cognitive level for a given scenario. In this regard, the BNN models and learns the posterior distribution of network weights $\hat{\theta} \sim P(\theta | D)$, which can be used to estimate epistemic uncertainty as follows:

$$\hat{Y} = f(\mathbf{X}, \mathbf{C}, D) = \int f(\mathbf{Y} | \mathbf{X}, \mathbf{C}, \theta) P(\theta | D), \quad (2)$$

where the primary challenge is to effectively estimate the posterior distribution of parameters.

The Bayesian approximate inference is a typical solution, which learns an approximate distribution $q(\theta)$ of $P(\theta | D)$. MCD has been shown to be an effective sample-based method for approximate inference, where the network weights are assumed to follow the Bernoulli distribution. After adding appropriate regularization during training and turning on dropout during testing, K different predictions can be obtained by sampling multiple times. In addition, DE has shown excellent uncertainty estimation ability. Specifically, random initialization of neural network parameters and random shuffle of dataset are performed because they have been proven to have enough good performance in practice. After training, K models of isomorphism and different parameters are obtained.

The algorithm 1 demonstrates the training and testing process of the DE-framework. For the MCD-based framework, only one model is trained in the training phase, but in the test phase the same number of results are obtained by turning on dropout and propagating forward K times. The predictive entropy is calculated to quantify the epistemic uncertainty, where entropy increases with uncertainty. To realize a prediction-task-wise uncertainty estimation, the predictive entropy at multiple moments is integrated to obtain the average

Algorithm 1 Epistemic Uncertainty Estimation Based on Deep Ensemble

Input: Training data \mathcal{D} , number of ensemble models K , number of iterations T , Loss function ℓ , base learning rate η , learning rate decay rate γ , mini-batch size B , and other hyperparameters

- 1 **Training Phase:** for $k \leftarrow 1$ to K do
- 2 Initialize parameter θ_k of the k th neural network randomly and shuffle training data randomly;
- 3 **for** $t \leftarrow 1$ to T do
- 4 Sample a mini-batch of size B from \mathcal{D}_{train} ;
- 5 Compute gradient $\nabla_{\theta_k} \frac{1}{B} \sum_{i=1}^B \ell$ using backpropagation;
- 6 Update the parameters using stochastic gradient descent with momentum and weight decay;
- 7 Decay the learning rate according to γ ;
- 8 Obtain the K models with the lowest validation error;
- 9 **Testing Phase:** for $k \leftarrow 1$ to K do
- 10 Obtain the prediction results of the k th trained model for the given input;
- 11 Compute APE, FPE, and synthetic trajectory based all prediction results for the given input.
- 12 **return** K trained models, APE, FPE and the predicted trajectories during testing phase

predictive entropy (APE) as follows:

$$\text{APE} = \frac{1}{t_f} \sum_{i=1}^{t_f} \mathcal{H}[\hat{s}^{(i)}] = \frac{1}{t_f} \sum_{i=1}^{t_f} - \int p(\hat{s}^{(i)}) \ln p(\hat{s}^{(i)}) d\hat{s}_i. \quad (3)$$

Assuming that $\hat{s}^{(i)} = [\hat{x}^{(i)}, \hat{y}^{(i)}]$ obeys the two-dimensional Gaussian distribution, $\hat{x}^{(i)}$ and $\hat{y}^{(i)}$ are independent of each other, and the APE can be expressed as follows:

$$\begin{aligned} \text{APE} &= \frac{1}{t_f} \sum_{i=1}^{t_f} \left[(\ln 2\pi + 1) + \frac{1}{2} \ln |\hat{\Sigma}^{(i)}| \right] \\ &= \frac{1}{t_f} \sum_{i=1}^{t_f} \left[(\ln 2\pi + 1) + \frac{1}{2} \ln \sigma^2(\hat{x}^{(i)}) \sigma^2(\hat{y}^{(i)}) \right]. \quad (4) \end{aligned}$$

Similarly, the final predictive entropy (FPE) is defined as:

$$\begin{aligned} \text{FPE} &= \mathcal{H}[\hat{s}^{(t_f)}] = (\ln 2\pi + 1) + \frac{1}{2} \ln |\hat{\Sigma}^{(t_f)}| \\ &= (\ln 2\pi + 1) + \frac{1}{2} \ln \sigma^2(\hat{x}^{(t_f)}) \sigma^2(\hat{y}^{(t_f)}). \quad (5) \end{aligned}$$

$\sigma^2(\bullet)$ in Eq. 4 and 5 are calculated from all the predicted trajectories. For GRIP++ and Trajectron++ (ml), the proposed framework outputs K trajectories $\hat{Y}_k = [\hat{s}_k^{(1)}, \hat{s}_k^{(2)}, \dots, \hat{s}_k^{(t_f)}]$ for a given input, each of which contains the predicted position at multiple future moments. The corresponding APE and FPE represent the degree of dispersion of all models. Trajectron++ (mul)-based framework outputs $K \times n_{sam}$ trajectories simultaneously, where a single model output

n_{sam} trajectories. The Trajectron++ (mul)-based framework generates $K \times n_{sam}$ trajectories concurrently, where a single model produces n_{sam} trajectories. The prediction entropy associated with these trajectories represents the combined uncertainty of both the model and TA motion.

Furthermore, for GRIP++ and Trajectron++ (ml), the synthetic trajectory is obtained by integrating all the predictions:

$$\hat{\mathbf{Y}} = K^{-1} \sum_{k=1}^K f(\mathbf{Y} | \mathbf{X}, \mathbf{C}, \hat{\theta}_k), \quad (6)$$

where $\hat{\theta}_k$ denotes the parameter of the k th model from DE-based or the k th sampling from MCD.

B. Scenario Features Extraction

In prediction scenarios, a TA's movement is influenced by its past state, interactions with neighboring TPs, and other factors. While current prediction algorithms have taken into account various factors either explicitly or implicitly, their performances may still be hindered by the aforementioned features due to algorithmic limitations. This work considers three types of features: 1) kinematic features of a TA, related to its historical or future motion states; 2) features of surrounding TPs, which refer to their states and interactions with the TA; 3) other scenario features, which include the type, behavior pattern, compliance with traffic rules, and location of the TA.

1) *Kinematic Features of TA*: Historical motion state of TA is crucial for predicting its future behavior and directly affects the output of a prediction model. Additionally, the future motion state of a TA is a key reference for evaluating the model's prediction accuracy. Therefore, the kinematic features of TA are extracted to analyze their impact on the prediction algorithm performance.

Velocity is one of the primary kinematic features, which directly affects the discrete degree of a continuous trajectory. Considering the trajectory prediction model characteristics, three velocity sub-features are extracted: 1) average velocity of the historical trajectory (AVHT), which indicates the aggressiveness of the model's input trajectory; 2) current velocity (CV), which directly represents a TA's current state and has a key impact on the trajectory prediction output; 3) average velocity of the future trajectory (AVFT), which reflects the spatial span of the future trajectory points.

In addition, the velocity variations indicate trajectory stationarity, which may have a significant influence on prediction results. For instance, a sudden start of a parked vehicle may be difficult for the model to predict timely and accurately. Therefore, the acceleration value at each moment is calculated to obtain the following sub-features: 1) average acceleration of the historical trajectory (AAHT), which represents the speed mutation degree of the input trajectory; 2) average acceleration of the future trajectory (AAFT), which reflects the overall situation of a TA's future speed mutation; 3) maximum acceleration of the future trajectory (MAFT), considering that a sudden speed change at any moment can lead to severe deformation of the overall trajectory, it is necessary to extract the fastest speed change in the future as a feature for analysis.

Similarly, changes in the TA moving direction denote a potentially influential factor of prediction performance. For instance, a vehicle going straight may suddenly swerve or make a U-turn, thus posing a serious challenge to the prediction algorithm. Therefore, the heading change speed (HCS) is extracted for analysis. In detail, the absolute value of the change speed of the heading angle at each moment is calculated and used as a basic feature, and then the analysis of the following parameters is performed: 1) average HCS of the historical trajectory (AHCSHT); 2) average HCS of the future trajectory (AHCSFT), which reflects the overall curvature or volatility of the future trajectory; 3) maximum HCS of the future trajectory (MHCSFT), which increases when there is a sudden large change in direction at any point in the future.

2) *Features of Surrounding TPs*: Convoluted interactions with other agents increase the difficulty in trajectory prediction, and although many of the existing prediction methods can explicitly or implicitly model interactions, it has not been fully discussed whether the performance of these black-box models is sensitive to actual interactions. To examine this situation, a set of hierarchical prediction scenario complexity metrics is proposed to analyze the effect of a TA's interactions with surrounding agents on the prediction algorithm performance.

First, with a TA as a center, the prediction scenario complexity has a positive correlation with the number of its surrounding TPs, and a basic feature, the number of TPs within x meters from the TA (NTP_x), is defined.

In addition, the distance between the TA and its surrounding TPs directly affects the prediction scenario complexity. Assuming that a set of TPs within x meters of a TA i is denoted by $N_x(i)$, then for any $j \in N_x(i)$, its distance from i is $d_j = \text{dist}(s_i, s_j)$. The density of TPs within x meters around TA (DTP_x) is given by:

$$DTP_x = \sum_{j=1}^{N_x(i)} e^{-\lambda d_j}, \quad (7)$$

where λ is a scaling factor.

Furthermore, the potential conflicts due to the movement of surrounding TPs are analyzed. As shown in Fig. 3, for a TP $j \in N_x(i)$ within x meters around a TA i , its current state is given by $[s_j^{(0)}, v_j^{(0)}]$; then, its position after t seconds is expressed as $s_j^{(t)} + v_j^{(t)}t$, and the degree of conflict from TPs within x meters around TA ($DCTP_x$) is defined as follows:

$$DCTP_x = \sum_{j=1}^{N_x(i)} e^{-\lambda \text{agg}^{(T)}(\alpha^{(t)} d_j^{(t)})}, \quad (8)$$

where $d_j^{(t)}$ is the distance between TA i and TP j , T is the time horizon used for evaluation; $\alpha^{(t)}$ is the scaling factor for the distance at time t , and in this study, it is set to grow faster over time to reinforce the focus on the short-term risk. $\text{agg}(\bullet)$ represents the aggregation operation, which is used to synthesize the conflicts at T times. The two basic modes used in this study include the mean value ($DCTP_{x,\text{mean}}$) and the maximum value ($DCTP_{x,\text{max}}$).

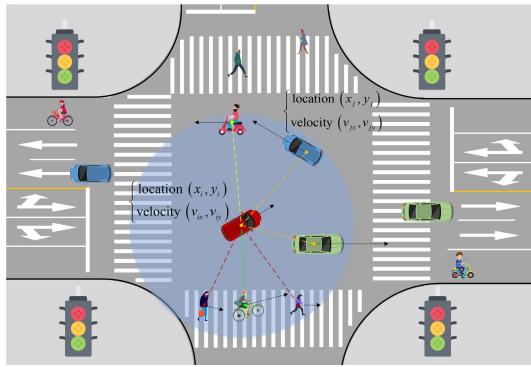


Fig. 3. Schematic diagram of the conflict degree calculation from TPs within x meters from a TA.

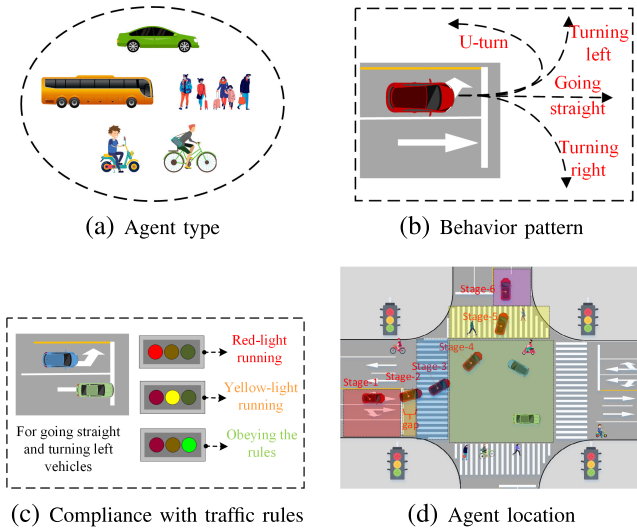


Fig. 4. Illustration of the other extracted scenario features.

3) *Other Scenario Features*: In addition to the above two categories of features, several other representative features are also studied, as shown in Fig. 4, including:

- TA type: Each type of agent has its movement pattern, which may cause different prediction performances. Referring to the research presented in [23], TAs can be divided into four types: small vehicles, vehicles, pedestrians, motorcyclists, and bicyclists;
- TA behavior pattern: This study mainly focused on three basic behavior patterns of vehicles at intersections: going straight, turning left, and turning right. Moreover, U-turn, as a certain corner case, is extracted;
- TA's compliance with traffic rules: Traffic rules partially constrain the behaviors of participants, but in real-world scenarios, some of TAs may violate the rules, thus affecting the trajectory prediction. For instance, in a signalized intersection, the behaviors of TAs can be classified based on their compliance with the traffic signal into obeying the rules, yellow-light running, and red-light running;
- TA location: The whole process of a vehicle passing through the intersection is divided according to the time sequence into six stages: stage 1: ex-entering an intersection; stage 2: in the gap; stage 3: in the first

crosswalk; stage 4: inside an intersection; stage 5: in the last crosswalk; stage 6: exiting an intersection.

C. Scenario Features Analysis

To analyze the above-mentioned features systematically, the qualitative and quantitative analysis methods are adopted. The main methods include feature correlation analysis and feature importance analysis based on random forest regression.

1) *Feature Correlation Analysis*: Correlation analysis is to calculate the degree of correlation between two or more feature variables using correlation coefficients as quantitative indicators. Typical correlation coefficients include the Pearson correlation coefficient and Spearman rank correlation coefficient. The Pearson correlation coefficient requires evaluated variables to conform to the normal distribution, but this is a strong assumption that experimental results can hardly satisfy. In contrast, the Spearman rank correlation coefficient does not have such strict requirements on data as the Pearson correlation. Namely, it requires only that observed values of the two variables are paired rank data or rank data transformed from continuous variable observation data. The Spearman rank coefficient can be mainly used in the monotonic relationship evaluation. Specifically, it is assumed that the original data (x_i, y_i) are arranged in ascending order, and $R(x_i)$ and $R(y_i)$ are defined as the ranking of x_i and y_i in their corresponding data, respectively; $\overline{R(x)}$ and $\overline{R(y)}$ denote the mean of the ranking of the two groups of data, and n is the number of data pairs. Then, the Spearman rank correlation coefficient ρ can be defined as follows:

$$\rho = \frac{\sum_{i=1}^n (R(x_i) - \overline{R(x)}) (R(y_i) - \overline{R(y)})}{\sqrt{\sum_{i=1}^n (R(x_i) - \overline{R(x)})^2 \cdot \sum_{i=1}^n (R(y_i) - \overline{R(y)})^2}} = 1 - \frac{6 \sum_{i=1}^n (R(x_i) - R(y_i))^2}{n(n^2 - 1)}. \quad (9)$$

2) *Feature Importance Analysis Based on Random Forest Regression*: Feature correlation analysis can be used to assess linear and ordinal consistent correlations, but cannot identify other types of correlations. To address this limitation, a feature importance analysis method is proposed as a supplement.

Random forest regression is a powerful technique that combines the decision tree framework with ensemble learning. This involves generating multiple decision trees from data and averaging multiple outputs to obtain the final result. The decision tree is a key component of this algorithm, consisting of a hierarchical tree structure with root, internal, and leaf nodes. In addition, ensemble learning techniques, such as bootstrapping, are used to improve prediction accuracy.

Feature importance analysis is performed by random forest regression. As shown in Algorithm 2, in this study, the extracted scenario features are regarded as input variables, while the prediction error or uncertainty of the prediction model are regarded as the label. Then the random forest regression models are constructed, and the contribution of each feature to the trees in the random forest is analyzed, which is denoted as the feature importance score (FIS). It is assumed

Algorithm 2 Feature Importance Analysis Based on Random Forest Regression

Data: Training dataset $(x_i, y_i)_{i=1}^n$, where x_i is the i th feature vector and y_i is the corresponding label.
Grid search parameter settings \mathcal{H}

- 1 Initialize empty dictionary FIS_{all} ;
- 2 **for** each combination of hyperparameters H in \mathcal{H} **do**
- 3 Train a random forest regression model f_H on the training dataset with hyperparameters H ;
- 4 **for** j in 1 to J **do**
- 5 **for** i in 1 to I **do**
- 6 Compute the importance score $\text{FIS}_j^{(i)}$ of feature j in tree i in f_H according to the difference of Gini indices of nodes before and after branching;
- 7 Calculate FIS_j according to Eq. 10 in f_H ;
- 8 Update FIS_j with $\text{FIS}_j^{(i)}$ in f_H
- 9 Update FIS_{all} with FIS_j in f_H ;
- 10 Compute the mean and variance of FIS_j in FIS_{all} ;
- 11 **return** Feature importance scores for each feature.

that there are J features and I decision trees. Then, FIS_j denotes the average change in node split impurity of the j th feature in all decision trees, and it is calculated by:

$$\text{FIS}_j = \frac{\sum_{i=1}^I \text{FIS}_j^{(i)}}{\sum_{j'=1}^J \sum_{i=1}^I \text{FIS}_{j'}^{(i)}}, \quad (10)$$

where $\text{FIS}_j^{(i)}$ represents the importance of the j th feature in the i th decision tree, and it can be obtained by calculating the difference of Gini indices of nodes before and after branching.

To mitigate the impact of hyperparameter settings, the grid-search approach is adopted to traverse different hyperparameter configurations. According to the number of trees in the forest, the maximum depth of the tree, and the number of features to consider when looking for the best split, multiple sets of hyperparameters were set in this study. A model is trained for each combination of hyperparameters H , and finally FIS_{all} from all models are obtained for statistics.

D. Prediction Across Different Intersection Datasets

The cross-dataset analysis aims to analyze differences in scenarios between multiple datasets and the corresponding prediction algorithm performance disparity. In this study, the scenario type is limited to the intersection, and multiple intersection datasets involving various countries are studied. First, the scenario features are extracted to analyze distributional shifts between different intersection datasets. Next, comprehensive cross-validation experiments are performed to investigate the prediction algorithm performance in terms of distributional shifts fully. Particularly, N intersection datasets are selected, and each of them is divided into training and test subsets. Then, for each of the training subsets, a trajectory prediction model is developed and trained using the training

subset first and then evaluated on the corresponding test subset. Finally, N^2 sets of results are obtained.

The main concerns of the experimental analysis are as follows: 1) distribution shifts among different intersection datasets and its influence on trajectory prediction performance; 2) the effect of the DE-based framework on improving prediction robustness and estimating epistemic uncertainty; 3) the effect of different intersection datasets as training sets for trajectory prediction and the prediction challenges when they are used as test sets.

IV. EXPERIMENTAL SETUP

A. Intersection Datasets

Focusing on the urban intersection scenario, multiple trajectory datasets are used for evaluation and analysis, involving various periods, weather, countries and regions, and TP types.

1) *SinD* [47]: To focus on the prediction for the intersection scenario, the SinD dataset was collected from a signalized intersection in Tianjin, China and labeled. The data were recorded by a drone at a sampling frequency of 10 Hz. It comprises approximately 420 minutes of traffic recordings and includes over 13,000 TPs of 7 types, such as cars, trucks, buses, pedestrians, tricycles, bikes, and motorcycles. The original SinD dataset contained some data from vehicles parked for long periods of time that might interfere with the analysis, so it was filtered out for this work.

2) *INTERACTION* [19]: *INTERACTION* contains 12 subsets, of which five intersection subsets are used in this study: USA_Intersection_EP1 (EP1), USA_Intersection_EP2 (EP2), USA_Intersection_MA (MA), USA_Intersection_GL (GL) and TC_Intersection_VA (VA). They contain about 493 minutes of recordings in total. The first four were from unsignalized intersections in the US, mainly involving trajectories of vehicles, pedestrians, and bicycles recorded by drones. VA was collected from a signalized intersection in Bulgaria, which involves trajectories of cars, buses, trucks, motorcycles, and bicycles recorded by traffic cameras.

Each dataset is divided into a training set and a test set, with the latter being not visible during the training process.

B. Prediction Error Metrics

For different trajectory prediction output forms, different prediction error metrics [2], [11], [14] are used to quantify the prediction performance. In GRIP++ and Trajectron++ (ml), each prediction model only outputs a single trajectory, so the following metrics are adopted: 1) *Average Displacement Error (ADE)*, which is the mean square error of all predicted points of a trajectory compared to the ground truth; 2) *Final Displacement Error (FDE)*, which is the distance between the predicted final destination and the true final destination at t_f . Trajectron++ (mul) is a typical multiple output model, thus the *minimum ADE (minADE)* and the *minimum FDE (minFDE)* over all predictions of a single model are adopted.

C. Joint Analysis of Prediction Error and Epistemic Uncertainty

Although prediction error and epistemic uncertainty are distinct metrics, as previously mentioned, epistemic uncertainty

indicates a model's confidence in its output and provides insight into how the model will perform in the given scenario. Thus, an accurate estimation of epistemic uncertainty should exhibit a strong correlation with prediction error and enhance the original predictive model. Therefore, for the prediction model that outputs only a single trajectory, this study evaluate *the prediction error of the synthesized trajectory* from the prediction framework with epistemic uncertainty estimation.

In addition, referring to [48] and [49], *the error-retention curves* are used to evaluate the estimated epistemic uncertainty. The curves depict the error over a dataset as a model's predictions are replaced by ground-truth labels in order of decreasing uncertainty. The abscissa value of a point represents the proportion of the retained true error (i.e., retention fraction), while the ordinate value represents the comprehensive error under this proportion. Similarly, the optimal and random curves are obtained by replacing the predictions in order of decreasing error and random order, respectively. *The area under the retention curves* (R-AUC) is an evaluation metric of both the robustness of prediction models and the quality of uncertainty estimation, and an efficient uncertainty estimation is considered to achieve a low R-AUC.

D. Implementation Details

Refer to the configuration in [5] and [6], the datasets are resampled to 2 Hz. In the experiment, the prediction model uniformly uses the historical data of 6 frames to predict the future trajectory of 6 frames. Moreover, the original dataset was further processed to obtain the labels for the subsequent analysis. The TAs' locations are classified by combining their raw coordinate data with the map in the Lanelet2 format [50].

The implementation of GRIP++ and Trajectron++, including the input preprocessing, model architecture and design parameters, and the basic loss function during training follow their original details. For Trajectron++ (the mul), n_{sam} is set to 10. In the implementation of MCD and DE, the value of K was set to 5, which was the result of a trade-off between uncertainty estimation quality and computational cost [35]. In addition, during the training process of the MCD-model, a regularization term was added to the loss to improve its ability of uncertainty estimation [29], where the regularization coefficient was set to 0.0001, and the dropout rate was set to 0.5

To ensure professionalism and comparability of experiments, this study employed standardized experimental equipment to test the required models. The tests for all models were executed using an NVIDIA GeForce RTX 2080 SUPER GPU and an Intel i7-9700 CPU, resulting in consistent test results. Additionally, all models were implemented using Python Programming Language and PyTorch library.

V. RESULTS AND DISCUSSION

A. Evaluation of Trajectory Prediction and Epistemic Uncertainty Estimation

The training and test performances of the proposed model obtained by following the train-test process in the same

intersection dataset are presented in TABLE II. Although the MCD can be used to estimate epistemic uncertainty, it increases the prediction error, which is due to the modifications in the original loss function. In contrast, the DE-based method not only does not affect the error of individual prediction models while estimating epistemic uncertainty, but also significantly reduces the prediction error of the synthetic trajectory based on DE. In the GRIP++-based method, ADE_{DE} was reduced by 3.7% to 8.8% compared to ADE, and FDE_{DE} was reduced by 3.7% to 8.7% compared to FDE. For the Trajectron++ (ml)-based method, ADE_{DE} was reduced by 4.2% to 17.1% compared to ADE, and FDE_{DE} was reduced by 4.4% to 18.9% compared to FDE. Thus, by integrating the results of multiple models, DE could effectively avoid the prediction performance degradation caused by the deviation of a single model, which is conducive to improving the prediction algorithm robustness.

The evaluation results of the estimated epistemic uncertainty are shown in Fig. 5. By observing the R-AUC for different prediction networks and error metrics, it can be found that the estimated epistemic uncertainty in this study has a positive effect on reflecting the magnitude of prediction errors, and is much better than the random method without prior. Compared with the MCD, the DE had obvious advantages in improving the model prediction accuracy and uncertainty estimation. Therefore, in the subsequent analysis, the epistemic uncertainty estimation framework based on the DE was adopted. In addition, as shown in Fig. 5, APE and FPE showed a high degree of consistency in the retention curve evaluation. Moreover, both the retention curve and the retention score exhibit a consistent pattern, where the retention score decreases as the fraction increases. This implies that cases with extremely high prediction errors do not necessarily correspond to high levels of epistemic uncertainty, and vice versa.

B. Scenario Features Analysis Results

1) *Feature Correlation Comparison*: Based on the scenarios from the test dataset, the TA's kinematic features and the features of the TA's surrounding TPs, then the correlation between the model performance and these two types of features was calculated to analyze the influence of scenario feature on trajectory prediction, where the prediction performance was represented by prediction error and epistemic uncertainty. For the features of the TA's surrounding TPs, four groups of distances of $x = 10, 20, 30, 50$ were studied. Different prediction models and performance metrics were tested separately to analyze the consistency of the conclusions. The analysis results are presented in Fig. 6, and the following conclusions can be drawn:

- By comparing the results for various prediction models (i.e., different rows in Fig. 6), it is evident that the distribution of feature correlation is similar. This similarity enables us to draw common conclusions that are applicable to the different prediction algorithms employed.

TABLE II
 COMPARISON OF TRAJECTORY PREDICTION ERRORS¹

Dataset	GRIP++			Trajectron++ (ml)		Trajectron++ (mul)
	ADE/FDE	ADE _{MCD} /FDE _{MCD}	ADE _{DE} /FDE _{DE}	ADE/FDE	ADE _{DE} /FDE _{DE}	minADE/minFDE
SinD	0.404±0.002 / 0.865±0.005	0.477 / 1.021	0.389 / 0.832	0.567±0.012 / 1.177±0.025	0.543 / 1.125	0.319±0.002 / 0.599±0.006
VA	0.652±0.006 / 1.406±0.020	0.651 / 1.385	0.615 / 1.327	0.619±0.026 / 1.382±0.051	0.557 / 1.261	0.281±0.003 / 0.577±0.010
EP0	0.812±0.007 / 1.828±0.018	1.053 / 2.393	0.745 / 1.678	0.964±0.147 / 2.173±0.315	0.866 / 1.959	0.347±0.004 / 0.672±0.007
EP1	0.850±0.024 / 1.909±0.057	1.173 / 2.640	0.775 / 1.743	0.939±0.074 / 2.101±0.147	0.834 / 1.866	0.392±0.010 / 0.776±0.019
MA	0.867±0.007 / 2.002±0.017	1.083 / 2.546	0.811 / 1.879	0.984±0.128 / 2.379±0.397	0.816 / 1.930	0.424±0.018 / 0.840±0.034
GL	0.624±0.004 / 1.458±0.008	0.709 / 1.646	0.591 / 1.386	0.791±0.036 / 1.863±0.076	0.693 / 1.655	0.362±0.006 / 0.760±0.014

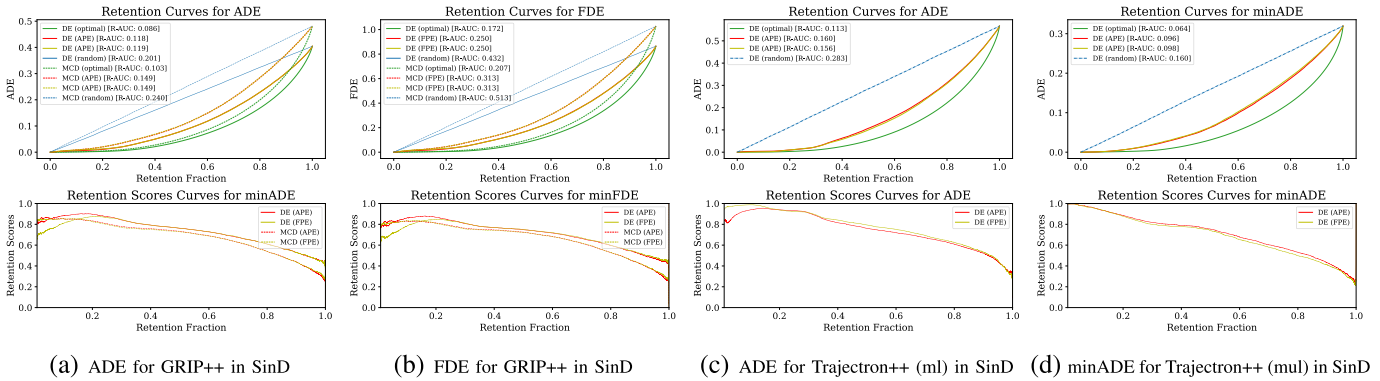


Fig. 5. The (min)ADE/(min)FDE-retention curves (top) and retention scores curves (bottom) on the training and test sets of the SinD dataset. The optimal curve (solid green line) was obtained by replacing the model's predictions with the ground-truth labels in order of decreasing error. Similarly, the random curve (blue dotted line) was obtained by replacing the model's predictions with the ground-truth labels in random order. The red and yellow solid lines correspond to the results captured in order of decreasing APE and decreasing FPE, respectively. The retention scores corresponding to each retention fraction can be calculated by: $(\text{error}_{\text{random}} - \text{error}_{\text{uncertainty}}) / (\text{error}_{\text{random}} - \text{error}_{\text{optimal}})$.

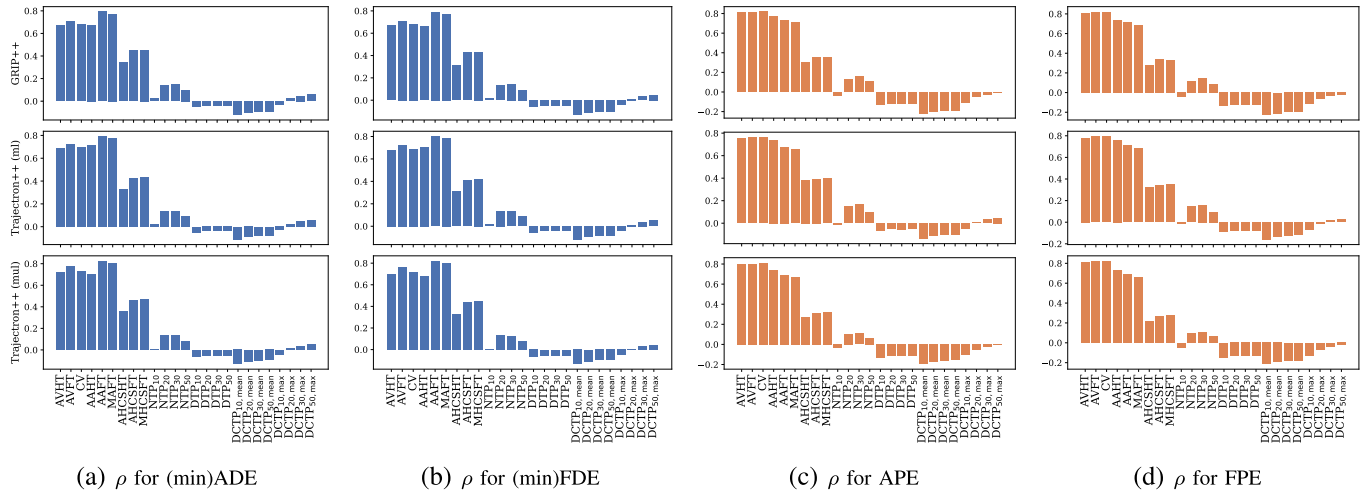


Fig. 6. Comparison of the correlation between prediction model performance and scenario features.

- By comparing the feature correlation distributions for different error metrics (ADE/FDE), a high degree of consistency is observed between them. Besides, similar distribution trends are found from feature correlation analysis for different uncertainty metrics (APE/FPE).
- The comparison of a TA's kinematic features with the features of its surrounding TPs shows that the former had a significant positive correlation with the error and epistemic uncertainty, while the latter shows only weak correlations with such metrics ($-0.2 < \rho < 0.2$).

- The comparison of the correlation between different kinematic features and the prediction error shows that:
 - The correlation-based ranking was as follows: acceleration-related features > velocity-related features > heading change speed-related features. This suggests that changes in a TA's speed or position

¹ADE/FDE/minADE/minFDE represents the error of a single prediction model, obtained from all submodels in the DE-based framework, and the values on the left and right of ± 1 represent the mean and standard deviation, respectively. ADE_{MCD}/DE represents the error of the synthetic trajectory obtained by MCD or DE.

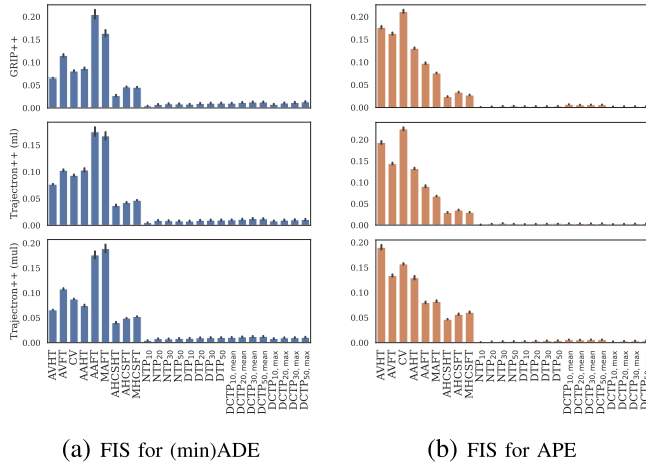


Fig. 7. Comparison of FIS based on the random forest regression.

had a more significant impact on prediction error compared to changes in the TA's movement direction.

- In terms of correlation, features related to future trajectories have a greater impact than features related to historical trajectories. This implies that the prediction model studied in this work had limited adaptability to speed and position mutations occurring at certain points in the future.
- The correlation between different kinematic sub-features of a TA and the epistemic uncertainty showed that the epistemic uncertainty was highly sensitive to the velocity and acceleration features, namely, when a TA was driving at high speed or had a speed mutation, the model tended to show lower confidence in the predictions.

2) *Feature Importance Comparison*: As mentioned above, the feature correlation analysis only shows whether the relationship between two variables conforms to the order consistency. Therefore, the feature importance analysis experiment was performed based on the random forest regression to explore whether there were other types of correlation between the above scenario features and the prediction model. The datasets and training settings used for the prediction algorithm were the same as in the feature correlation analysis. The grid-based search was employed to obtain multiple random forest regression model, then the comprehensive feature importance analysis was performed. The results are presented in Fig. 7.

As shown in Fig. 7, the distribution trend of FIS remains largely consistent across various hyperparameter settings, and is similar to the results obtained from feature correlation analysis. For instance, the features obtained from the surrounding TPs had little effect on the prediction error and epistemic uncertainty uniformly. In contrast, the kinematic features of a TA had a stronger influence, where velocity- and acceleration-related features had higher importance. In random forest regression for the ADE, the AAFT had the highest importance, while in the random forest regression for the APE, the CV had the strongest influence.

3) *Other Scenario Features Analysis*: Apart from the aforementioned features, the impacts of several other scenario features on the prediction performance were explored, including

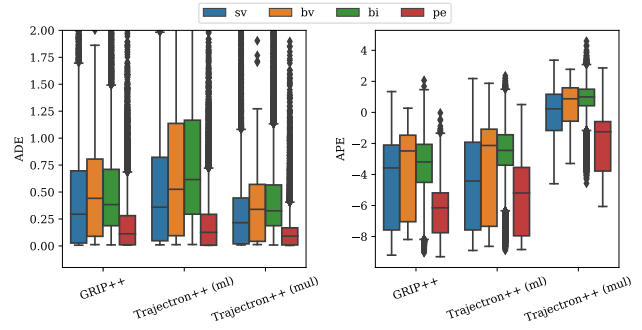


Fig. 8. The results of the trajectory prediction error and epistemic uncertainty of different TA types; sv: small vehicle, bv: large vehicle; bi: motorcyclist or bicyclist; pe: pedestrian.

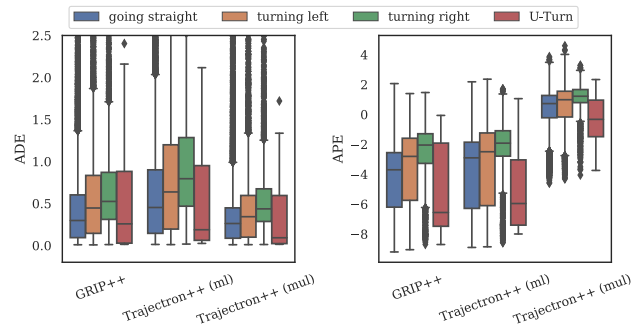


Fig. 9. The results of the trajectory prediction error and epistemic uncertainty under different behavioral patterns.

the type, behavior patterns, compliance with traffic rules, and location of a TA. Without loss of generality, the ADE was adopted as an error metric, and the APE was used for epistemic uncertainty quantification. The prediction model was trained on the SinD training set and analyzed on the SinD test set

Fig. 8 presents the clear distinctions in the distribution of prediction errors among various TAs. Despite the high degree of freedom and randomness in pedestrian movement, their speed and acceleration were typically low, resulting in small prediction errors. Moreover, the epistemic uncertainty distribution for different TA types exhibited a significant resemblance to the prediction error distribution. However, the APE based on Trajectron++ (mul) is higher than the results from GRIP++ and Trajectron++ (ml) primarily due to its derivation from a combination of DE and multimodal predictions.

Fig. 9 illustrates the prediction error and epistemic uncertainty for various TA behavior patterns. The results indicate that the prediction model tends to exhibit larger errors when the TA is turning left or right, compared to the going-straight behavior. Notably, the error in the right-turn scenario was relatively large, possibly due to the large curvature of the right-turn trajectory and the higher driving speed resulting from less impact from traffic lights and other TPs when turning right, compared to turning left or going straight. Although U-turns represent a typical corner case, the overall error in this pattern was small, which may be attributed to the generally low speed during the U-turning process. Additionally, the epistemic uncertainty distributions under different behavioral patterns were consistent with the error distributions.

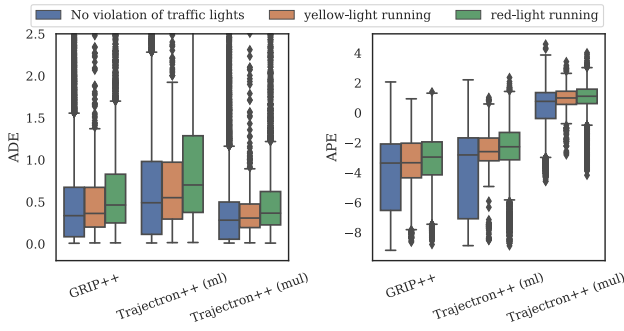


Fig. 10. The results of the trajectory prediction error and epistemic uncertainty under different traffic rule compliance.

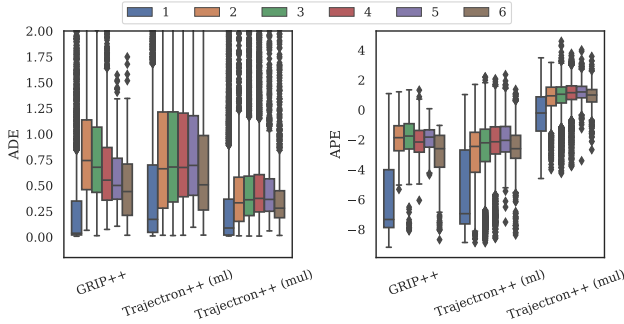


Fig. 11. The results of the trajectory prediction error and epistemic uncertainty at different locations; 1: ex-entering the intersection; 2: in the gap; 3: in the first crosswalk; 4: inside the intersection; 5: in the last crosswalk; and 6: exiting the intersection.

The relationship between the vehicles' compliance with traffic lights and the trajectory prediction performance is presented in Fig. 10, where it can be seen that the prediction error was larger when the vehicle ran red or yellow light than when there was no violation of traffic lights, and simultaneously the proposed model could output higher epistemic uncertainty.

Fig. 11 shows the impact of TA's location on prediction performance. When the vehicle was in the gap area or first crosswalk before entering the intersection, there were multiple strategic options, such as whether to enter the intersection and how to pass through it, which increased prediction complexity and resulted in higher prediction error and epistemic uncertainty. When the vehicle was inside the intersection, the prediction model exhibited considerable error and uncertainty due to numerous interactions with other TPs and higher freedom of movement. In contrast, before entering and after leaving the intersection, vehicles mainly followed the lane, resulting in lower prediction complexity and behavioral diversity, and thus lower prediction error and uncertainty during these stages.

C. Cross-Scenario Prediction Evaluation

Different intersection datasets were collected at different times and locations, and the corresponding environmental conditions might be relatively different, resulting in shifts in data distribution. The distributions of velocity, acceleration, heading, and HCS of objects in six intersection datasets are presented in Fig. 12, where it can be seen that there

were obvious differences in the trajectory features between the datasets. For instance, the velocity and acceleration of a portion of trajectories in the SinD dataset were concentrated around zero. One of the main reasons was the stopping of vehicles and pedestrians while waiting for the green light. Furthermore, the velocity in the SinD dataset exhibited a distinct multimodal distribution, which could be related to the multiple movement patterns caused by various TPs in the dataset. In contrast, the velocity and acceleration in the GL, MA, and VA datasets tended to be higher, reflecting more aggressive motions in these datasets. In addition, distributional shifts could also be observed by comparing the distribution of heading and its speed of change in different datasets.

The aforementioned differences in data distribution between datasets may pose application challenges of the trajectory prediction algorithms in real-world environments. Therefore, the cross-scenario experiments were performed. The approach involved training DE-based prediction models on each of the six training datasets mentioned previously and evaluating them on all six test datasets. As shown in Fig. 13, distributional shifts in the real-traffic environments had a strong impact on trajectory prediction performance. As shown in Fig. 13, even for the same type of scenario, models trained on one intersection dataset struggle to generalize to other datasets of the same type. For instance, for the results for the SinD test dataset, the models based on GRIP/Trajectron++ (ml)/Trajectron++ (mul) that achieved the best accuracy was trained on the SinD training dataset, while their prediction error on non-SinD datasets increased by 116.9%/97.7%/85.4% on average and 258.8%/147.0%/133.8% in the worst case. In addition, the experimental results can also provide some reference for quantifying the allocation offset between different datasets. For example, the cross-scenario test between EP0 and EP1 showed a small deterioration in accuracy, which was related to the similar behavior pattern and base map caused by the adjacency of the two intersections.

Fig. 13 demonstrates the promising performance of DE-based synthesis trajectory in improving the robustness of the prediction model. Compared to the single-model-based ADE, ADE_{DE} based on GRIP++/Trajectron++ (ml) has an average improvement of 6.4%/10.8% in the same-intersection test and an average improvement of 6.3%/10.8% in the cross-scenario test. Furthermore, The extracted epistemic uncertainty could indicate distributional shifts between different datasets. Particularly, the proposed model tend to output higher APE in the face of large distributional shifts. It should be noted that the calculation of APE based on Trajectron++ (mul) takes into account the modeling of motion uncertainty by a single model, thus it is not only related to distributional shifts.

Overall, the models fitted on VA exhibit the worst generalization ability, resulting in an average increase of 152.7%/65.8%/82.8% in prediction errors for GRIP/Trajectron++ (ml)/Trajectron++ (mul). On one hand, there is a limited amount of data available for training in VA. On the other hand, the movement trajectories in VA may not be complex enough, which could also be

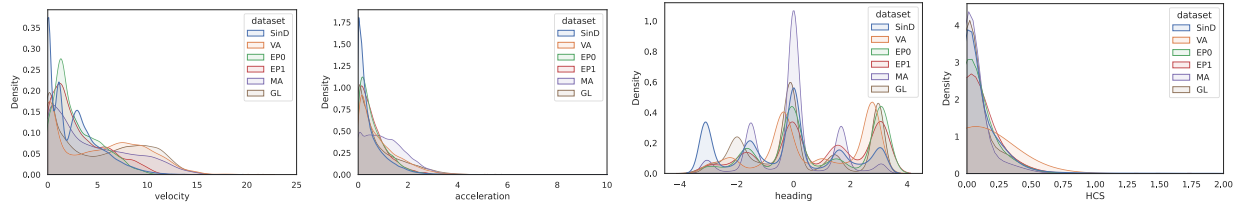


Fig. 12. Comparison of feature distribution at different intersections.

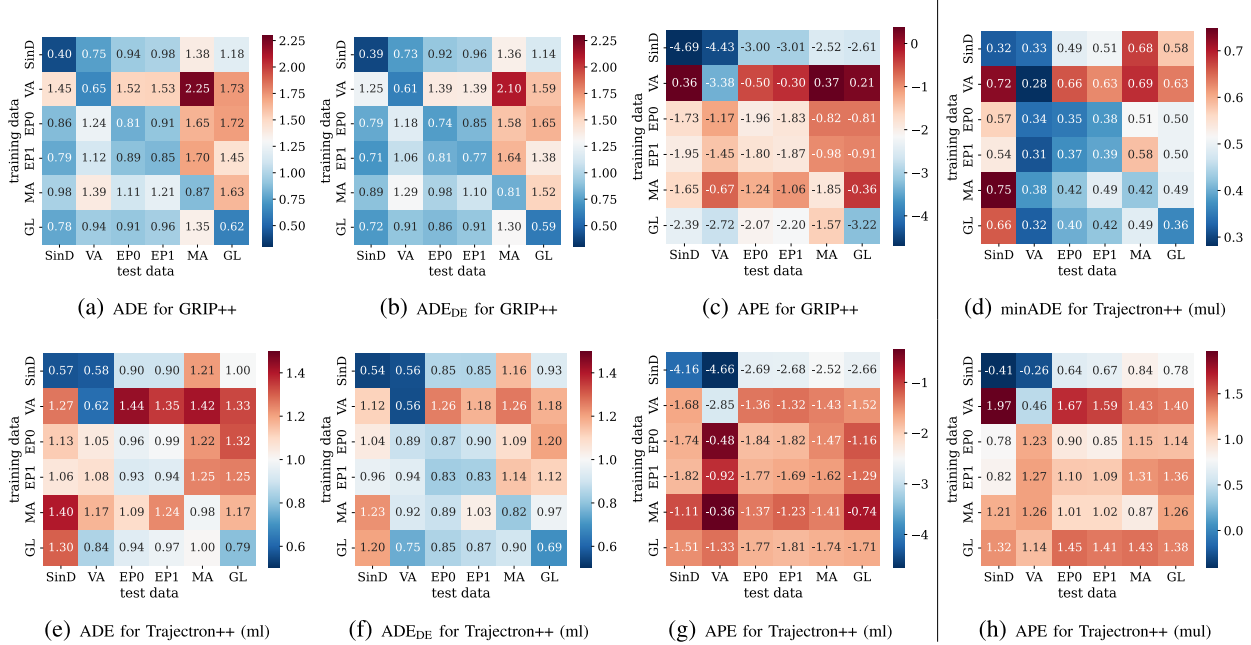
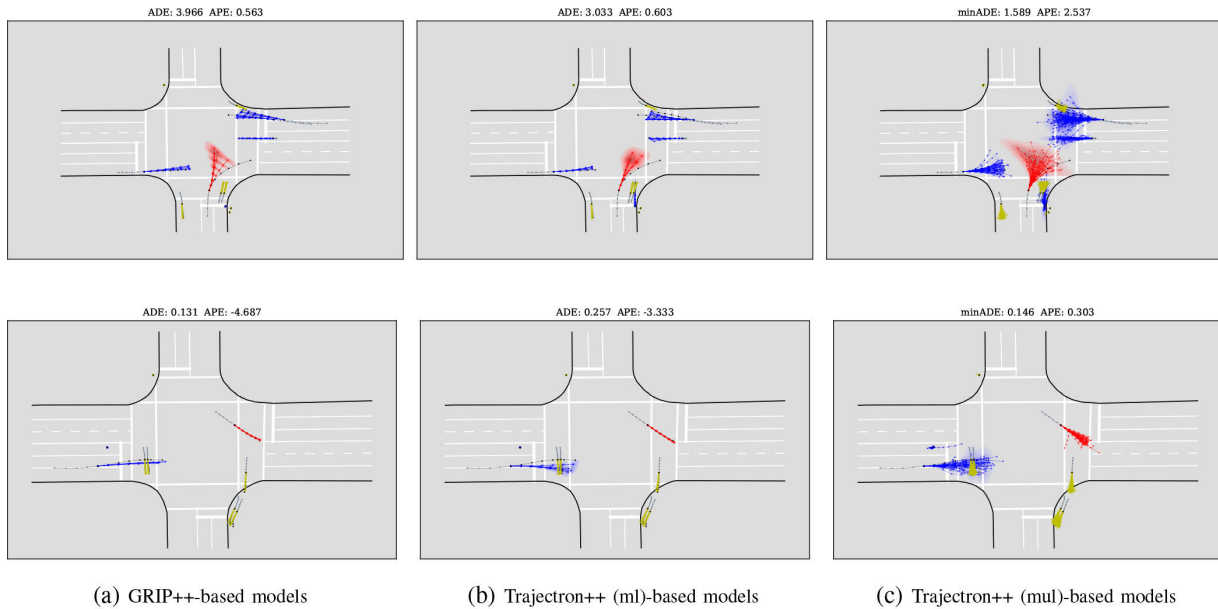
Fig. 13. The cross-scenario prediction error and uncertainty matrix. The i th row and j th column of each matrix represent the results obtained by evaluating the model on the test subset at intersection j after training it on the training subset at intersection i .

Fig. 14. The qualitative results for trajectory prediction and uncertainty estimation. The historical trajectory is depicted by the slate gray thick solid line, while the true future trajectory is represented by the black thin solid line. The predicted trajectories for different types of TAs are denoted by different colors: blue for vehicle and yellow for pedestrian. Furthermore, the object of interest is highlighted in red in each subgraph, and the prediction error and uncertainty estimation results for the object are presented above the subgraph.

a significant contributing factor to the lower error rates observed when testing different models on VA than the other datasets. In contrast, the SinD training dataset provides better

generalization ability for prediction models due to its larger and more diverse dataset, as well as a wider variety of traffic participants and movement patterns.

D. Qualitative Results

Fig. 14 presents the prediction results of the proposed framework, with objects of interest highlighted in red. As shown in the first row, when the TA enters the intersection, the prediction error is significant due to the lack of accurate estimation of TA's right turn intention, with a considerable level of estimated uncertainty. It should be noted that Trajectron++ (mul)-based models, which also models motion uncertainty, exhibits a significantly higher APE compared to the results obtained from GRIP++-based and Trajectron++ (ml)-based models. Observing the second row, it can be seen that in some relatively simple scenarios, both the prediction error and epistemic uncertainty are relatively small. However, Trajectron++ (mul) still models different motion patterns, resulting in a relatively higher APE.

VI. CONCLUSION

This paper proposes a trajectory prediction framework that integrates epistemic uncertainty estimation and studies the effects of traffic environment on prediction performance. Various representative trajectory prediction algorithms are adopted and improved to demonstrate the validity of the proposed framework and the reliability of the analysis results. Several key scenario features are designed and their influences on prediction performance are examined through feature correlation and feature importance analyses. Additionally, the distributional shifts between different intersection datasets and resulting performance degradation of the prediction model are analyzed. Based on the results, the following conclusions are drawn:

(1) The extracted epistemic uncertainty represents the model's confidence in its current predictions and has the potential to identify unknown scenarios where the model may be underpowered. The DE-based framework performs significantly better in estimating epistemic uncertainty and improving trajectory prediction robustness compared to the MCD-based method.

(2) The feature correlation and feature importance analyses show that TA's kinematic features have a positive correlation with prediction performance. Higher velocity, acceleration, and speed of heading change pose a greater challenge to the prediction process, resulting in higher epistemic uncertainty. However, one interesting finding is that the features of surrounding TPs, which reflect the complexity of interactions in the scenario, show little impact on the proposed prediction model performance. Additionally, the prediction model's error and uncertainty vary with other scenario features such as TA's type, behavior pattern, compliance with traffic rules, and location. The conducted analyses are helpful in locating the limitations in the prediction algorithms, thus providing guidance for the improvement of autonomous driving functions.

(3) Different intersection datasets show distributional shifts due to local driving habits, road structures, and national cultures, posing challenges to prediction algorithms. However, the proposed DE-framework improves trajectory prediction robustness, and the extracted epistemic uncertainty

responds to reduced confidence of the model in a new environment, improving the self-awareness ability of autonomous driving.

Although the used basic prediction algorithms have strong representativeness, it is still difficult to completely avoid the specificity of analysis conclusions. However, the proposed method shows promise for analyzing other prediction algorithms, and subsequent work could consider combining more types of algorithms for systematic analysis. Furthermore, the role of algorithms uncertainty estimation of a trajectory prediction model in an autonomous driving decision-making mechanism needs to be further studied to improve robustness against the risks of insufficient functions.

REFERENCES

- [1] K. Xu, X. Xiao, J. Miao, and Q. Luo, "Data driven prediction architecture for autonomous driving and its application on Apollo platform," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Oct. 2020, pp. 175–181.
- [2] X. Mo, Z. Huang, Y. Xing, and C. Lv, "Multi-agent trajectory prediction with heterogeneous edge-enhanced graph attention network," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 9554–9567, Jul. 2022.
- [3] A. Rudenko, L. Palmieri, M. Herman, K. M. Kitani, D. M. Gavrila, and K. O. Arras, "Human motion trajectory prediction: A survey," *Int. J. Robot. Res.*, vol. 39, no. 8, pp. 895–935, Jul. 2020, doi: [10.1177/0278364920917446](https://doi.org/10.1177/0278364920917446).
- [4] J. Gu, C. Sun, and H. Zhao, "DenseTNT: End-to-end trajectory prediction from dense goal sets," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 15303–15312.
- [5] X. Li, X. Ying, and M. Choo Chuah, "GRIP++: Enhanced graph-based interaction-aware trajectory prediction for autonomous driving," 2019, *arXiv:1907.07792*.
- [6] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, "Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data," in *Computer Vision—ECCV 2020 (Lecture Notes in Computer Science)*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham, Switzerland: Springer, 2020, pp. 683–700.
- [7] T. Gilles, S. Sabatini, D. Tsishkou, B. Stanculescu, and F. Moutarde, "Uncertainty estimation for cross-dataset performance in trajectory prediction," 2022, *arXiv:2205.07310*.
- [8] J. Gawlikowski et al., "A survey of uncertainty in deep neural networks," 2021, *arXiv:2107.03342*.
- [9] L. Peng, H. Wang, and J. Li, "Uncertainty evaluation of object detection algorithms for autonomous vehicles," *Automot. Innov.*, vol. 4, no. 3, pp. 241–252, Aug. 2021, doi: [10.1007/s42154-021-00154-0](https://doi.org/10.1007/s42154-021-00154-0).
- [10] S. Lefèvre, D. Vasquez, and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," *ROBOMECH J.*, vol. 1, no. 1, pp. 1–10, Jul. 2014, doi: [10.1186/s40648-014-0001-z](https://doi.org/10.1186/s40648-014-0001-z).
- [11] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social LSTM: Human trajectory prediction in crowded spaces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 961–971.
- [12] N. Deo and M. M. Trivedi, "Convolutional social pooling for vehicle trajectory prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 1468–1476.
- [13] F. Diehl, T. Brunner, M. T. Le, and A. Knoll, "Graph neural networks for modelling traffic participant interaction," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2019, pp. 695–701.
- [14] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social GAN: Socially acceptable trajectories with generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2255–2264.
- [15] J. Liu, X. Mao, Y. Fang, D. Zhu, and M. Q.-H. Meng, "A survey on deep-learning approaches for vehicle trajectory prediction in autonomous driving," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2021, pp. 978–985.
- [16] X. Li, X. Ying, and M. C. Chuah, "GRIP: Graph-based interaction-aware trajectory prediction," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 3960–3966.

- [17] L. Zhang, W. Xiao, Z. Zhang, and D. Meng, "Surrounding vehicles motion prediction for risk assessment and motion planning of autonomous vehicle in highway scenarios," *IEEE Access*, vol. 8, pp. 209356–209376, 2020.
- [18] B. Wilson et al., "Argoverse 2: Next generation datasets for self-driving perception and forecasting," 2023, *arXiv:2301.00493*.
- [19] W. Zhan et al., "Interaction dataset: An international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps," 2019, *arXiv:1910.03088*.
- [20] F. Zheng et al., "Unlimited neighborhood interaction for heterogeneous trajectory prediction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 13168–13177.
- [21] R. Chandra, U. Bhattacharya, A. Bera, and D. Manocha, "TraPHic: Trajectory prediction in dense and heterogeneous traffic using weighted interactions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jul. 2019, pp. 8483–8492.
- [22] N. Deo, E. Wolff, and O. Beijbom, "Multimodal trajectory prediction conditioned on lane-graph traversals," in *Proc. 5th Conf. Robot Learn.*, Jan. 2022, pp. 203–212.
- [23] Y. Ma, X. Zhu, S. Zhang, R. Yang, W. Wang, and D. Manocha, "TrafficPredict: Trajectory prediction for heterogeneous traffic-agents," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, Feb. 2019, vol. 33, no. 1, pp. 6120–6127.
- [24] D. Ulmer, C. Hardmeier, and J. Frellsen, "Prior and posterior networks: A survey on evidential deep learning methods for uncertainty estimation," 2021, *arXiv:2110.03051*.
- [25] Z. Nado et al., "Uncertainty baselines: Benchmarks for uncertainty & robustness in deep learning," 2021, *arXiv:2106.04015*.
- [26] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural network," in *Proc. Int. Conf. Mach. Learn.*, Jun. 2015, pp. 1613–1622.
- [27] G. E. Hinton and D. van Camp, "Keeping the neural networks simple by minimizing the description length of the weights," in *Proc. 6th Annu. Conf. Comput. Learn. Theory*, 1993, pp. 5–13.
- [28] C. Louizos, K. Ullrich, and M. Welling, "Bayesian compression for deep learning," in *Advances in Neural Information Processing Systems*, vol. 30. Red Hook, NY, USA: Curran Associates, 2017.
- [29] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *Proc. Int. Conf. Mach. Learn.*, Jun. 2016, pp. 1050–1059.
- [30] Y. Gal, J. Hron, and A. Kendall, "Concrete dropout," in *Advances in Neural Information Processing Systems*, vol. 30. Red Hook, NY, USA: Curran Associates, 2017.
- [31] M. Sensoy, L. Kaplan, and M. Kandemir, "Evidential deep learning to quantify classification uncertainty," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 1–12.
- [32] A. Amini, W. Schwarting, A. Soleimany, and D. Rus, "Deep evidential regression," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 14927–14937.
- [33] A. Malinin and M. Gales, "Predictive uncertainty estimation via prior networks," in *Advances in Neural Information Processing Systems*, vol. 31. Red Hook, NY, USA: Curran Associates, 2018.
- [34] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Advances in Neural Information Processing Systems*, vol. 30. Red Hook, NY, USA: Curran Associates, 2017.
- [35] J. Snoek et al., "Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, p. 12.
- [36] Y. Wen, D. Tran, and J. Ba, "BatchEnsemble: An alternative approach to efficient ensemble and lifelong learning," 2020, *arXiv:2002.06715*.
- [37] F. Wenzel, J. Snoek, D. Tran, and R. Jenatton, "Hyperparameter ensembles for robustness and uncertainty quantification," in *Advances in Neural Information Processing Systems*, vol. 33. Red Hook, NY, USA: Curran Associates, 2020, pp. 6514–6527.
- [38] Z. Yang et al., "EdgeDuet: Tiling small object detection for edge assisted autonomous mobile vision," *IEEE/ACM Trans. Netw.*, early access, Dec. 2, 2022, doi: [10.1109/TNET.2022.3223412](https://doi.org/10.1109/TNET.2022.3223412).
- [39] Z. Zhou, M. Shojafar, R. Li, and R. Tafazolli, "EVCT: An efficient VM deployment algorithm for a software-defined data center in a connected and autonomous vehicle environment," *IEEE Trans. Green Commun. Netw.*, vol. 6, no. 3, pp. 1532–1542, Sep. 2022.
- [40] C. Neurohr, L. Westhofen, M. Butz, M. H. Bollmann, U. Eberle, and R. Galbas, "Criticality analysis for the verification and validation of automated vehicles," *IEEE Access*, vol. 9, pp. 18016–18041, 2021.
- [41] G. Bagschik, T. Menzel, and M. Maurer, "Ontology based scene creation for the development of automated vehicles," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 1813–1820.
- [42] J. Wang, C. Zhang, Y. Liu, and Q. Zhang, "Traffic sensory data classification by quantifying scenario complexity," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 1543–1548.
- [43] O. Makansi et al., "You mostly walk alone: Analyzing feature attribution in trajectory prediction," 2021, *arXiv:2110.05304*.
- [44] I. Hasan, S. Liao, J. Li, S. U. Akram, and L. Shao, "Pedestrian detection: Domain generalization, CNNs, transformers and beyond," 2022, *arXiv:2201.03176*.
- [45] O. Styles, T. Guha, and V. Sanchez, "Multiple object forecasting: Predicting future object locations in diverse environments," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2020, pp. 690–699.
- [46] J. Gesnouin, S. Pechberti, B. Stanculescu, and F. Moutarde, "Assessing cross-dataset generalization of pedestrian crossing predictors," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2022, pp. 419–426.
- [47] Y. Xu et al., "SIND: A drone dataset at signalized intersection in China," in *Proc. IEEE 25th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2022, pp. 2471–2478.
- [48] A. Malinin et al., "Shifts: A dataset of real distributional shift across multiple large-scale tasks," 2021, *arXiv:2107.07455*.
- [49] A. Malinin. (Oct. 2019). *Uncertainty Estimation in Deep Learning with application to Spoken Language Assessment*. University of Cambridge. Accepted: Nov. 13, 2019. [Online]. Available: <https://www.repository.cam.ac.uk/handle/1810/298857>
- [50] F. Poggenhans et al., "Lanelet2: A high-definition map framework for the future of automated driving," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 1672–1679.



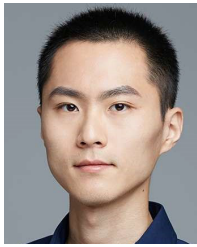
Wenbo Shao received the B.E. degree in vehicle engineering from Tsinghua University, Beijing, China, in 2019, where he is currently pursuing the Ph.D. degree in mechanical engineering. He is a member of the Tsinghua Intelligent Vehicle Design and Safety Research Institute (IVDAS) and supervised by Prof. Jun Li and Assoc. Prof. Hong Wang. His research interests include safety of the intended functionality of autonomous driving, trajectory prediction, decision-making, and uncertainty theory and applications.



Yanchao Xu received the B.E. degree in vehicle engineering from Hainan University, China, in 2020. He is currently pursuing the M.S. degree in mechanical engineering with the Beijing Institute of Technology. He is also one of the visiting students at the Tsinghua Intelligent Vehicle Design and Safety Research Institute (IVDAS) since 2020. His research interests include prediction, trajectory data mining, and scenario parameterization for autonomous driving.



Jun Li received the Ph.D. degree in vehicle engineering from Jilin University, Changchun, Jilin, China, in 1989. He is currently an Academician with the Chinese Academy of Engineering and a Professor with the School of Vehicle and Mobility, Tsinghua University. His research interests include internal combustion engine, electric drive systems, electric vehicles, intelligent vehicles, and connected vehicles. He is the President of the Society of Automotive Engineers of China and the Director of the Expert Committee of China Industry Innovation Alliance for the Intelligent and Connected Vehicles.



Chen Lv (Senior Member, IEEE) received the Ph.D. degree from the Department of Automotive Engineering, Tsinghua University, China, in 2016. From 2014 to 2015, he was a Joint Ph.D. Researcher with the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, Berkeley, CA, USA. From 2016 to 2018, he was a Research Fellow with the Advanced Vehicle Engineering Center, Cranfield University, U.K. He is currently an Assistant Professor with the School of Mechanical and Aerospace Engineering, and the Cluster Director of Future Mobility Solutions at ERI@N, Nanyang Technological University, Singapore. His research interests include advanced vehicles and human-machine systems, where he has contributed over 100 articles and obtained 12 granted patents.



Hong Wang (Senior Member, IEEE) received the Ph.D. degree from the Beijing Institute of Technology, China, in 2015. From 2015 to 2019, she was a Research Associate of mechanical and mechatronics engineering with the University of Waterloo. She is currently a Research Associate Professor with Tsinghua University. She has published more than 60 papers on top international journals. Her research interests include the safety of the on-board AI algorithm, the safe decision-making for intelligent vehicles, and the test and evaluation of SOTIF. She serves as an Associate Editor for IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, and IEEE TRANSACTIONS ON INTELLIGENT VEHICLES.



Weida Wang (Senior Member, IEEE) received the Ph.D. degree from Beihang University, Beijing, China, in 2009. He is currently a Professor with the School of Mechanical Engineering, Beijing Institute of Technology, Beijing. He is also the Director of the Research Institute of Special Vehicle, Beijing Institute of Technology. His current research interests include electric vehicle, automated vehicle motion planning and control, and electromechanical transmission control.