Interactive Behavior Modeling for Vulnerable Road Users With Risk-Taking Styles in Urban Scenarios: A Heterogeneous Graph Learning Approach

Zirui Li, Jianwei Gong, *Member, IEEE*, Zheyu Zhang, *Graduate Student Member, IEEE*, Chao Lu, *Member, IEEE*, Victor L. Knoop, and Meng Wang, *Member, IEEE*

***Abstract*— The deep understanding of the behaviors of traffic participants is essential to guarantee the safety of automated vehicles (AV) in mixed traffic with vulnerable road users (VRUs). Precise trajectory prediction of traffic participants can provide reasonable solution space for motion planning of AV. Early works mainly focused on handcrafting the feature representation and designing complicated architectures in deep learning-based prediction models. However, these approaches overlooked the fact that different road users perceive the safety of the same interaction differently and also exhibit heterogeneous risk-taking styles. In this paper, we will develop a model for trajectory prediction based on risk-taking styles. The model accounts for the expected positions and occupancy of traffic participants in the surrounding environment. It consists of two sequential steps: risk-taking styles of multi-modal road users under interactive scenes are first clustered, and then reformulated in the heterogeneous graph model for trajectory prediction. The model is validated by the driving data collected on the urban road using a public dataset. Comparative experiments demonstrate that the proposed method can predict the trajectory of traffic participants much more accurately than the state-of-the-art methods.**

***Index Terms*— Vulnerable road users, risk-taking behaviors, interactive behavior modeling, heterogeneous graph model, trajectory prediction.**

I. Introduction

A. Motivation

MODELING traffic participants' behaviors in complex urban scenarios is essential for decision-making and motion planning of automated driving systems [1]. Such models will enable automated vehicles (AVs) to predict future behaviors of traffic participants [2]. Accurately representing interactions is essential for AV to achieve safe and socially acceptable motion planning. However, due to the uncertainty of traffic participants and the complex urban environment, this precise representation of interactions still remains a challenge. Several works focus on the issue and develop numerous models.

Physics-based motion models, that represent vehicles as dynamic entitie governed by the laws of physics, were first applied to predict future motion based on constraints from dynamic and kinematic models. For short-term prediction with a time horizon of less than 0.5 seconds, it may be practical to predict the trajectories of vehicles based on physical models. However, pure physical models cannot deal with the uncertainty of riders and pedestrians in the prediction longer than 1 second [3]. A few studies proposed the so-called "two-stage" framework to recognize the intentions of traffic participants first, and then predict the trajectory in each intention mode. For example, on a freeway, intentions of the target vehicle are defined as left lane-changing, right lane-changing and lane-keeping [4]. However, these approaches cannot simultaneously consider interactions between traffic participants and predict future behaviors. Recent works try to address the above issues by modeling interactions between participants with deep neural networks, which are named "social-aware" methods [5], [6]. However, most of these models apply a pooling mechanism or concatenation operations to directly fuse features from interactive participants without interpretability. Moreover, these methods cannot model "high-level" interactions (indirect influence), which are beyond adjacent entities and cannot be explicitly modeled with relational reasoning.

Therefore, in this research, we take a step forward to model the spatial and temporal influence between traffic participants. It aims to represent spatial interactions with the graph structure, while the relationship in the temporal domain is described by long short-term memory (LSTM). The work most closely related to this research is Social Attention [6], which can simultaneously model a number of pedestrians and accurately predict future behaviors. Considering the heterogeneity in traffic participants (riders, vehicles and pedestrians), each category perceives the risk of the same interactive scenario differently and reflects heterogeneous preferences. In this study, these preferences are named risk-taking styles. Therefore, in this

Manuscript received 21 February 2022; revised 17 August 2023 and 3 March 2024; accepted 24 April 2024. Date of publication 22 May 2024; date of current version 1 August 2024. This work was supported by the National Natural Science Foundation of China under Grant 52372405 and Grant 52272411. The Associate Editor for this article was S. P. Raja. *(Corresponding author: Chao Lu.)*

Zirui Li, Jianwei Gong, and Chao Lu are with the School of Mechanical Engineering, Beijing Institute of Technology, Beijing 100081, China (e-mail: 3120195255@bit.edu.cn; gongjianwei@bit.edu.cn; chaolu@bit.edu.cn).

Zheyu Zhang was with the School of Mechanical Engineering, Beijing Institute of Technology, Beijing 100081, China. She is now with the School of Aeronautical and Automotive Engineering, Loughborough University, LE11 3TU Loughborough, U.K. (e-mail: z.zhang8@lboro.ac.uk).

Victor L. Knoop is with the Department of Transport and Planning, Faculty of Civil Engineering and Geosciences, Delft University of Technology, 2628 CD Delft, The Netherlands (e-mail: v.l.knoop@tudelft.nl).

Meng Wang is with the Chair of Traffic Process Automation, "Friedrich List" Faculty of Transport and Traffic Sciences, TU Dresden, 01069 Dresden, Germany (e-mail: meng.wang@tu-dresden.de).

Digital Object Identifier 10.1109/TITS.2024.3399481

research, besides the behavior prediction in interactive scenarios, the risk-taking styles of different participants are also analyzed. An interactive behavior model with risk-taking styles is proposed for trajectory prediction. Four kinds of styles in perceiving risk behaviors are first generated. Then, specific risk-taking styles are re-formulated in the prediction module to capture spatial-temporal interactions and predict trajectories of heterogeneous traffic participants.

B. Related Works

The motivation of interaction-aware methods is to model the influence among traffic participants. Reference [7] showed a microscopic traffic interaction model of driving behaviors inside the intersections. It optimized the interactive behaviors in two-dimensional space by considering various constraints. Based on game theory, [8] and [9] proposed to recognize drivers' intentions in lane-changing scenarios by formulating the problem as a non-cooperative Stackelberg Game problem. References [10] and [11] developed game theory-based lane-changing models in the connected environment. Reference [12] proposed a virtual-game-based interaction model to recognize and comprehend sociality, encompassing both implicit social norms and individualized social preferences of human drivers. Based on [12], an active motion planner was developed to improve the active interaction capability of the automated vehicle [13]. Reference [14] applied game theory in estimating the social gap in the discretionary lane-changing game and offering solutions to reduce the impact of disturbances caused by inappropriate lane changes. The game theory-based interaction model was also widely used in decision-making. A game-theoretic decision-making method was shown in [15] to resolve the driving conflict and improve the safety and efficiency of autonomous vehicles at unsignalized intersections. The similar solution was applied in helping automated vehicles with left-turn maneuvers at intersections [16]. However, most of game theory-based methods focused on the interactions between vehicles.

Based on an extension of LSTM, [5] firstly proposed Social LSTM to model the interaction between pedestrians as social behaviors. The "social" represents the impact for one pedestrian from neighbors. With the development of Generative Adversarial Networks (GAN), Social GAN was developed by applying Social LSTM in the generator of GAN [17], which outperformed the prior work in reducing the predicted error. Some extended works were conducted based on Social LSTM by taking context information and complex constraints into consideration [18], [19]. In [18], static features from the scene context were combined in the convolutional encoding stream by the proposed multi-agent tensor fusion (MATF) network. However, the degree of interactions was less quantitatively modeled in previous works. Some researchers propose to construct interactions of traffic participants as a graph structure, which is described by nodes and edges. In [6], a spatial-temporal graph neural network (ST-GNN) named social attention was proposed by modeling graph nodes as LSTMs, which obtained better performance in the trajectory prediction compared to social LSTM. Keeping in mind that heterogeneous participants exist in the urban road, TrafficPredict developed a novel category layer to describe different kinds of participants [20]. It obtains a better performance compared to the model for homogeneous participants. In this research, we address the issue of multi-agent interactions by the heterogeneous ST-GNN with both spatial-temporal attention mechanisms to capture the indirect interaction and predict future behaviors.

Various methods mentioned above pay more attention to the development of the architecture to improve the predicted accuracy. Fewer studies focus on applying risk-taking styles of surroundings in the interactive model. In order to guarantee the safety of automated vehicles in interactive scenarios, some studies proposed to assess the level of risk. It can be used in the selection of strategies for vehicle planning and control. Reference [21] proposed a utility-based approach for drivers' cognitive behaviors in car-following scenarios. It analyzed and models characteristics of safety proxy indices in risk-taking scenarios. In [22], a probabilistic field approach for risk assessment was proposed by artificial field theory in the modeling of driving risk. The probabilistic method quantitatively describes and analyzes the risk level in safety-critical scenarios. Besides the artificial field theory-based method in [22], a lane-based collision risk assessment algorithm was developed in [23]. It incorporated model probability distribution of lanes and time-to-collision to evaluate the risk of collision. Experimental results show compelling risk predictions on the highway. It provides the ego vehicle with sufficient time to react to potential collisions. However, the performance of risk evaluation depends on the accuracy of trajectory prediction, which is difficult to be guaranteed in complex scenarios [24].

Considering the uncertainty in prediction-based risk evaluations, some studies proposed to label risk scenarios manually. Labeled results can be used to train machine learning models for an in-vehicle warning system. In [25], a database for abnormal driving detection was released to detect, localize, and recognize anomalous events from egocentric videos. The manual labeling process in public datasets is time-consuming and relies on subjective judgment. Different characteristics of participants who label the collected data will influence the supervised model. Another issue of this approach is that collected data cannot cover all abnormal events, which are rarely happening and recorded. Some researchers propose to build a rule-based framework to divide risk scenarios into different levels [26]. In the car-following scenario, the rule can be constructed based on the acceleration of the leader and the follower. For example, the leading vehicle decelerates and the following vehicle accelerates. Risk levels can be divided according to different pairs of deceleration and acceleration. Although the rule-based method can cover corner cases by the pre-defined rule set, the rationality of rules and the switch between strategies will influence the performance of vehicle planning and control.

To solve problems in prediction-based and rule-based approaches, unsupervised methods are applied by investigating the distribution of features in driving data [23], [27]. The risk-taking style can be effectively learned and mined by analyzing the characteristics of distributions. In other words,

it can cope with the shortcoming of pre-defined rules. In [28] and [29], an integrated framework is proposed to cluster the sequential driving data into several continuous segments by an unsupervised method (sticky hierarchical Dirichlet process). It can be applied to label risk lane-changing behaviors from the naturalistic driving dataset automatically. Besides the development in detecting risk driver behaviors, the unsupervised method is also applied in the cluster of braking behaviors. In [30], Gaussian mixture model (GMM) is developed to cluster the braking pressure into three categories according to the difference in distributions. It provides a detailed analysis of drivers' braking intensity, which can recognize the level of emergency or risk in braking scenarios. This research considers risk-taking styles in the heterogeneous ST-GNN, which is developed to model interactive behaviors.

C. Contributions

As mentioned, interactive behavior prediction and real-time risk assessment have been widely studied. However, very few studies focus on different risk-taking styles in modeling interactive behaviors. In this paper, we develop a risk-taking style-oriented method to model interactive behaviors. Specifically, A risk-taking style-oriented interactive behavior model is proposed for trajectory prediction in urban scenarios. It is developed based on the unsupervised risk-taking style generation (RSG) and the heterogeneous graph model (HGM). It is a hybrid learning structure, which can sequentially generate the risk-taking style and predict the trajectory. Meanwhile, the proposed model combines intention-based and interaction-aware methods by sequential steps, which obtains advantages from both sides. It firstly clusters and generates the risk-taking style by unsupervised methods and secondly predicts the trajectory by ST-GNN. Compared to our previous works that only focus on the architecture design of the prediction model [31], [32], the proposed model also provides the analysis of the risk-taking style between the ego vehicle and surroundings. The analysis result benefits the risk level identification and the generated risk-taking style can be formulated into ST-GNN to improve the performance of trajectory prediction.

D. Outlines

The remainder of this paper is organized as follows. We first formulate the problem of sequential steps in Section II. Then we describe the unsupervised clustering method for risk behavior analysis in Section III. Section IV then presents the heterogeneous graph neural network in the trajectory prediction. Section V and Section VI present the experimental results for unsupervised risk behaviors analysis and supervised trajectory prediction, respectively. Finally, Section VII presents the conclusion and future work.

II. Problem Formulation and Preliminaries

The overall illustration of the proposed method is shown in Fig.1, which consists of two modules: unsupervised risk-taking style generation (RSG) and supervised heterogeneous graph model (HGM) for trajectory prediction. In the RSG module, data collected in urban scenarios are processed for feature extraction, which includes relative positions, relative velocity and time to collision (TTC). Considering that humans perceive the same interactive scenario with heterogeneous traffic participants differently, the risk-taking style between ego vehicle and surrounding participants need to be appropriately described and accurately modeled. Existing methods use either prediction-based models or manual labeling to evaluate the risk level [22], [25]. These methods are limited by the influence of prediction accuracy and their poor generalization ability. According to [27] and [29], unsupervised learning approaches are successful in modeling the spatial-temporal risk level and patterns for the vehicle-pedestrian interaction. Therefore, the unsupervised method is chosen due to its advantages in investigating the distribution of driving data.

For interactive behavior modeling, the GNN is good at describing the degree of interactions between heterogeneous traffic participants compared to convolutional-based methods [20], [33]. In the HGM, the graph structure is constructed to describe the spatial interaction and the interaction in the temporal domain is modeled by LSTM. With features and specific risk-taking styles from the upper module, the HGM is trained to generate future trajectories of surrounding traffic participants. Finally, the performance of trajectory prediction is evaluated and discussed for comparative study.

1) Unsupervised RSG: In the module of unsupervised RSG, given the historical states input $\mathbf{S}_i^{t_c-M+1:t_c}$ of i^{th} traffic participant, the formulation of RSG can be represented by the following equation:

$$r_{c_i} \leftarrow f_{\mathrm{RSG}}(\theta_c^*; \mathbf{S}_i^{t_c-M+1:t_c}) \tag{1}$$

where $c_i \in \{vehicle, pedestrain, rider\}$ is the category of i^{th} traffic participant and $r_{c_i} \in \{1, 2, \ldots, K_c\}$ is one of K_c risk-taking styles of i^{th} traffic participant. θ_i^* is the optimal parameters of clustering algorithm after the training process. M is the length of historical features and t_c is the current timestep. The unsupervised RSG module f_{RSG} represents the clustering method, which are detailed in section III. Observed features are used to generate clustering results by approximating parameters θ^*, which are sent to the supervised HGM as risk-taking styles for trajectory prediction. Selected features $\mathbf{S}_i^{t_c-M+1:t_c}$ of i^{th} participant can be observed and calculated by the onboard sensors. In this research, three features are selected as inputs of clustering methods [27], [29]: relative position, relative velocity and time to collision (TTC), which are detailed in section IV.

2) Supervised HGM: The general formulation of supervised HGM for N_p participants can be described as the following:

$$\Delta\mathbf{p}^{t_c+1:t_c+N} \leftarrow \mathrm{f}_{\mathrm{HGM}}(\mathbf{W}^*; \Delta\mathbf{p}^{t_c-M+1:t_c}, \mathbf{c}, \mathbf{r}) \tag{2}$$

where $\Delta\mathbf{p}^{t_c+1:t_c+N} = \{\Delta\mathbf{p}_i^{t_c+1:t_c+N} \mid i = 1, 2, 3, \ldots, N_p\}$ and $\Delta\mathbf{p}^{t_c-M+1:t_c} = \{\Delta\mathbf{p}_i^{t_c-M+1:t_c} \mid i = 1, 2, 3, \ldots, N_p\}$ are historical and predicted trajectories for all N_p traffic participants. $\mathbf{c} = \{c_i \mid i = 1, 2, 3, \ldots, N_p\}$ is the type of traffic participant. $\mathbf{r} = \{r_{c_i} \mid i = 1, 2, 3, \ldots, N_p\}$ is the risk taking style in category c_i, which is the time-invariant feature by assuming the style is consistent in a short period. The target of HGM is to accurately predict future trajectory

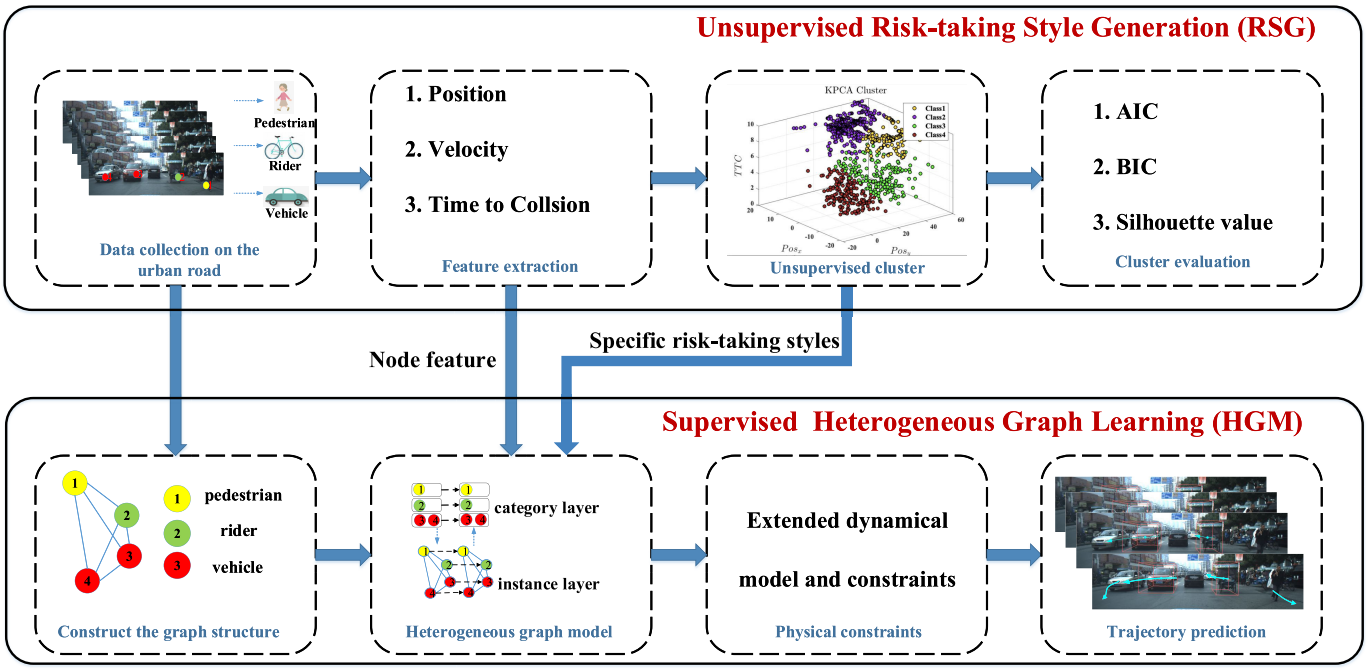


Fig. 1. The overall illustration of the proposed method.

$\Delta \mathbf{p}^{t_c+1:t_c+N}$ based the historical trajectory $\Delta \mathbf{p}^{t_c-M+1:t_c}$ and trained weights $\mathbf{W}^*$ in the GNN. In this research, traffic participants in interactive scenarios are described and modeled by the graph structure, which is the basic unit of GNN.

According to the definition in [20], all traffic participants are modeled as instance nodes, and the participants with the same type are modeled as category nodes to represent the difference of heterogeneous participants. The heterogeneous graph structure G with instance and category layers can be defined as:

$$G = (A_{\text{instance}}, A_{\text{category}}, E_{\text{Spatial}}, E_{\text{Temporal}}) \tag{3}$$

where A_{instance} and A_{category} are instance and category nodes, and E_{Spatial} and E_{Temporal} are spatial and temporal edges:

- At each timestep t, for i^{th} traffic participant, the feature $\mathbf{f}_i^t$ of the instance node A_{instance} is described as

$$\mathbf{f}_i^t = (\Delta x_i^t, \Delta y_i^t, r_{c_i}, c_i) \tag{4}$$

  where $(\Delta x_i^t, \Delta y_i^t)$ is the relative position to the ego vehicle. r_{c_i} is the risk-taking style from the unsupervised RSG module. $c_i \in \{vehicle, pedestrain, rider\}$ is the category of i^{th} traffic participant.
- The category node A_{category} is built to model the similarity of traffic participants with the same type.
- In the constructed spatial-temporal graph, interactions between participants are modeled by spatial edges E_{Spatial} and temporal edges E_{Temporal}. Specifically, at each timestep t, the influence from A_j^t to A_i^t is described as $E_{ij,\text{Spatial}}^t = (A_i^t, A_j^t)$, which can be calculated by the spatial edge attribute $\mathbf{f}_{ij}^t = (\Delta x_{ij}^t, \Delta y_{ij}^t, r_{c_{ij}}, c_{ij})$ with $\Delta x_{ij}^t = \Delta x_i^t - \Delta x_j^t$ and $\Delta y_{ij}^t = \Delta y_i^t - \Delta y_j^t$. The unique encoder is applied to represent $r_{c_{ij}}$ and c_{ij} [20]. The formulation of temporal edges E_{Temporal} is similar to the edge in the spatial domain, which represents the correlation of the same participant in adjacent frames. The attribute of $E_{ii,\text{Temporal}}^t = (A_i^t, A_i^{t+1})$ can be calculated by substituting A_j^t with A_i^{t+1}.

The above definitions focus on describing variables and feature representations in the instance layer. Because traffic participants in urban scenarios are heterogeneous, the category layer is constructed to model similar movement patterns for each type of traffic participant, which can be defined based on variables in the instance layer. All instance nodes with the same type are integrated and embedded to transfer information from the instance to the category layer, which is realized by the edge oriented from the instance node to the category node. After the calculation in the category layer, category nodes send the processed and valuable information from the category to instance nodes. Details of instance and category layers are introduced in Section IV.

III. Unsupervised Risk-Taking Style Generation (RSG)

In this work, we assume that humans perceive the safety of the same interactive scenario differently for heterogeneous traffic participants. With the same kinematics (positions, velocities, accelerations etc.) in a scenario, different types of traffic participants react differently to the ego vehicle. Therefore, they need to be modeled separately. Moreover, among each category of traffic participants, different risk levels need to be distinguished based on relative distance, relative speed, TTC and other factors. Because threshold-based methods are difficult to cope with the complexity of multi-dimensional features, this work uses an unsupervised clustering-based method to learn the risk level of each category of traffic participants from the data. The clustering process can present explainable results

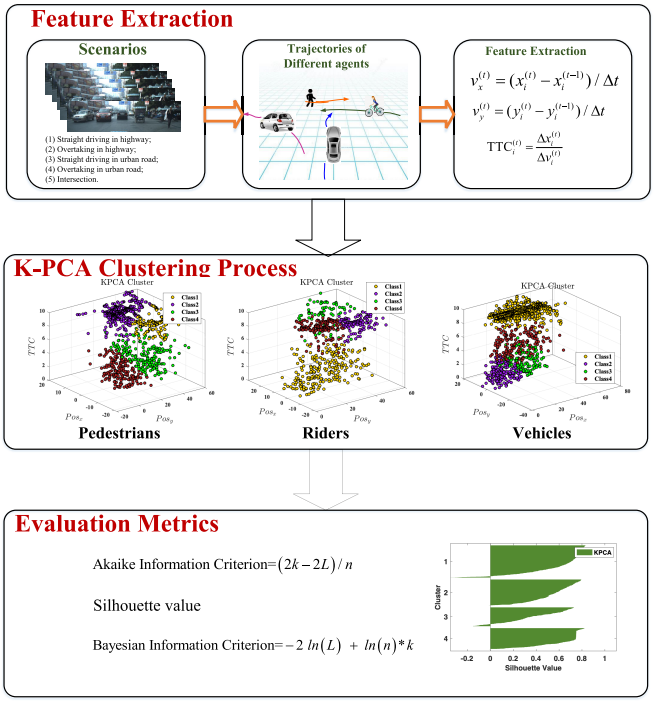


Fig. 2. The overall illustration of unsupervised risk-taking style generation.

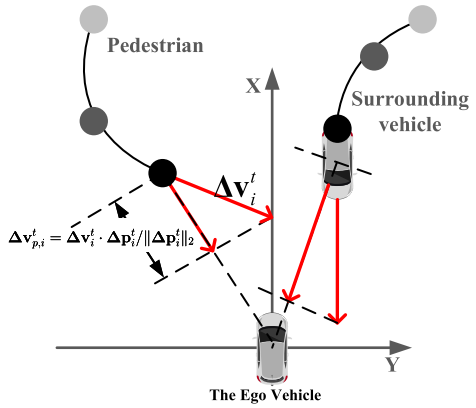


Fig. 3. The schematic diagram of TTC.

for the risk assessment, which includes the rank of risk-taking styles and the performance of the cluster. In this section, the detailed formulation of unsupervised RSG is presented in three parts: feature extraction, K-PCA clustering process and evaluation metrics. The relationship is shown in Fig.2.

A. Feature Extraction

In this step, three features are selected and extracted for RSG [27], [29], which are detailed as follows:

1) Relative Position: The relative position $\Delta \mathbf{p}_i^t$ of surrounding traffic participants is defined as the relative distance between i^{th} participant and the ego vehicle in longitudinal and lateral directions.

$$\Delta \mathbf{p}_i^t = (\Delta x_i^t, \Delta y_i^t) \tag{5}$$

2) Relative Velocity: The relative velocity can represent the tendency of relative movement, which can be applied to describe the risk in interactive scenarios by combining it with relative positions. For example, the scenario with a higher relative velocity and a lower relative distance can be empirically understood as a dangerous condition. In this research, we assume that the velocity for the ego vehicle and surrounding participants is stable and the acceleration is zero in a short time. The relative velocity $\Delta \mathbf{v}_i^t = (\Delta v_{x,i}^t, \Delta v_{y,i}^t)$ of t^{th} traffic participant at timestamp t can be calculated as follows:

$$\Delta v_{x,i}^t = (\Delta x_i^t - \Delta x_i^{t-1})/\Delta t \tag{6}$$

$$\Delta v_{y,i}^t = (\Delta y_i^t - \Delta y_i^{t-1})/\Delta t \tag{7}$$

3) Time to Collision (TTC): According to [29], [34], and [27], the value of TTC can be used to describe the risk potential field and rear-end collisions, which is formulated as:

$$\mathrm{TTC}_i^t = \frac{\sqrt{\Delta x_i^2 + \Delta y_i^2}}{\Delta \mathbf{v}_{p,i}^t} \tag{8}$$

where TTC_i^t is the value of TTC for i^{th} surrounding participants at time t. $\Delta \mathbf{v}_{p,i}^t = \Delta \mathbf{v}_i^t \cdot \Delta \mathbf{p}_i^t / \|\Delta \mathbf{p}_i^t\|_2$ is the projection of $\Delta \mathbf{v}_i^t$ in the direction of $\Delta \mathbf{p}_i^t$.When TTC_i^t is negative, the surrounding traffic participant is moving away from the ego vehicle and TTC_i^t will be set as 10 in statistical analysis. The schematic diagram of TTC is shown in Fig.3. In summary, for i participants, inputs $\mathbf{S}_i^{t_c-M+1:t_c}$ of unsupervised RSG are

$$\mathbf{S}_i^{t_c-M+1:t_c} = \{\mathbf{s}_i^{t_c-M+1}, \ldots, \mathbf{s}_i^{t_c}\} \tag{9}$$

where $\mathbf{s}_i^t$ is the feature vector of i^{th} traffic participant at each timestep t,

$$\mathbf{s}_i^t = [\Delta x_i^t, \Delta y_i^t, \Delta v_{x,i}^t, \Delta v_{y,i}^t, \mathrm{TTC}_i^t] \tag{10}$$

At each timestep t, given the $\mathbf{s}_i^t$ as input, the RSG module generates the corresponding cluster results for each timestep, which means the identified risk-taking styles at different timestep may be inconsistent. The final risk-taking style for surrounding participants sent to the module of trajectory prediction is static and represented by the majority of risk styles during this period.

$$r_{c_i} = \mathbf{mode}(r_{c_i}^t) \quad \forall t \in [t_c - M + 1, t_c] \tag{11}$$

where $\mathbf{mode}(\cdot)$ is the function that captures the value appearing most often in a set. $c_i \in \{vehicle, pedestrain, rider\}$ is the category of i^{th} traffic participant and $\mathbf{r}_{c_i} \in \{1, 2, \ldots, K_c\}$ is one of K_c risk-taking styles of i^{th} traffic participant. Similarly, $\mathbf{r}_{c_i}^t \in \{1, 2, \ldots, K_c\}$ is one of K_c risk-taking styles of i^{th} traffic participant at timestep t. In this process, the final risk-taking style r_{c_i} is selected. It will maintain a stable and static value of risk-taking styles. In the implementation, in a short period, We found that in most cases risk-taking styles are consistent and stable. By this way, the RSG module can provide a static value of risk-taking style in a short historical period.

B. Kernel Principal Component Analysis (KPCA)

The data process above is applied to extract useful features, which are used to describe the styles of participants. The goal of the unsupervised cluster is to divide participants into different categories based on the similarity of extracted features. The participants in the same cluster group indicate that they have the same style in perceiving the safety of interactive scenarios. The proposed framework for RSG is a general solution, which provides a formal flow of clustering, analysis and evaluation. This framework can be applied to all suitable methods. In order to demonstrate the basic principle and process, KPCA is selected and adopted in this research [27], [29], [35].

According to [36], the target of PCA is to realize the dimensional reduction for samples with high dimensional features. And the mean value of data with low-dimensional features is scaled and converted to zero. If samples are more scattered in a certain dimension, the data in this dimension is considered to have a higher performance in explanation, which is applied to achieve the purpose of retaining key features in representation and eliminating redundant information. Kernel PCA (KPCA) is developed based on PCA, which projects linearly inseparable data into a new high-dimensional space to make it suitable for alignment and linear classification. Kernel PCA can transform data $\mathbf{s}_i^t$ into a high-dimensional space through nonlinear mapping ϕ, use PCA in the high-dimensional space $\mathbf{s}_i^t$ to map it to another low-dimensional space $\hat{\phi}(\mathbf{s}_i^t)$, and finally divide samples by a linear classifier. Samples are transformed by KPCA and sent to K-means cluster (KMC) for clustering into K groups (denoted by $\mathbf{C} = \{\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_K\}$) by minimizing the within-cluster sum of squares. KMC is a general and efficient clustering method and its objective function can be illustrated as:

$$\arg\min \sum_{k=1}^{K} \sum_{\hat{\phi}(\mathbf{s}_{k,i}^t) \in \mathbf{C}_k} ||\hat{\phi}(\mathbf{s}_{k,i}^t) - \mu_k||^2 \tag{12}$$

where μ_k is the centroid of the k^{th} cluster.

C. Evaluation Metrics

A key parameter of unsupervised methods is K, which influences the performance and rationality of clustering results. In this research, three metrics are selected for evaluation: Akaike Information Criterion (AIC) metric, Bayesian Information Criterion (BIC) and silhouette value. AIC is developed based on information theory. It is applied to measure the performance of fitting results and evaluate the quality of models by punishing the increase of model parameters. It will avoid overfitting in the clustering process. The equation of AIC is formulated as the following:

$$AIC = \sum_{k=1}^{K} \sum_{\mathbf{s}_{k,i}^t \in \mathbf{C}_k} |\mathbf{s}_{k,i}^t - \mu_k|^2 + 2(K \cdot N) \tag{13}$$

where μ_k is the center of k^{th} cluster and $\mathbf{s}_{k,i}^t$ is the i^{th} sample in the k^{th} cluster. K and N are the number of clusters and dimensions, respectively. $\mathbf{C}_k$ is the k^{th} set of clustering category. Our goal is to select the minimum value of AIC. In general, when the complexity of the model increases (K increases), the AIC becomes smaller. However, if K is too large, the model is too complex to easily cause overfitting. AIC not only improves the fit of the model but also introduces a penalty term to make the model parameters as few as possible, which helps reduce the possibility of overfitting. It can be seen that the AIC criterion effectively and reasonably controls the dimension of parameters.

Obviously, the number of model parameters will grow with the increase of training samples. To reflect the penalty of this item, BIC is also applied for evaluation in the RSG. BIC is partly based on the likelihood function and is closely related to AIC. Similarly, the model with the lowest BIC is the best. The penalty item for the increase of training samples in BIC is greater than that in AIC. The formulation of BIC is as follows:

$$BIC = \sum_{k=1}^{K} \sum_{\mathbf{s}_{k,i}^t \in \mathbf{C}_k} |\mathbf{s}_{k,i}^t - \mu_k|^2 + \ln(M) \cdot (K \cdot N) \tag{14}$$

where M is the total number of training samples. By setting different values of clustering numbers K and comparing the variation of AIC and BIC, we can find the appropriate and explainable K.

In order to better analyze results with clear visualization, the silhouette value is applied. This method can be used to measure the distance of each point in the cluster relative to the cluster center. It can visualize the distribution of samples in each cluster. For sample i in clustering category I, the formulation of the silhouette value is written as follows:

$$a(i) = \frac{1}{|C_I| - 1} \sum_{j \in C_I, i \neq j} d(i, j) \tag{15}$$

$$b(i) = \min_{J \neq I} \frac{1}{|C_J|} \sum_{j \in C_J} d(i, j) \tag{16}$$

$$SV(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{17}$$

where $a(i)$ is the measure of the distance between a certain sample i and other samples in the category I. $|C_{I/J}|$ is the number of samples in category I or J. $b(i)$ is a measure of the smallest distance from a certain sample i to other categories $\{J | J \neq I, J \in \{1 : K\}\}$, and $d(i, j)$ is the Euclidean distance between samples i and j. SV is the normalized silhouette value distance, and all output results are between -1 and $+1$. According to the results, the larger the value of silhouette value, the better the performance of clustering.

IV. Risk-Taking Style-Oriented Graph Neural Network

In the previous section, the modeling process of unsupervised RSG is formulated, which is applied to generate levels of risk-taking styles. In this section, the heterogeneous graph is re-formulated as a risk-taking style-oriented GNN to predict future trajectories of traffic participants in interactive scenarios. The overall illustration of GNN is shown in Fig.4, which consists of four modules: graph structure in the spatial space with risk-taking styles, graph structure in the temporal space with the relationship in time series, the instance and categories for heterogeneous traffic participants, and the training and evaluation for trajectory prediction. Details of the instance layer, the category layer and the loss function are introduced in following parts.

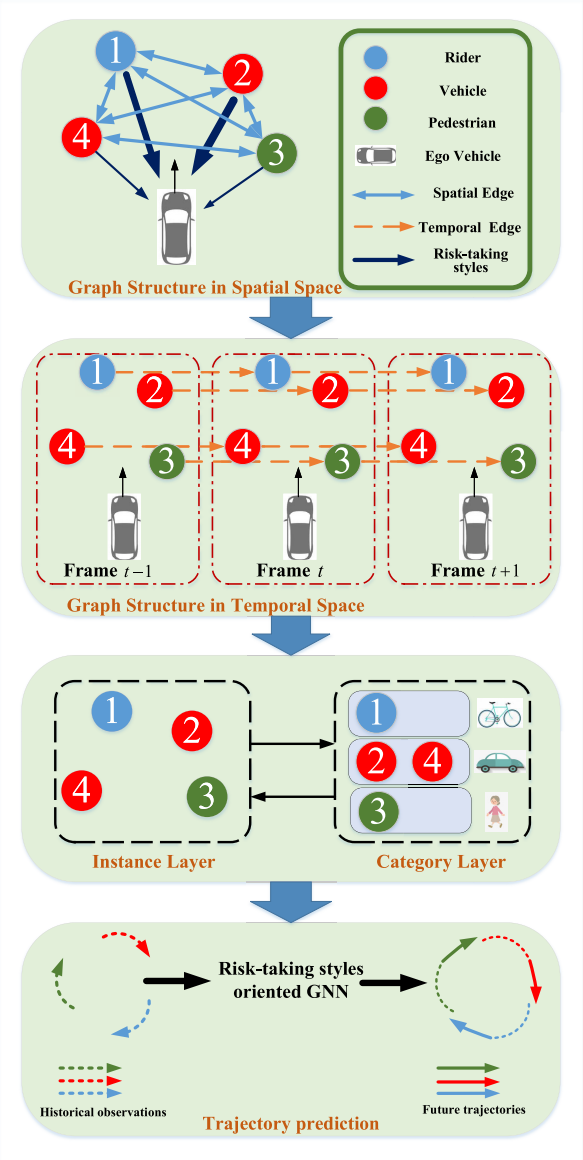


Fig. 4. The relationship and information passing between variables in GNN.

TABLE I
MAIN SYMBOLS OF INSTANCE AND CATEGORY LAYERS IN THE HETEROGENEOUS GNN

Parameters	Annotations
A_{instance}	Instance nodes
A_{Category}	Super nodes in the category layer
E_{Spatial}	Spatial edges
E_{Temporal}	Temporal edges
L_{ij}	The spatial edge LSTM
L_{ii}	The temporal edge LSTM
L_i	The instance LSTM
$\mathbf{h}_{ij}^t$	The hidden state in spatial edge LSTM
$\mathbf{h}_{ii}^t$	The hidden state in temporal edge LSTM
$\mathbf{h}1_i^t$	The first hidden state of the instance node LSTM
$\mathbf{h}2_i^{t-1}$	The final hidden state of the instance node LSTM
$\mathbf{h}1_m^t$	The hidden state in spatial edge LSTM
$\mathbf{h}2_m^t$	The final output of instance node
$\mathbf{h}_u^t$	The hidden state of category LSTM (super node)
$\mathbf{h}_{uu}^t$	The hidden state of temporal edge LSTM
$\mathbf{d}_m^t$	The movement feature for m^{th} instance node
$\mathbf{F}_u^t$	The feature of the corresponding super node
$\mathbf{F}_{uu}^t$	The feature of temporal edge in the category layer

A. Instance Layer

The goal of the instance layer is to capture attributes of instance nodes in the GNN, which represent characteristics of traffic participants. In the ST-GNN, LSTM can describe the relationship and influence of samples in time series, which is applied to model the interaction of the same participants. The nodes, spatial edges and temporal edges are all modeled by LSTMs in the temporal domain. We assume that participants with the same type have similar motion patterns. Therefore, three types of instance node LSTM (vehicles, riders and pedestrians) are built and trained in the ST-GNN. In each category, instance nodes with the same type share the uniform parameters of LSTM. Besides modeling instance nodes by LSTMs, the spatial and temporal edges in the graph structure are also required to be formulated in the temporal space, which is similar to the instance node LSTM. All spatial edges share the same parameters and all temporal edges are divided into three classes according to the type of participants. Definitions of primary variables in the instance and category layers are detailed in Table I.

At each timestep t, features $\mathbf{f}_{ij}^t$ of the directional spatial edge $E_{ij,\text{Spatial}}^t = (A_i^t, A_j^t)$ is embedded into a fix vector $\mathbf{z}_{ij}^t$,

$$\mathbf{z}_{ij}^t = \Omega(\mathbf{f}_{ij}^t; \mathbf{W}_{\text{spa}}^{\text{e}}) \tag{18}$$

where $\mathbf{W}_{\text{spa}}^{\text{e}}$ are weights of the embedding function $\Omega(\cdot,\cdot)$. The feature vector $\mathbf{z}_{ij}^t$ is fed into spatial edge LSTM L_{ij} to generate hidden states $\mathbf{h}_{ij}^t$, which is formulated as follow:

$$\mathbf{h}_{ij}^t = LSTM(\mathbf{h}_{ij}^{t-1}, \mathbf{z}_{ij}^t; \mathbf{W}_{\text{spa}}^{\text{L}}) \tag{19}$$

where $\mathbf{W}_{\text{spa}}^{\text{L}}$ are weights of spatial edge LSTM L_{ij}. Similarly, according to the definition of spatial edges, the LSTM L_{ii} of temporal edges $E_{ii,\text{Temporal}}^t = (A_i^t, A_i^{t+1})$ can be modeled in the same way and the hidden state $\mathbf{h}_{ii}^t$ is applied to describe the relationship in time series.

In complicated and dynamic urban scenarios, for a specific traffic participant, surrounding vehicles have different influences on their behaviors, which is reflected in the difference of weights in spatial and temporal edges. According to [6], the soft attention mechanism is applied to generate weights w for edges between instance nodes.

$$w(\mathbf{h}_{ij}^t) = \text{softmax}(\frac{k}{\sqrt{d_e}} Dot(\mathbf{W}_{ii}\mathbf{h}_{ii}^t, \mathbf{W}_{ij}\mathbf{h}_{ij}^t)) \tag{20}$$

where $\mathbf{W}_{ii}$ and $\mathbf{W}_{ij}$ represent embedding weights from temporal and spatial perspectives, respectively. The scaling factor is given by $k/\sqrt{d_e}$ and $Dot(\cdot,\cdot)$ donates the dot product. From the view of spatial interactions, the influence of different instance nodes can be calculated by the weighted sum of $\mathbf{h}_{ij}^t$:

$$\mathbf{z}_i^t = \Omega(\mathbf{f}_i^t; \mathbf{W}_{\text{instance}}^{\text{node}}) \tag{21}$$

$$\mathbf{a}_i^t = \Omega(concat(\mathbf{h}_{ii}^t, \mathbf{H}_i^t); \mathbf{W}_{\text{instance}}^{\text{edge}}) \tag{22}$$

$$\mathbf{h}1_i^t = LSTM(\mathbf{h}2_i^{t-1}, concat(\mathbf{z}_i^t, \mathbf{a}_i^t); \mathbf{W}_{\text{instance}}^{\text{L}}) \tag{23}$$

where $\mathbf{h}1_i^t$ and $\mathbf{h}2_i^{t-1}$ are first and final hidden states in L_i, respectively. The sum of $\mathbf{h}_{ij}^t$ is given by $\mathbf{H}_i^t$, which represents

the effect of spatial interactions. Hidden states of spatial interactions $\mathbf{H}_i^t$ and temporal interactions $\mathbf{h}_{ii}^t$ are concatenated and embedded into a fixed vector $\mathbf{a}_i^t$, which is concatenated with instance node feature and sent to the instance LSTM L_i. In (21)-(23), $\mathbf{W}_{\text{instance}}^{\text{node}}$ and $\mathbf{W}_{\text{instance}}^{\text{edge}}$ are embedding weights, and $\mathbf{W}_{\text{instance}}^{\text{L}}$ is the weight of i^{th} instance node LSTM cell.

B. Category Layer

Traffic participants with the same categories have similar dynamic properties (velocity, acceleration, etc.) and reaction time to other participants. To efficiently learn movement patterns from the same category of instances, we can better model and understand the interactive behaviors of heterogeneous participants. Therefore, the category layer is applied in the HGM. Similar to the structure of the instance layer, three super nodes (rider, pedestrian and vehicle) are defined in the category layer. Meanwhile, according to the architecture design in [20], three super nodes LSTMs are built to model the temporal interaction for rider, pedestrian and vehicle, respectively. The whole category layer consists of four parts: super nodes, temporal edges, edges from instance nodes to the super node and edges from the super node to instance nodes.

In the category layer, the feature $\mathbf{F}_u^t$ can be seen as the averaging of the movement feature $\mathbf{d}_m^t$ of instance nodes, which can be formulated as follow:

$$\mathbf{F}_u^t = \frac{1}{n}\Sigma_{m=1}^{n}\mathbf{d}_m^t \tag{24}$$

where $\mathbf{d}_m^t$ is the movement feature for m^{th} instance node in the u^{th} category. For example, the feature $\mathbf{F}_{\text{pedestrian}}$ of the pedestrian category node is the sum of features of the pedestrian instance node. At each timestep t, $\mathbf{d}_m^t$ can be calculated by hidden states $\mathbf{h}1_i^t$ and cell states $\mathbf{c}_i^t$ of instance nodes:

$$\mathbf{d}_m^t = \mathbf{h}1_i^t \otimes \text{softmax}(\mathbf{c}_i^t) \tag{25}$$

The equations above indicate that the calculation process of super node features considers the information ($\mathbf{h}1_i^t$ and $\mathbf{c}_i^t$) transferred from instance node in the corresponding category, which is represented by directional edges from instance nodes to the super node. Meanwhile, similar to edges in the instance layer, the temporal edge $\mathbf{F}_{uu}^t = \mathbf{F}_u^t - \mathbf{F}_u^{t-1}$ between super nodes is combined with the hidden state $\mathbf{h}_{uu}^t$:

$$\mathbf{z}_{uu}^t = \Omega(\mathbf{F}_{uu}^t; \mathbf{W}_{\text{st}}^{\text{e}}) \tag{26}$$

$$\mathbf{h}_{uu}^t = LSTM(\mathbf{h}_{uu}^{t-1}, \mathbf{z}_{uu}^t; \mathbf{W}_{\text{st}}^{\text{L}}) \tag{27}$$

where $\mathbf{W}_{\text{st}}^{\text{e}}$ and $\mathbf{W}_{\text{st}}^{\text{L}}$ are weights of embedding layer and temporal LSTM cells, respectively. Features of instance group $\mathbf{F}_u^t$ and $\mathbf{h}_{uu}^t$ are combined with features from hidden states $\mathbf{h}_u^t$ in the category layer to model the super node LSTM:

$$\mathbf{z}_u^t = \Omega(\mathbf{F}_u^t; \mathbf{W}_{\text{u}}^{\text{node}}) \tag{28}$$

$$\mathbf{h}_u^t = LSTM(\mathbf{h}_u^{t-1}, concat(\mathbf{z}_u^t, \mathbf{h}_{uu}^t); \mathbf{W}_{\text{u}}^{\text{L}}) \tag{29}$$

where $\mathbf{W}_{\text{u}}^{\text{e}}$ and $\mathbf{W}_{\text{u}}^{\text{L}}$ are embedding weights and super node LSTM cells, respectively. Finally, the hidden state $\mathbf{h}_u^t$ is concatenated with $\mathbf{h}1_m^t$ and sent back to the instance node to generate the final output $\mathbf{h}2_m^t$ of m^{th} instance node:

$$\mathbf{h}2_m^t = \Omega(concat(\mathbf{h}1_m^t, \mathbf{h}_u^t); \mathbf{W}_m^{\text{e}}) \tag{30}$$

where $\mathbf{W}_{\text{i2}}^{\text{e}}$ are weights in the embedding function.

C. Dynamically-Feasible Trajectory Generation

In the training process, if the driving dataset can provide the position in a fixed coordinate system (e.g., Waymo dataset.), the dynamic characteristic of participants can be considered in the neural network model. In this setting, we do not need to generate the relative coordinates of the predicted participant relative to the ego car, but directly obtain the position in the fixed coordinate system. Specifically, in the problem formulation of this part, x and y are fixed coordinates, whose origin does not change with the movement of the ego vehicle. To capture the dynamical constraints of heterogeneous traffic participants, the physical model is injected into the data-driven neural network. Specifically, at timestep t, each LSTM module generates the parameters of bivariate Gaussian distribution $(\mu_{\mathbf{u}}^t, \sigma_{\mathbf{u}}^t, \rho_{\mathbf{u}}^t)$ corresponding to the control action $\mathbf{u}^t$ (e.g. yaw rate and longitudinal acceleration), where $\mu_{\mathbf{u}}^t$, $\sigma_{\mathbf{u}}^t$ and $\rho_{\mathbf{u}}^t$ are mean value, standard deviation, and correlation coefficient at timestep t, respectively.

For pedestrians, the single integrator is applied with control action $\mathbf{u}^t = \dot{\mathbf{p}}^t$, where $\mathbf{p}$ is the position vector $[x, y]^T$ and $\mathbf{u}$ is the velocity vector $\dot{\mathbf{p}} = [\dot{x}, \dot{y}]^T$. The position $\mu_{\mathbf{p}}^{t+1}$ at $t+1$ is calculated based on the $\mu_{\mathbf{p}}^t$ and $\mu_{\mathbf{u}}^t$:

$$\mu_{\mathbf{p}}^{t+1} = \mu_{\mathbf{p}}^t + \mu_{\mathbf{u}}^t \Delta t \tag{31}$$

The propagation of uncertainty from control action to state is detailed in Appendix A. For vehicles and riders, the dynamically-extended unicycle model is adopted to capture the dynamical constraints. Usually, the unicycle model takes velocity v and yaw rate ω as inputs. However, considering that the vehicle is controlled by the throttle pedal, the dynamically-extended unicycle model is applied using the longitudinal acceleration a and yaw rate ω. The nonlinear continuous-time dynamics can be expressed as:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} v\cos(\phi) \\ v\sin(\phi) \\ \omega \\ a \end{bmatrix} \tag{32}$$

where ϕ is the yaw angle. The uncertainty of the dynamically-extended unicycle model can be approximated by linearizing the dynamics. The detailed derivation is presented in Appendix B.

D. Loss Function

In the training process of interactive behavior prediction, at each timestep t, predicted results of future trajectories are assumed to conform to the bivariate Gaussian distribution and can be described as

$$(\Delta x_i^t, \Delta y_i^t) \sim [\mu_i^t, \sigma_i^t, \rho_i^t] \tag{33}$$

where $\mu_i^t = (\mu_x, \mu_y)_i^t$ is the mean value, $\sigma_i^t = (\sigma_x, \sigma_y)_i^t$ is the standard deviation and ρ_i^t is the correlation coefficient. Similarly, the trajectories considering dynamical constraints can be expressed as:

$$(x_i^t, y_i^t) \sim [\mu_{\mathbf{u},i}^t, \sigma_{\mathbf{u},i}^t, \rho_{\mathbf{u},i}^t] \tag{34}$$

where $\mathbf{u}$, σ and ρ are the mean value, standard deviation, and correlation coefficient of control actions, respectively. The hidden layer of the instance node LSTM is applied to generate the parameters above by the linear function$\Omega(\cdot, \cdot)$:

$$[\mu_i^t, \sigma_i^t, \rho_i^t] = \Omega(\mathbf{h}_{i2}^{t-1}, \mathbf{W}_f) \tag{35}$$

$$[\mu_{\mathbf{u},i}^t, \sigma_{\mathbf{u},i}^t, \rho_{\mathbf{u},i}^t] = \Omega(\mathbf{h}_{i2}^{t-1}, \mathbf{W}_f) \tag{36}$$

Similar to [31], the final form of the loss function in the training process is described by the negative log Likelihood:

$$L(\mathbf{W}_{\text{GNN}}) = -\Sigma_{t=T_{\text{obs}}+1}^{T_{\text{pred}}} \log(P(\Delta x_i^t, \Delta y_i^t \,|\mu_i^t\,, \sigma_i^t, \rho_i^t)) \tag{37}$$

where $\mathbf{W}_{\text{GNN}} = (\mathbf{W}_{\text{spa}}, \mathbf{W}_{\text{tem}}, \mathbf{W}_{\text{ins}}, \mathbf{W}_{\text{st}}, \mathbf{W}_{\text{sup}}, \mathbf{W}_m, \mathbf{W}_{\text{f}})$ are weighted parameters. The objective of the training process is to minimize the loss function (37) by back-propagating the predicted error through parameters $\mathbf{W}_{\text{GNN}}$ in instance and category layers.

V. Experiments for Unsupervised Risk-Taking Style Generation

In Section III and Section IV, the formulation of unsupervised RSG and supervised HGM are detailed, respectively, which are two sequential steps in the proposed method. In this section, experiments for the first step are presented for riders, pedestrians, and vehicles. The dataset used in this research is first introduced. Then, the results for the unsupervised cluster method are presented. Finally, the analysis for model selection is detailed.

A. Dataset and Experimental Settings

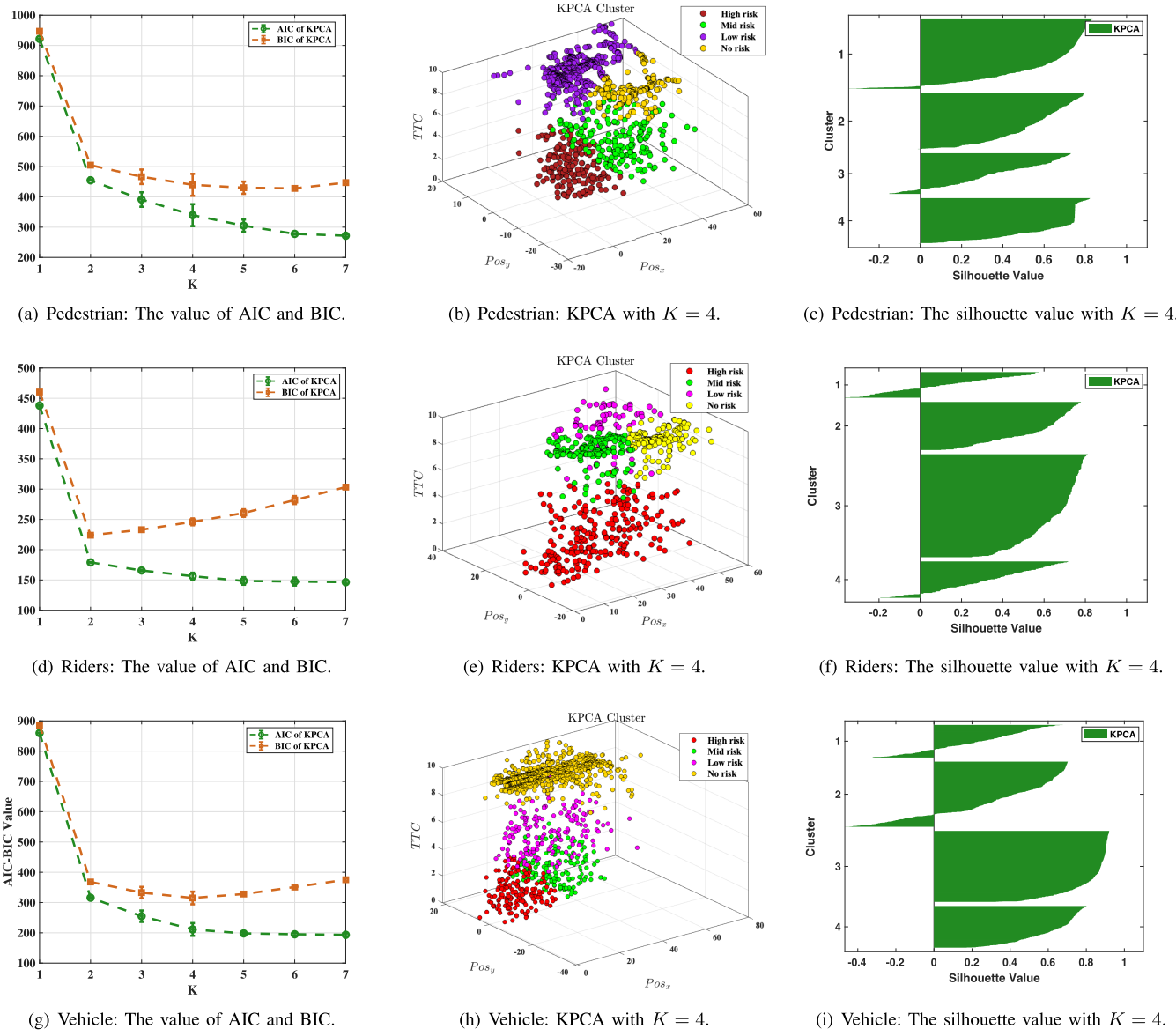
1) Dataset: This research focuses on interactive behavior modeling in urban scenarios. The dataset for training and verification needs to cover complex influence and interaction between heterogeneous participants. Meanwhile, well-labeled positions and classes for traffic participants are required in the training process of the proposed risk-taking style-oriented GNN. Therefore, the datasets from the highway scenario (HighD and NGSIM) cannot meeting the requirement [37], [38]. In this study, we used two datasets in experiments, namely, the BLVD and Waymo Open Motion Dataset (WOMD). The BLVD dataset was collected from urban roads in China, while the WOMD was gathered from multiple locations across the United States, including San Francisco, Mountain View, Los Angeles, Detroit, Seattle, and Phoenix. The details of the two datasets are shown as follows.

- **BLVD dataset** is first released in [39], which is mainly used to model interactions between heterogeneous traffic participants in urban scenarios. Different kinds of specific interactive events between ego-vehicle and surrounding participants are semantically labeled, which are detailed in [39]. Different from other public datasets, BLVD provides a dynamic 5D semantic benchmark (3D+temporal+interactive), including 654 calibrated video clips for three kinds of participants: vehicles, pedestrians and riders (cyclists and motorbikes). Four types of scene conditions are collected in BLVD: daytime & low densities, nighttime & low densities, daytime & high densities, and nighttime & high densities. Moreover, 13, 8 and 7 interactive events between the ego vehicle and traffic participants are labeled in BLVD for vehicles, pedestrians and riders, respectively. The diversity and coverage of the BLVD dataset make it more suitable for interactive behavior modeling in this research. Some trajectories are too short and unsuitable for training the prediction model. Therefore, 187, 1,039 and 2,222 tracks are finally selected for pedestrians, riders and vehicles, respectively.
- **WOMD** is a large-scale naturalistic driving dataset encompassing 103,354 unique driving scenarios, each of which spans a duration of 20 seconds. A data selection strategy is conducted inspired by [40], where we randomly selected 20% of the entire dataset (20,600 scenarios) for model training. Considering that the driving scenarios are randomly stored within the dataset, the selected 20% scenarios maintain a similar distribution to the original dataset. A sliding time window of 7 seconds is set to segment each 20 seconds scenario into input and output features for training. Within this window, the initial 2 seconds were utilized as model input, while the remaining 5 seconds served as model output. The time window was shifted to 1 second intervals. Finally, 80% of processed samples were used for training and the remaining 20% for testing.

2) Experimental Settings for Unsupervised Clustering: In order to obtain risk-taking styles based on unsupervised clustering methods, the first step is to select an appropriate value of clustering numbers. In the experiment, a set of cluster numbers $K = \{1, 2, 3, 4, 5, 6, 7\}$ are applied to evaluate the combined influence of AIC, BIC and silhouette value. In the KPCA, the number of components in the high dimensional space is 20. The selected kernel is a radial basis function (RBF) with a variance of 0.7.

B. Experimental Results and Analysis

Fig. 5(a), Fig. 5(d) and Fig. 5(g) present the variation of AIC and BIC with different cluster number K for pedestrians, riders and vehicles, respectively. With the increase of cluster number K, the value of AIC and BIC for KPCA shows a downward trend, which indicates that as the value of K increases, the ability of the unsupervised clustering model to explain the clustering process of data samples gradually increases. According to [27], the selection of clustering number K needs to trade off actual meanings, cluster metrics and downstream task performance. In the initial experiments, cluster numbers 2, 3, 4, and 5 were chosen for the integration with the trajectory prediction task. The ADEs for each choice are 0.17m, 0.13m, 0.093m and 0.10m. The results of $K = 3$ and $K = 4$ are close. Then, $K = 4$ also encapsulates the physical meanings of risk-taking styles. Therefore, Taking the above factors into



(a) Pedestrian: The value of AIC and BIC. (b) Pedestrian: KPCA with $K = 4$. (c) Pedestrian: The silhouette value with $K = 4$.

(d) Riders: The value of AIC and BIC. (e) Riders: KPCA with $K = 4$. (f) Riders: The silhouette value with $K = 4$.

(g) Vehicle: The value of AIC and BIC. (h) Vehicle: KPCA with $K = 4$. (i) Vehicle: The silhouette value with $K = 4$.

Fig. 5. Clustering results of pedestrians, riders and vehicles in BLVD dataset.

consideration, $K = 4$ is finally selected as the clustering number.

Fig. 5(c), Fig. 5(f) and Fig. 5(i) shows the silhouette value of KPCA with $K = 4$. In these figures, the vertical axis represents four clustering categories, and the horizontal axis represents the normalized silhouette value. In each category, the closer the silhouette value of all data points is to 1, the better the clustering effect.

For pedestrians, as the number of clusters increases from 1 to 7, the AIC of KPCA decreases from 921 to 271, and the BIC decreases from 946 to 447. According to [27], Coordinately taking the interpretability and the variation of AIC and BIC into consideration, $K = 4$ is finally chosen as the number of clusters. With $K = 4$, scatter plots of pedestrian samples in the dimension of x, y, and TTC are shown in Fig. 5(b). Meanwhile, Table II specifically shows the corresponding average values of several features in each category. Besides the relative position and relative velocity, the average Euclidean distance $\overline{D} = \frac{1}{N_c}\sum_{n=1}^{N_c}\sqrt{\Delta x_n^2 + \Delta y_n^2}$ and average relative velocity $\overline{Vel} == \frac{1}{N_c}\sum_{n=1}^{N_c}\sqrt{\Delta Vel_{x,n}^2 + \Delta Vel_{y,n}^2}$ are also calculated for analysis, where N_c is the number of samples in the corresponding cluster. It should be noted that the calculation of the average relative speed takes into account the influence of the speed direction. First, the synthesized relative velocity vector is calculated, and then the module of the vector is calculated to represent the value of the relative velocity. Therefore, the final average relative velocity may have negative values. When the average relative speed is negative, it means that there is a tendency for the ego vehicle to be relatively close to the surrounding traffic participants. On the contrary, it means that there is a tendency for the ego vehicle to move away from surrounding traffic participants. As shown in Table II, overall, the four risk-taking styles are associated

TABLE II
COMPARATIVE RESULTS OF MEAN VALUES FOR CLUSTERS IN BLVD DATASET

	Style	$\overline{\Delta x}$	$\overline{\Delta y}$	$\overline{\Delta v_x}$	$\overline{\Delta v_y}$	$\overline{TTC}$	$\overline{D}$	$\overline{Vel}$
Pedestrian	High risk	15.7	0.34	-9.39	0.10	1.90	16.1	-10.1
	Mid risk	30.3	-4.83	-8.8	0.13	4.28	33.4	-9.4
	Low risk	22.8	5.0	-0.29	-0.09	9.86	28.5	-0.79
	No risk	32.5	-10.17	-0.65	0.07	9.66	57.3	0.32
Rider	High risk	24.9	0.75	-11.0	0.34	2.75	27.1	-12.4
	Mid risk	19.1	-5.08	-0.86	-0.66	9.63	28.3	-2.31
	Low risk	35.2	12.77	-1.85	1.22	9.34	47.9	-0.94
	No risk	38.3	-9.36	0.91	-0.05	9.70	53.2	1.07
Vehicle	High risk	13.51	1.22	-2.17	-0.16	2.32	15.4	-2.87
	Mid risk	33.72	3.14	-2.90	-0.01	2.79	37.8	-3.94
	Low risk	27.35	1.11	-3.31	-0.05	6.12	30.4	-3.67
	No risk	25.85	1.28	-2.85	0.03	9.33	29.4	-2.92

with average relative distance and average relative speed. Specifically, smaller average distances and larger approaching speeds increase risk. Larger relative distances between ego and surrounding vehicles and smaller approaching speeds reduce risk. This is consistent with driver intuition. Combining the scatter Fig.5(b) and pedestrian part in Table II, four risk-taking styles corresponding to pedestrians are determined: high risk (red), middle risk (green), low risk (purple) and no risk (yellow). According to the silhouette value in Fig. 5(c), most of the samples for KPCA concentrates between 0 and 1, which shows that with $K = 4$, the clustering performance of KPCA is acceptable.

Fig.5(d) presents the variation of AIC and BIC for riders in the clustering process. The AIC value of KPCA decreases from 437 to 146 with the increase of clustering number K while the value of BIC reaches the lowest at $K = 2$. Similar to the analysis of pedestrians, taking into account the interpretability of risk-taking styles obtained by clustering, $K = 4$ is finally selected as the parameter of RSG. Fig.5(f) shows the result of SV. Most of the samples in KPCA concentrate between 0 and 1 which indicates that the unsupervised risk level clustering of KPCA obtains a good performance at $K = 4$.

The above analysis discusses the unsupervised RSG of pedestrians and riders, who usually take potentially high risks due to the lack of road constraints and substantial uncertainty in the movement. In urban scenarios, although vehicles are restricted by strong road boundaries and traffic rules, as the most important traffic participants, vehicles also need to be analyzed from the perspective of risk-taking styles, which are shown in Fig.5(g-i). Similar to the trend of pedestrians, as the number of clusters increases, the AIC of KPCA decreases from 859 to 193. The change of BIC value presents a trend of firstly decreasing and then slowly increasing, with the lowest point at $K = 4$. Taking into consideration the fact that the number of clusters cannot be too many, which can not correspond to the actual meaning, and the AIC and BIC should be as small as possible, the final selection of K is 4.

After analyzing the risk-taking styles of heterogeneous traffic participants in the BLVD dataset, we expanded our study to the analysis of WOMD in our experiment. The results are presented in Table III. Similar analysis methods were employed on the WOMD as applied to the BLVD dataset. For each type of traffic participant (pedestrians, riders, and vehicles), risk-taking styles were categorized into four types using unsupervised clustering. In experiments involving trajectory prediction, style labels generated from the WOMD will be integrated into the HGL based on the graph neural network.

In this research, the goal of unsupervised RSG is to cluster surrounding traffic participants into different levels. The proposed method in this research is a generic framework, which is adaptable to all similar unsupervised clustering methods. Based on the analysis in this section, $K = 4$ in KPCA is selected to be formulated in the GNN to construct the risk-taking style-oriented GNN. Detailed results of risk-taking-oriented GNN will be illustrated in the next section.

VI. EXPERIMENTS FOR TRAJECTORY PREDICTION

Based on the experimental analysis in Section V, the risk-taking style for vehicles, riders and pedestrians are generated, respectively. In this section, the results of risk-taking style-oriented trajectory prediction are presented and detailed. Meanwhile, baseline methods and evaluation metrics are introduced for comparative study with two datasets.

A. Baseline Methods

Four related methods are selected as baselines, which include one general model (LSTM) and three social-aware approaches:

1) LSTM: LSTM is a general and basic method to model time series problems, which is applied for trajectory prediction in [41].

2) Social LSTM: SL is firstly proposed in [5], which models specific interactions between pedestrians by social pooling layer and predicts the trajectory by LSTM.

3) Social Attention: SA proposes to model complicated spatial interactions as a graph structure and features of traffic participants are embedded as node attributes. LSTMs are applied on nodes and edges to model temporal interactions [33].

4) TrafficPredict GNN: TrafficPredict GNN is developed based on SA, which considers the influence of heterogeneous traffic participants in the category layer [20]. The proposed

TABLE III
COMPARATIVE RESULTS OF MEAN VALUES FOR CLUSTERS IN WOMD

	Style	$\overline{\Delta x}$	$\overline{\Delta y}$	$\overline{\Delta v_x}$	$\overline{\Delta v_y}$	$\overline{TTC}$	$\overline{D}$	$\overline{Vel}$
Pedestrian	High risk	17.3	2.7	-5.8	0.7	3.21	21.2	-6.1
	Mid risk	23.4	3.3	-3.3	0.2	4.45	25.1	-3.8
	Low risk	27.8	-0.8	0.9	0.1	7.37	29.6	1.3
	No risk	31.3	-2.9	7.8	-0.3	7.92	33.7	9.1
Rider	High risk	20.1	-0.7	-3.4	0.3	1.97	21.2	-3.8
	Mid risk	18.3	1.3	-2.9	0.5	3.68	21.7	-3.1
	Low risk	26.7	2.8	0.8	1.3	6.45	30.4	1.9
	No risk	35.4	3.3	0.4	1.1	9.17	38.2	1.4
Vehicle	High risk	15.2	0.4	-2.8	0.5	2.91	16.3	-3.1
	Mid risk	27.8	-1.3	-1.9	0.7	3.69	29.3	-2.2
	Low risk	31.3	0.7	-0.4	-1.3	8.49	34.4	-1.4
	No risk	26.5	1.5	1.3	0.9	9.53	28.9	1.8

risk-taking-oriented GNN is formulated based on TrafficPredict GNN. Therefore, the difference between both methods can reflect the positive effect of the proposed method.

5) Trajectron++: *Trajectron++* is developed in [2], which integrates the Conditional Variational Autoencoder (CVAE) and social-aware multi-agent trajectory prediction. It encodes agent interactions by the graph-based spatiotemporal neural networks and incorporates heterogeneous data with Convolutional Neural Network (CNN). However, the characteristic of different kinds of traffic participants is not considered in this work. In the interests of fairness in our comparative experiments with the method proposed in this paper, the impact of map information and the influence of the ego vehicle's motion planning on the interaction behavior of surrounding vehicles are blocked. All the basic parameter settings refer to [2].

6) RSG+Trajectron++: To validate the positive effect and the feasibility of the RSG module, *Trajectron++* is combined with RSG with the aggregation of the node feature tensor.

B. Experimental Settings and Evaluation Metrics

In this research, the GNN model is established based on Pytorch[1] platform for experiments. The PC configuration is Intel Core i5-6300HQ at 2.3GHz, 8GB RAM and 960M GPU. In the experiment, for all 3440 trajectories, we used cross-validation to select 90% (3,096 samples) as the training set, and the remaining 10% (344 samples) as the testing set. The GNN parameters used in the experiment are as shown in Table IV. In experiments with BLVD dataset, we found that most of the sequences are less than 5 seconds, which need to be divided as the input observation and ground truth in the training process. According to [31] and [32], the length of input observation is set as 10 frames (1.0s), 20 frames (2.0s) and 30 frames (3.0s). And the length of the output prediction is set as 5 frames (0.5s), 10 frames (1.0s), 15 frames (1.5s) and 20 frames (2.0s). In experiments with WOMD, the length of input observation is 20 frames (2.0s) and the output prediction is 50 frames (5.0s). To validate the performance of injecting dynamical constraints into the overall framework, two kinds of proposed methods are presented (Table VII). The first is the same as the model applied in the BLVD dataset. The second is *dynamic+proposed*, which integrates the physical model.

[1]https://pytorch.org/

TABLE IV
MAIN PARAMETERS IN EXPERIMENTS

Parameters	Values
Temporal edge cell	128
Spatial edge cell	128
Node cell	64
Embedding layer	64
Learning rate	0.001
Epoch	20

The comparison between these two models can highlight the influence of physical constraints.

According to [31] and [38], two metrics are selected to evaluate the performance of trajectory prediction:

1) Average Distance Errors (ADE): ADE represents the average Euclidean distance for the whole predicted trajectory, which is formulated as:

$$ADE = \frac{\Sigma_{i=1}^{M} \Sigma_{t=t_{\text{obs}}+1}^{t_{\text{obs}}+t_{\text{pred}}} \sqrt{(\Delta x_i^t - \Delta \hat{x}_i^t)^2 + (\Delta y_i^t - \Delta \hat{y}_i^t)^2}}{M * t_{\text{pred}}} \tag{38}$$

2) Final Distance Errors (FDE):: FDE represents the Euclidean distance for the final predicted position, which is formulated as:

$$FDE = \frac{\Sigma_{i=1}^{M} \sqrt{(\Delta x_i^{t_{\text{pred}}} - \Delta \hat{x}_i^{t_{\text{pred}}})^2 + (\Delta y_i^{t_{\text{pred}}} - \Delta \hat{y}_i^{t_{\text{pred}}})^2}}{M} \tag{39}$$

where M is the total number of traffic participants. $(\Delta \hat{x}_i^{t_{\text{pred}}}, \Delta \hat{y}_i^{t_{\text{pred}}})$ and $(\Delta x_i^t, \Delta y_i^t)$ are predicted locations and ground truth for i^{th} traffic participant at time t, respectively. The above calculation of ADE and FDE is formulated based on $(\Delta x, \Delta y)$ in the relative coordinate system. It fits the experiments in BLVD dataset. In the fixed coordinate system, we need to substitute $(\Delta x, \Delta y)$ with (x, y), which suits the WOMD.

C. Experimental Results and Analysis in BLVD Dataset

Fig. 6 shows the change of the loss function for trajectory prediction as the number of epochs increases during the

TABLE V
COMPARATIVE RESULTS FOR PREDICTION OF INTERACTIVE TRAJECTORIES IN BLVD DATASET(ADE/FDE)

Metrics	Agents	LSTM[41]	SL[5]	SA[33]	*TrafficPredict*[20]	*Trajectron++*[2]	*RSG+Trajectron++*	Proposed
ADE[m]	Pedestrian	0.27	0.14	0.13	0.12	0.133	0.128	**0.09**
	Rider	0.38	0.14	0.13	0.14	0.145	0.133	**0.10**
	Vehicle	0.38	0.14	0.12	0.09	0.144	0.108	**0.07**
	Average	0.347	0.14	0.127	0.117	0.131	0.123	**0.087**
FDE[m]	Pedestrian	0.39	0.25	0.24	0.21	0.193	0.176	**0.12**
	Rider	0.57	0.22	0.21	0.2	0.177	0.152	**0.13**
	Vehicle	0.56	0.18	0.17	0.18	0.158	0.134	**0.11**
	Average	0.507	0.217	0.207	0.197	0.176	0.154	**0.12**

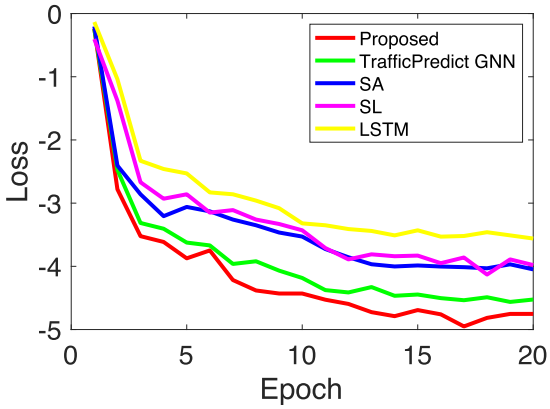


Fig. 6. The variation of loss in the training process (20 epochs).

TABLE VI
COMPARATIVE RESULTS FOR COMPUTATION TIME IN BLVD DATASET

Methods	Frames/per second
LSTM	27.2
SL	16.2
SA	17.3
TrafficPredict	12.6
Trajectron++	14.6
RSG+Trajectron++	15.6
Proposed	10.5

training process. Horizons of observations and prediction are 3.0s (30 frames) and 1.0s (10 frames), respectively. The loss of all compared methods (except for *Trajectron++*) declines, and the proposed method obtains the best performance and reaches the lowest loss after 20 epochs. It indicates that risk-taking style-oriented GNN can reduce errors in trajectory prediction and improve the efficiency of the training process.

For the model training process in Fig 6, Table V shows the comparative results of several methods with two evaluation metrics: ADE and FDE. In Table V, the length of input and output are both 20 frames (2s). Compared to LSTM, six social-based methods have achieved better results (SL, SA, *TrafficPredict*, *Trajectron++*, *RSG+Trajectron++* and Proposed). The ADE and FDE range of LSTM are 0.27m∼0.38m and 0.39m∼0.56m, respectively. As a comparison, the range of ADE and FDE for all social methods are 0.07m∼0.14m and 0.11m∼0.25m, respectively. Comparing the results of six social methods, SL obtains the largest error with 0.14m and 0.217m in ADE and FDE, respectively. By taking the attention mechanism into consideration, the performance of SA is better than SL in ADE and FDE, which indicates that weighted edges in the graph structure can benefit the modeling process of interactive behaviors. Based on SA, TrafficPredict formulates heterogeneous traffic participants and improves the performance by precisely modeling the motion pattern for each type of participant. The injection of RSG in *Trajectron++* improves the performance (6.1% in ADE and 12.5% in FDE), which indicates the positive effect in the interactive behavior prediction. For the proposed risk-taking style-oriented GNN in this research, focusing on the unsupervised RSG further reduces the predicted error, which is the core contribution of the proposed method.

In the integrated system of intelligent vehicles, the goal of trajectory prediction is to provide accurate solution space for motion planning. The high-precision prediction brought about by high computational time consumption cannot provide real-time adequate solution space for motion planning. Therefore, for the trajectory prediction of interactive traffic participants in the complex dynamic environment, the time consumption and prediction efficiency of the algorithm are important factors in evaluating the performance of the algorithm. Therefore, besides the comparison in the accuracy of trajectory prediction, this research also gives a comparison of the time consumption. The evaluation metric is the number of frames predicted per second (frames/per second). For the proposed method, the time cost to generate risk-taking styles by clustering is included. Table VI presents the result of computation time in prediction for all five methods. The larger the value, the better performance in computational efficiency in the real-time application. The computation time of the proposed method is slightly lower than that of other social-based methods, but it is within the same order of magnitude. The reason for the increase in time consumption is the integration of the unsupervised risk level clustering model in the overall framework.

In Fig. 6, Table. V and Table. VI, the length of input observation is 3.0s (30 frames) and the horizon of trajectory prediction is 1.0s (10 frames). To further investigate the performance of the proposed method with different observed and predicted horizons, Fig. 7 presents results of comparison with different methods. The length of input observation is set as 10 frames (1.0s), 20 frames (2.0s) and 30 frames (3.0s). And the length of the output prediction is set as 5 frames (0.5 s), 10 frames (1.0 s), 15 frames (1.5 s) and 20 frames (2.0s). Results of 120 pairs input/output horizons indicate that the

TABLE VII
COMPARATIVE RESULTS FOR PREDICTION OF INTERACTIVE TRAJECTORIES IN WOMD(ADE/FDE)

Metrics	Agents	LSTM[41]	SL[5]	SA[33]	*TrafficPredict*[20]	*Trajectron++*[2]	*RSG+Trajectron++*	Proposed	Dynamic+proposed
ADE[m]	Pedestrian	3.43	1.70	1.67	1.41	1.33	1.37	1.35	**1.23**
	Rider	5.23	2.33	2.24	2.15	2.14	1.93	1.84	**1.66**
	Vehicle	3.94	2.45	2.32	2.03	1.85	1.79	1.65	**1.61**
	Average	4.15	2.16	2.07	1.86	1.77	1.69	1.61	**1.50**
FDE[m]	Pedestrian	6.31	4.81	4.78	4.02	4.12	4.07	3.86	**3.54**
	Rider	7.12	4.98	4.85	4.44	4.33	4.33	4.21	**3.93**
	Vehicle	5.94	4.33	3.91	3.39	3.41	3.28	3.32	**3.07**
	Average	6.45	3.70	4.51	3.95	3.95	3.89	3.57	**3.51**

proposed risk-taking style-oriented GNN can obtain the best performance in trajectory prediction. The conclusion for the selected horizons above can be verified with results in Fig. 7. Meanwhile, for all compared methods, when the horizon of observation is fixed and the prediction horizon increase from 0.5s to 2.0s, the performance of ADE and FDE decreases. It shows that the increase of output horizon will weaken the ability of prediction. A longer prediction may not help the motion planner of automated vehicles.

Fig. 8(a) shows the impact of data volume changes in BLVD on the model performance. The length of the input and output feature is 2 seconds. Experimental results indicate that when the data volume is increased to 80%, the model's performance becomes stable. In experiments involving the BLVD model, we used all available data for model training and validation. Therefore, it is impossible to further investigate the impact of larger data volumes on the model results.

To thoroughly evaluate the impact of different clustering methods, the number of clusters, and feature selection on the results, comparative analyses are conducted and presented in Table. VIII. In these comparisons, besides the KPCA used in this study, Spectral clustering (SC) is also employed. SC is a versatile algorithm used in data mining for identifying patterns in data. Unlike traditional clustering methods such as K-means, SC does not make strong assumptions about the form of the clusters. It is effective for data where the clusters have irregular or complex shapes, or when the clusters are intertwined. Table. VIII provides a comprehensive evaluation of how various clustering algorithms and feature selection influence the performance of downstream trajectory prediction. It offers a deeper understanding of the efficacy of the chosen algorithms in the framework. In Table. VIII, three kinds of clustering number are selected for KPCA and SC. Four feature selection alternatives are compared, which are detailed as follows:

- S#1: relative position, relative velocity, TTC
- S#2: relative velocity, TTC
- S#3: relative position, TTC
- S#4: TTC

This experiment focuses on evaluating the effectiveness of clustering algorithms by analyzing their impact on the performance of downstream trajectory prediction within the overall framework. According to the results in Table. VIII, KPCA outperforms SC in general. Within each clustering approach, comparing across three different clustering numbers, the trajectory prediction model performs best when the cluster data is set to four. Regarding feature selection, in various comparative experiments, the S#1 strategy achieves the best result.

TABLE VIII
THE INFLUENCE OF UNSUPERVISED CLUSTER METHOD, CLUSTERING NUMBER AND FEATURE SELECTION

Method	**Number**	**S#1**	**S#2**	**S#3**	**S#4**
KPCA	3	0.142	0.156	0.149	0.210
	4	0.121	0.135	0.152	0.174
	5	0.171	0.182	0.174	0.185
SC	3	0.149	0.291	0.187	0.205
	4	0.138	0.243	0.167	0.193
	5	0.210	0.220	0.234	0.213

D. Experimental Results and Analysis in WOMD

In the previous part, we validated the effectiveness of the proposed method in the BLVD dataset. Comparative experiments demonstrate the enhancement of the RSG module in prediction. The scalability of RSG in other models was also verified. To illustrate the model's adaptability in different datasets, the WOMD is also applied to the experiment to validate the proposed model. Table VII presents the performance of the proposed model and eight other models when the input length is 2 seconds and the output length is 5 seconds. Within the fixed coordinate system of the WOMD, the dynamic constraints of traffic participants can be flexibly taken into account. From the table, it can be seen that models that *Dynamic+proposed* achieves the best results. Compared with models without dynamic constraints, the ADE and FDE are improved by 6.8% and 1.6%, respectively. Similar to the experimental results in the BLVD dataset, overall, methods that consider social interactions are superior to those that do not. Meanwhile, the comparison results of *trajectron++* and *RSG+trajectron++* indicate that RSG can enhance the model's performance. Specifically, compared with *trajectron++*, ADE and FDE were improved by 4.5% and 1.5%, respectively. As a data-driven neural network-based model, the amount and coverage of data will greatly affect model performance. Therefore, for the Waymo dataset used, Fig. 8(b) demonstrates the model's performance under different data quantities. The experimental results show that as the data volume gradually increases to 100%, the performance of the model tends to stabilize. The model performance improved by 69.9% when the data volume increased from 10% to 50%, while the performance improves by 35.8% when the

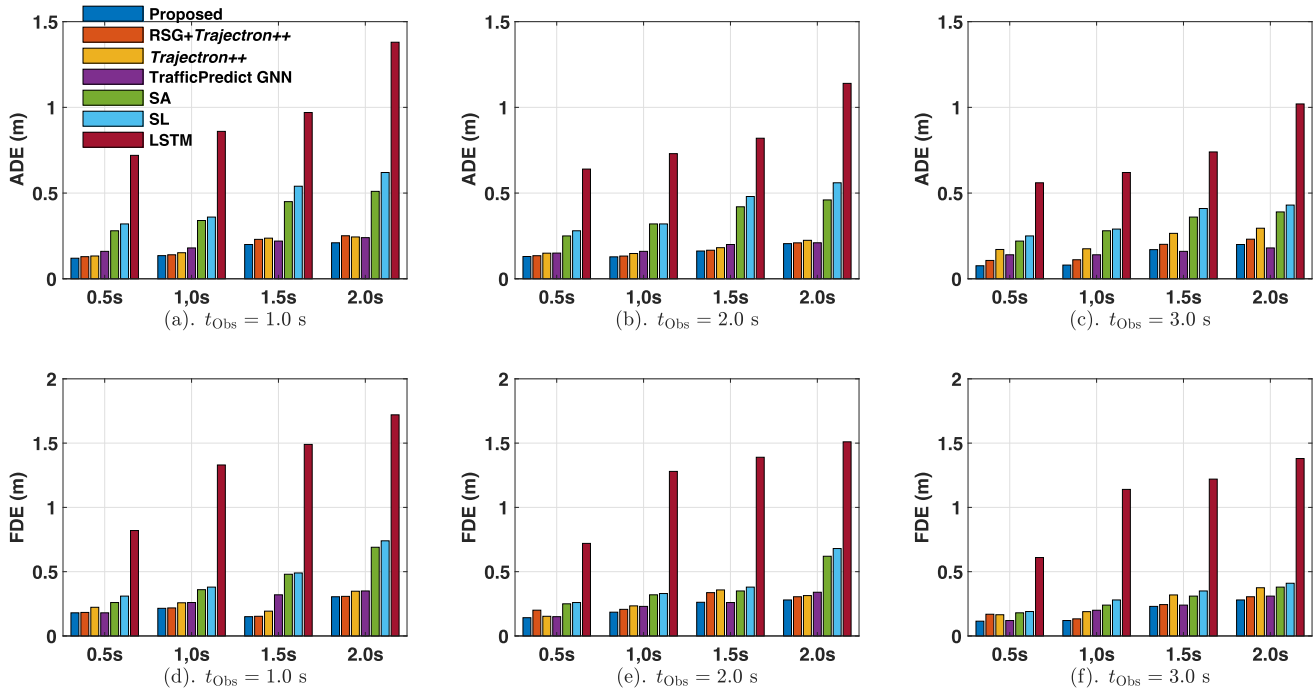


Fig. 7. Comparative results of the trajectory prediction with different input/output lengths in BLVD dataset. (a)-(c) and (d)-(f) present results of ADE and FDE, respectively. The X-axis in each figure represents different horizons of observation.

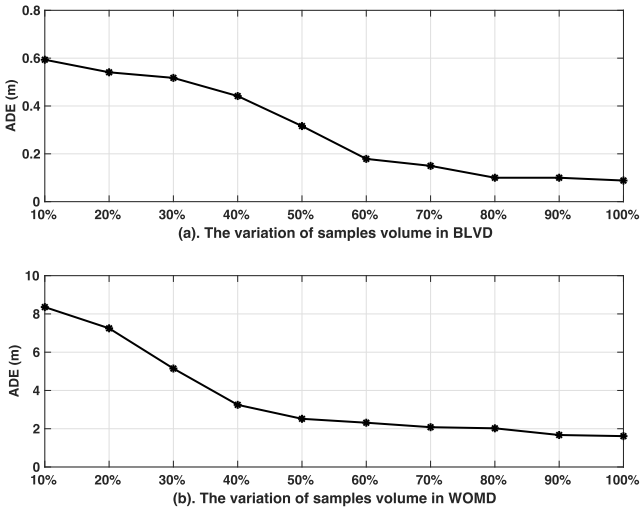


Fig. 8. The variation of ADE with different percentages of sample volume. (a). In the BLVD dataset, 100% means all samples in the dataset. (b). 100% means 20,600 samples, which are randomly selected from the WOMD.

data volume increased from 50% to 100%. The influence of data volume on model performance is affected by the type of data, the coverage of data, and the model itself. The results in Fig. 8(b) can provide a reference for the use of interaction models in the WOMD scenarios. It helps trade off the choices among the model performance requirements, model complexity, and model training consumption.

In the RSG module of the proposed framework, only the influence between the ego vehicle and the surrounding dynamic traffic participants is considered. There are two reasons: firstly, the final output of the RSG module is to assess the risk of surrounding traffic participants relative to the ego vehicle, which provides crucial input for decision-making and motion planning of the ego automated vehicle. Secondly, when the number of traffic participants increases, considering the risk between any two dynamic participants significantly increases computational demands and may prevent its online application. The final risk assessment might be a composite result of all participants, and it is infeasible to directly convey this to the ego vehicle for decision-making and motion planning. However, only focusing on the risk between the ego vehicle and surrounding vehicles in the RSG module may impact the performance of trajectory prediction. To assess the impact of this issue, an additional experiment is conducted. In this experiment, the risk-taking styles of samples on the clustering boundaries are manually changed. In the selection of data from the clustering boundaries, samples are chosen where the distance difference between the nearest cluster center and the second nearest is less than 5%. Then the modified features are directly used in the inference process of trajectory prediction. With this operation in the additional experiment, the influence of the "incorrect" classification of risk-taking styles on the performance of trajectory prediction can be evaluated. The experimental results are shown in Table. IX. It suggests that such modifications do slightly

TABLE IX
THE INFLUENCE OF WRONG RISK-TAKING STYLES GENERATION TO TRAJECTORY PREDICTION (ADE[M]/FDE[M])

	Revised model	Proposed model
Pedestrian	1.47/3.97	1.32/3.61
Vehicle	1.64/4.17	1.58/4.11
Rider	1.81/4.30	4.22

reduce performance but do not lead to a significant increase in prediction errors. It indicates that the performance of trajectory prediction is primarily affected by the modeling of interactions in heterogeneous graph learning. The generation of risk-taking styles can enhance performance to some extent but does not determine the overall performance.

VII. Conclusion

In this paper, an interactive behavior model with risk-taking styles is proposed to predict trajectories of heterogeneous traffic participants, which combines an unsupervised risk-taking style generation module and a supervised graph model. The proposed method firstly clusters the behaviors of surrounding participants and generates their risk-taking styles. Subsequently, the heterogeneous graph neural network takes specific risk-taking styles into consideration to predict interactive behaviors.

The proposed method synthesizes the advantages of unsupervised clustering and the supervised graph learning approaches. It is verified on the public BLVD dataset and obtains the best performance in the prediction of interactive behaviors compared to baseline methods. Comparative results indicate that the method can predict trajectories more accurately. The average distance error and final distance error of the proposed method are 0.087m and 0.12m with 3.0s observation and 1.0s prediction. The computation time is 10.5 frames/per second. The results show that it can provide more accurate and real-time input for motion planning in urban scenarios, which is important to guarantee the safety of AV. The improvement in performance is due to the reformulation of the heterogeneous graph model by considering specific risk-taking styles. The proposed method is a generic solution that can be applied to other interactive scenarios (e.g., urban intersections and highway merging).

This research focused on clustering and evaluating risk events with unsupervised methods. Future work will consider the constraints from static obstacles (e.g., road boundaries) and the influence of ego vehicle's future planning [42] in the modeling process.

Appendix A
Single Integrator: The Linear Dynamic Model for Pedestrian

For the single integrator system, we establish the definition of the state as the position vector, denoted as $\mathbf{p} = [x, y]^T$. The control is represented by the velocity vector, denoted as $\mathbf{u} = [\dot{x}, \dot{y}]^T$, where the dot above the variables signifies the time derivative. Furthermore, we express the linear discrete-time dynamics as follows:

$$\mathbf{p}^{t+1} = I_{2\times 2}\mathbf{p}^t + \Delta t I_{2\times 2}\dot{\mathbf{p}}^t \tag{40}$$

At each timestep t, the neural network generates the parameters of Gaussian distribution over control actions $\mathcal{N}(\mu_{\mathbf{u}}, \mathbf{\Sigma}_{\mathbf{u}})$. $\mu_{\mathbf{u}}$ and $\mathbf{\Sigma}_{\mathbf{u}}$ are

$$\mu_{\mathbf{u}} = \begin{bmatrix} \mu_{\dot{x}} \\ \mu_{\dot{y}} \end{bmatrix} \tag{41}$$

$$\mathbf{\Sigma}_{\mathbf{u}} = \begin{bmatrix} \sigma_{\dot{x}}^2 & \rho_{\dot{x}\dot{y}}\sigma_{\dot{x}}\sigma_{\dot{y}} \\ \rho_{\dot{x}\dot{y}}\sigma_{\dot{x}}\sigma_{\dot{y}} & \sigma_{\dot{y}}^2 \end{bmatrix} \tag{42}$$

where $\mu_{\dot{x}}$ and $\mu_{\dot{y}}$ represent the mean velocities in the participant's longitudinal and lateral directions, respectively. Similarly, $\sigma_{\dot{x}}$ and $\sigma_{\dot{y}}$ denote the longitudinal and lateral velocity standard deviations, respectively. Additionally, $\rho_{\dot{x}\dot{y}}$ signifies the correlation between the longitudinal velocity $\dot{x}$ and the lateral velocity $\dot{y}$. Notably, (40) characterizes a linear Gaussian system as $\mathbf{\Sigma}_{\mathbf{u}}$ is the only origin of uncertainty in the prediction model. The mean of the predicted position $\mathbf{p}^{t+1}$ at timestep $t+1$ can be obtained by (40). Meanwhile, the covariance of positions is derived as:

$$\begin{aligned} \mathbf{\Sigma}_{\mathbf{p}}^{t+1} &= I_{2\times 2}\mathbf{\Sigma}_{\mathbf{p}}^t I_{2\times 2}^T + \Delta t I_{2\times 2}\mathbf{\Sigma}_{\mathbf{u}}^t \Delta t I_{2\times 2}^T \\ &= \mathbf{\Sigma}_{\mathbf{p}}^t + (\Delta t)^2 \mathbf{\Sigma}_{\mathbf{u}}^t \end{aligned} \tag{43}$$

Appendix B
Dynamically-Extended Unicycle Model: The Nonlinear Dynamic Model for Vehicles and Riders

Usually, the control input of the unicycle model is the yaw rate and velocity. Considering that the vehicle is controlled by accelerating/braking pedals and the steering wheel, the general unicycle model is extended to the dynamically-extended unicycle model:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} v\cos(\phi) \\ v\sin(\phi) \\ \omega \\ a \end{bmatrix} \tag{44}$$

where $\dot{x}$ and $\dot{y}$ are velocities in longitudinal and lateral directions of the ego vehicle. x and y are longitudinal and lateral positions. a is the acceleration from accelerating/braking pedals' behaviors. And ω, v, and ϕ are yaw rate, velocity, and heading angle, respectively. In the extended model, we have $\mathbf{s} = \mathbf{p} = [x, y]^T$ and $\mathbf{u} = [\omega, a]^T$. In the discretization of the extended model, we make the assumption of a zero-order hold on the controls during each sampling step. In other words, we consider the control actions to be piecewise constant. Therefore, the discrete equivalent dynamics $\mathbf{s}^{t+1} = \mathbf{f}(\mathbf{s}^t, \mathbf{u}^t)$ is formulated as:

$$\begin{aligned} \begin{bmatrix} x^{t+1} \\ y^{t+1} \\ \phi^{t+1} \\ v^{t+1} \end{bmatrix} &= \begin{bmatrix} x^t \\ y^t \\ \phi^t \\ v^t \end{bmatrix} + \begin{bmatrix} v^t \cdot D_S^t + \dfrac{a^t \sin\left(\phi^t + \omega^t \Delta t\right)\Delta t}{\omega^t} + \dfrac{a^t}{\omega^t} \cdot D_C^t \\ -v^t \cdot D_C^t - \dfrac{a^t \cos\left(\phi^t + \omega^t \Delta t\right)\Delta t}{\omega^t} + \dfrac{a^t}{\omega^t} \cdot D_S^t \\ \omega^t \Delta t \\ a^t \Delta t \end{bmatrix} \end{aligned} \tag{45}$$

with $D_S^t = \frac{\sin(\phi^t + \omega^t \Delta t) - \sin(\phi^t)}{\omega^t}$ and $D_C^t = \frac{\cos(\phi^t + \omega^t \Delta t) - \cos(\phi^t)}{\omega^t}$. To avoid the singularity in (45),

we slightly change the dynamics when $|\omega| \leq 10^{-3}$. Treating ω infinitely close to 0, the discrete dynamics is modified as:

$$\begin{bmatrix} x^{t+1} \\ y^{t+1} \\ \phi^{t+1} \\ v^{t+1} \end{bmatrix} = \begin{bmatrix} x^{t} \\ y^{t} \\ \phi^{t} \\ v^{t} \end{bmatrix} + \begin{bmatrix} v^t \cos\left(\phi^t\right) \Delta t + 0.5\ a^t \cos\left(\phi^t\right) (\Delta t)^2 \\ v^t \sin\left(\phi^t\right) \Delta t + 0.5\ a^t \sin\left(\phi^t\right) (\Delta t)^2 \\ 0 \\ a^t \Delta t \end{bmatrix} \tag{46}$$

Finally, the full dynamics can be formulated as follows:

$$\begin{bmatrix} x^{t+1} \\ y^{t+1} \\ \phi^{t+1} \\ v^{t+1} \end{bmatrix} = \begin{cases} (45) & \text{if } |\omega| > 10^{-3} \\ (46) & \text{otherwise .} \end{cases} \tag{47}$$

At each timestep t, the neural network-based model generates the predicted $\mathbf{u}$ with Gaussian distributions $\mathcal{N}(\mu_{\mathbf{u}}, \mathbf{\Sigma_u})$:

$$\mu_{\mathbf{u}} = \begin{bmatrix} \mu_\omega \\ \mu_a \end{bmatrix} \tag{48}$$

$$\mathbf{\Sigma_u} = \begin{bmatrix} \sigma_\omega^2 & \rho_{\omega a}\sigma_\omega\sigma_a \\ \rho_{\omega a}\sigma_\omega\sigma_a & \sigma_a^2 \end{bmatrix} \tag{49}$$

where μ_ω and μ_a are the mean value of yaw rate and acceleration. Σ_ω and Σ_a are the standard deviation. $\rho_{\omega a}$ is the correlation of yaw rate and acceleration. The mean value of the position distribution can be obtained by (47) based on $\mu_{\mathbf{u}}$. Considering that $\mathbf{\Sigma_u}$ is the only source of uncertainty in predicted positions. The state $\mathbf{s}$ and control $\mathbf{u}$ can be linearized by the linear Gaussian system with Jacobians matrix $\mathbf{F}$ and $\mathbf{G}$ of (45):

$$\begin{aligned} \mathbf{F}^t &= \frac{\partial \mathbf{f}}{\partial \mu_{\mathbf{s}}^t} \\ &= \begin{bmatrix} 1 & 0 & v^t D_C^t - \dfrac{a^t D_S^t}{\omega^t} + \dfrac{a^t \cos\left(\phi^t + \omega^t \Delta t\right) \Delta t}{\omega^t} & D_S^t \\ 0 & 1 & v^t D_S^t + \dfrac{a^t D_C^t}{\omega^t} + \dfrac{a^t \sin\left(\phi^t + \omega^t \Delta t\right) \Delta t}{\omega^t} & -D_C^t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{50}$$

with

$$\mathbf{G}^t = \frac{\partial \mathbf{f}}{\partial \mu_{\mathbf{u}}^t} = \begin{bmatrix} G_{11}^t & \dfrac{D_C^t}{\omega^t} + \dfrac{\sin\left(\phi^t + \omega^t \Delta t\right) \Delta t}{\omega^t} \\ G_{21}^t & \dfrac{D_S^t}{\omega^t} - \dfrac{\cos\left(\phi^t + \omega^t \Delta t\right) \Delta t}{\omega^t} \\ \Delta t & 0 \\ 0 & \Delta t \end{bmatrix} \tag{51}$$

$$\begin{aligned} G_{11}^t = &\frac{v\cos(\phi + \omega\Delta t)\Delta t}{\omega} - \frac{vD_S}{\omega} - \frac{2a\sin(\phi + \omega\Delta t)\Delta t}{\omega^2} \\ &- \frac{2aD_C}{\omega^2} + \frac{a\cos(\phi + \omega\Delta t)(\Delta t)^2}{\omega} \end{aligned} \tag{52}$$

$$\begin{aligned} G_{21}^t = &\frac{v\sin(\phi + \omega\Delta t)\Delta t}{\omega} + \frac{vD_C}{\omega} + \frac{2a\cos(\phi + \omega\Delta t)\Delta t}{\omega^2} \\ &- \frac{2aD_S}{\omega^2} + \frac{a\sin(\phi + \omega\Delta t)(\Delta t)^2}{\omega}. \end{aligned} \tag{53}$$

Then, the covariance of predicted positions is formulated as:

$$\mathbf{\Sigma}_{\mathbf{p},\theta,v}^{t+1} = \mathbf{F}^t \mathbf{\Sigma}_{\mathbf{p},\theta,v}^t \mathbf{F}^{t^T} + \mathbf{G}^t \mathbf{\Sigma}_{\mathbf{u}}^t \mathbf{G}^{t^T} \tag{54}$$

REFERENCES

[1] S. Yoon, H. Jeon, and D. Kum, "Predictive cruise control using radial basis function network-based vehicle motion prediction and chance constrained model predictive control," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 10, pp. 3832–3843, Oct. 2019.

[2] T. Salzmann et al., "Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data," in *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*. Springer, 2020, pp. 683–700.

[3] 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. 1–14, Dec. 2014.

[4] L. Xin, P. Wang, C.-Y. Chan, J. Chen, S. E. Li, and B. Cheng, "Intention-aware long horizon trajectory prediction of surrounding vehicles using dual LSTM networks," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 1441–1446.

[5] 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.

[6] A. Vemula, K. Muelling, and J. Oh, "Social attention: Modeling attention in human crowds," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 4601–4607.

[7] J. Zhao, V. L. Knoop, and M. Wang, "Microscopic traffic modeling inside intersections: Interactions between drivers," *Transp. Sci.*, vol. 57, no. 1, pp. 135–155, Jan. 2023.

[8] P. Hang, C. Lv, C. Huang, J. Cai, Z. Hu, and Y. Xing, "An integrated framework of decision making and motion planning for autonomous vehicles considering social behaviors," *IEEE Trans. Veh. Technol.*, vol. 69, no. 12, pp. 14458–14469, Dec. 2020.

[9] P. Hang, C. Lv, Y. Xing, C. Huang, and Z. Hu, "Human-like decision making for autonomous driving: A noncooperative game theoretic approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 4, pp. 2076–2087, Apr. 2021.

[10] Y. Ali, Z. Zheng, M. M. Haque, and M. Wang, "A game theory-based approach for modelling mandatory lane-changing behaviour in a connected environment," *Transp. Res. C, Emerg. Technol.*, vol. 106, pp. 220–242, Sep. 2019.

[11] A. Talebpour, H. S. Mahmassani, and S. H. Hamdar, "Modeling lane-changing behavior in a connected environment: A game theory approach," *Transp. Res. Proc.*, vol. 7, pp. 420–440, Jan. 2015.

[12] X. Zhao, J. Sun, and M. Wang, "Measuring sociality in driving interaction," 2023, *arXiv:2306.13992*.

[13] X. Zhao, M. Wang, S. Fang, and J. Sun, "Towards active motion planning in interactive driving scenarios: A generic utility term of interaction activeness," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2023, pp. 1–6.

[14] A. Ji and D. Levinson, "Estimating the social gap with a game theory model of lane changing," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 10, pp. 6320–6329, Oct. 2021.

[15] X. Lu, H. Zhao, C. Li, B. Gao, and H. Chen, "A game-theoretic approach on conflict resolution of autonomous vehicles at unsignalized intersections," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 11, pp. 12535–12548, Nov. 2023.

[16] Y. Rahmati, M. K. Hosseini, and A. Talebpour, "Helping automated vehicles with left-turn maneuvers: A game theory-based decision framework for conflicting maneuvers at intersections," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 11877–11890, Aug. 2022.

[17] 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.

[18] T. Zhao et al., "Multi-agent tensor fusion for contextual trajectory prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12126–12134.

[19] 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.

[20] 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.*, vol. 33, 2019, pp. 6120–6127.

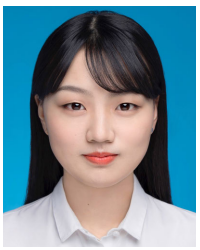
[21] S. H. Hamdar, H. S. Mahmassani, and M. Treiber, "From behavioral psychology to acceleration modeling: Calibration, validation, and exploration of drivers' cognitive and safety parameters in a risk-taking environment," *Transp. Res. B, Methodol.*, vol. 78, pp. 32–53, Aug. 2015.

[22] F. A. Mullakkal-Babu, M. Wang, X. He, B. van Arem, and R. Happee, "Probabilistic field approach for motorway driving risk assessment," *Transp. Res. C, Emerg. Technol.*, vol. 118, Sep. 2020, Art. no. 102716.

[23] J. Kim and D. Kum, "Collision risk assessment algorithm via lane-based probabilistic motion prediction of surrounding vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 9, pp. 2965–2976, Sep. 2018.
[24] B. Ivanovic and M. Pavone, "Injecting planning-awareness into prediction and detection evaluation," 2021, *arXiv:2110.03270*.
[25] Y. Yao, X. Wang, M. Xu, Z. Pu, E. Atkins, and D. Crandall, "When, where, and what? A new dataset for anomaly detection in driving videos," 2020, *arXiv:2004.03044*.
[26] H. H. S. N. Subraveti, V. L. Knoop, and B. van Arem, "Rule based control for merges: Assessment and case study," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 3006–3013.
[27] Z. Zhang, C. Lu, J. Li, Y. Xu, J. Lu, and Z. Li, "Prediction of pedestrian risky level for intelligent vehicles," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Oct. 2020, pp. 169–174.
[28] C. Zhang, J. Zhu, W. Wang, and J. Xi, "Spatiotemporal learning of multivehicle interaction patterns in lane-change scenarios," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 6446–6459, Jul. 2022.
[29] Z. Zhang, C. Lu, G. Cui, X. Meng, C. Gong, and J. Gong, "Prediction of pedestrian spatial–temporal risk levels for intelligent vehicles: A data-driven approach," *IEEE Trans. Veh. Technol.*, early access, Jan. 22, 2024, doi: 10.1109/TVT.2024.3356658.
[30] C. Lv et al., "Hybrid-learning-based classification and quantitative inference of driver braking intensity of an electrified vehicle," *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 5718–5729, Jul. 2018.
[31] Z. Li, C. Lu, Y. Yi, and J. Gong, "A hierarchical framework for interactive behaviour prediction of heterogeneous traffic participants based on graph neural network," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 9102–9114, Jul. 2022.
[32] Z. Li, J. Gong, C. Lu, and Y. Yi, "Interactive behavior prediction for heterogeneous traffic participants in the urban road: A graph-neural-network-based multitask learning framework," *IEEE/ASME Trans. Mechatronics*, vol. 26, no. 3, pp. 1339–1349, Jun. 2021.
[33] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena, "Structural-RNN: Deep learning on spatio-temporal graphs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5308–5317.
[34] Y. Li, L. Zhang, and Y. Song, "A vehicular collision warning algorithm based on the time-to-collision estimation under connected environment," in *Proc. 14th Int. Conf. Control, Autom., Robot. Vis. (ICARCV)*, Nov. 2016, pp. 1–4.
[35] U. von Luxburg, "A tutorial on spectral clustering," *Statist. Comput.*, vol. 17, no. 4, pp. 395–416, Dec. 2007.
[36] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley Interdiscipl. Rev., Comput. Statist.*, vol. 2, no. 4, pp. 433–459, Jul. 2010.
[37] K. Messaoud, I. Yahiaoui, A. Verroust-Blondet, and F. Nashashibi, "Non-local social pooling for vehicle trajectory prediction," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2019, pp. 975–980.
[38] J. Li, W. Zhan, Y. Hu, and M. Tomizuka, "Generic tracking and probabilistic prediction framework and its application in autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 9, pp. 3634–3649, Sep. 2020.
[39] J. Xue et al., "BLVD: Building a large-scale 5D semantics benchmark for autonomous driving," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 6685–6691.
[40] Z. Huang, H. Liu, J. Wu, and C. Lv, "Differentiable integrated motion prediction and planning with learnable cost function for autonomous driving," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jun. 19, 2023, doi: 10.1109/TNNLS.2023.3283542.
[41] F. Altché and A. de La Fortelle, "An LSTM network for highway trajectory prediction," in *Proc. 2017 IEEE 20th Int. Conf. Intell. Transport. Syst. (ITSC)*, Oct. 2017, pp. 353–359.
[42] B. Ivanovic, A. Elhafsi, G. Rosman, A. Gaidon, and M. Pavone, "MATS: An interpretable trajectory forecasting representation for planning and control," 2020, *arXiv:2009.07517*.

Zirui Li received the B.S. degree from Beijing Institute of Technology (BIT), Beijing, China, in 2019, where he is currently pursuing the Ph.D. degree in mechanical engineering. From June 2021 to July 2022, he was a Visiting Researcher with Delft University of Technology (TU Delft). Since August 2022, he has been a Visiting Researcher with the Chair of Traffic Process Automation, "Friedrich List" Faculty of Transportation and Traffic Sciences, TU Dresden. His research interests include interactive behavior modeling, risk assessment, and motion planning of automated vehicles.

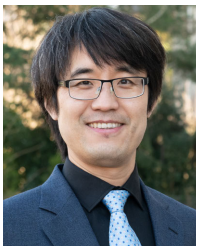
Jianwei Gong (Member, IEEE) received the B.S. degree from the National University of Defense Technology, Changsha, China, in 1992, and the Ph.D. degree from Beijing Institute of Technology, Beijing, China, in 2002. From 2011 to 2012, he was a Visiting Scientist with the Robotic Mobility Group, Massachusetts Institute of Technology, Cambridge, MA, USA. He is currently a Professor with the School of Mechanical Engineering, Beijing Institute of Technology. His research interests include intelligent vehicle environment perception and understanding, decision making, path/motion planning, and control.



Zheyu Zhang (Graduate Student Member, IEEE) received the M.S. degree from Beijing Institute of Technology, Beijing, China, in 2019. She is currently pursuing the Ph.D. degree in automotive engineering with Loughborough University, Loughborough, U.K. Her research interests include intelligent vehicles, risk assessment, and decision making under uncertainty.

Chao Lu (Member, IEEE) received the B.S. degree in transport engineering from Beijing Institute of Technology (BIT), Beijing, China, in 2009, and the Ph.D. degree in transport studies from the University of Leeds, Leeds, U.K., in 2015. In 2017, he was a Visiting Researcher with the Advanced Vehicle Engineering Centre, Cranfield University, Cranfield, U.K. He is currently an Associate Professor with the School of Mechanical Engineering, BIT. His research interests include intelligent transportation and vehicular systems, driver behavior modeling, reinforcement learning, and transfer learning and its applications.

Victor L. Knoop was born in The Netherlands in 1981. He received the M.S. degree in physics from Leiden University in 2005 and the Ph.D. degree from Delft University of Technology in 2009 on the effects of incidents on driving behavior and traffic congestion. From 2009 to 2010, he was a Post-Doctoral Researcher in lane changing with the University of Lyon. He was with Imperial College London and the University of California at Berkeley. Since 2018, he has been a tenured Associate Professor with the Transport and Planning Department, Delft University of Technology. He is currently the Co-Director of the Traffic Dynamics Modelling and Control Laboratory (TDMaC-Lab). His main research interests include interaction between microscopic and macroscopic traffic flow phenomena.



Meng Wang (Member, IEEE) received the Ph.D. degree from TU Delft in 2014. From 2014 to 2015, he was a Post-Doctoral Researcher with the Faculty of Mechanical Engineering, TU Delft. From 2015 to 2021, he was an Assistant Professor with the Department of Transport and Planning, TU Delft. Since 2021, he has been a Full Professor and the Head of the Chair of Traffic Process Automation, "Friedrich List" Faculty of Transport and Traffic Sciences, TU Dresden. His main research interests include control design and impact assessment of cooperative intelligent transportation systems. He is an Associate Editor of IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS and *Transportmetrica B*.