

D2-TPred: Discontinuous Dependency for Trajectory Prediction under Traffic Lights

Yuzhen Zhang¹, Wentong Wang², Weizhi Guo¹, Pei Lv^{1✉}, Mingliang Xu¹,
Wei Chen³, and Dinesh Manocha⁴

¹ School of Computer and Artificial Intelligence, Zhengzhou University, Zhengzhou, China. zyzzhang@gs.zzu.edu.cn,

[ielvpei, iexumingliang}@zzu.edu.cn](mailto:{ielvpei, iexumingliang}@zzu.edu.cn), ieguoweozhi@163.com

² Henan Institute of Advanced Technology, Zhengzhou University, Zhengzhou, China. wangwentong@gs.zzu.edu.cn

³ State Key Lab of CAD&CG, Zhejiang University, Zhejiang, China. chenwei@cad.zju.edu.cn

⁴ Department of Computer Science, University of Maryland, College Park, MD, USA. dmanocha@umd.edu

Abstract. A profound understanding of inter-agent relationships and motion behaviors is important to achieve high-quality planning when navigating in complex scenarios, especially at urban traffic intersections. We present a trajectory prediction approach with respect to traffic lights, D2-TPred, which uses a spatial dynamic interaction graph (SDG) and a behavior dependency graph (BDG) to handle the problem of discontinuous dependency in the spatial-temporal space. Specifically, the SDG is used to capture spatial interactions by reconstructing sub-graphs for different agents with dynamic and changeable characteristics during each frame. The BDG is used to infer motion tendency by modeling the implicit dependency of the current state on priors behaviors, especially the discontinuous motions corresponding to acceleration, deceleration, or turning direction. Moreover, we present a new dataset for vehicle trajectory prediction under traffic lights called VTP-TL. Our experimental results show that our model achieves more than 20.45% and 20.78% improvement in terms of ADE and FDE, respectively, on VTP-TL as compared to other trajectory prediction algorithms. The dataset and code are available at: <https://github.com/VTP-TL/D2-TPred>.

Keywords: Spatial dynamic interaction graph, behavior dependency graph, discontinuous dependency, traffic lights

1 Introduction

The interaction relationships and behavioral intentions of vehicles or agents are frequently used for autonomous driving [4,27,32]. A key problem is to predict the future trajectory of each vehicle or road agent, which is used to perform safe navigation or traffic forecasting [1,40,6,48]. Existing trajectory prediction methods are mainly designed to extract the spatial-temporal information from

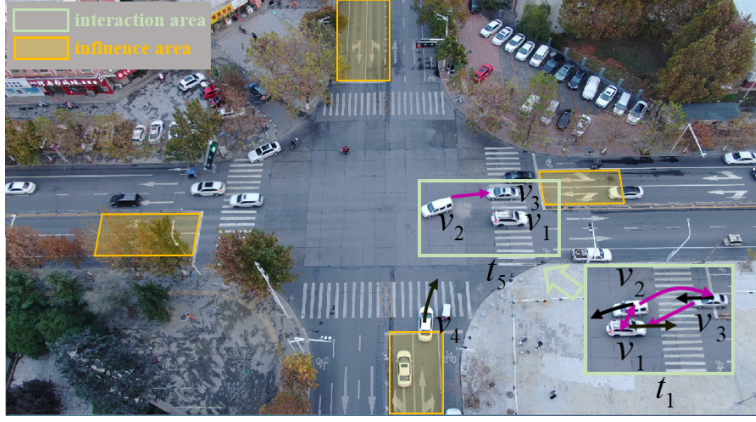


Fig. 1. Illustration of discontinuous dependency among vehicles at crossroad intersection near traffic lights. We highlight the local trajectories of four vehicles, $v_1 \dots v_4$ using black directed curves. The orange boxes represent the influence area of the corresponding traffic lights, which are fixed regions and restrict the motion behavior of vehicles while they are passing. We show the position of three vehicles, v_1, v_2, v_3 at time t_1 and t_5 , along with the corresponding green boxes which show the dynamic interaction area determined by the moving vehicles. The purple directed edges within each green box represent the interactions among vehicles. In this case, v_3 interacts with v_1 and v_2 at time t_1 . However, v_3 is not affected by v_1 at time t_5 , even though it is located in the same region. This indicates the discontinuity in the interaction between v_1 and v_3 during this time period. The vehicle v_4 located in the influence areas is not constrained by the red light because it is right-turning.

spatial interactions and behavior modeling. In terms of spatial interaction, most previous works determine the interaction among objects according to the pre-defined interaction areas, such as the entire scene [1,40,36,42,48], localized regions [2,12,6], and the area corresponding to visual attention [19]. However, these methods do not fully consider the varying interactions and dependency between neighbors that occur due to different behaviors, such as changing lanes or turning directions that can lead to new pairwise interactions. In terms of behavior dependency, these prediction algorithms obtain the relevant information of the current state from previous states based on LSTM-based methods [23,48] or graph-based approaches [29,38].

In this paper, we address the problem of trajectory prediction in areas close to traffic lights or intersections. Due to the constraints of traffic signs and traffic lights with red, green, and yellow states labeled by discrete indexes, the vehicles usually do not exhibit the first-order continuity in their movement behaviors with stopping, going straight, turning right, and turning left. Instead, their trajectory is governed by the discontinuous effects from the environment or other agents. For example, in the green boxes of Figure 1, the interactions among vehicle v_1 , v_2 and v_3 change from time t_1 to t_5 . Even though these vehicles are within the same interaction regions determined by distance which shown using green boxes,

the spatial and behavior interaction between the vehicles changes considerably and we need to model such changes. For vehicle v_4 , the most important influence on its current state is the change in its behavior due to the right-turn, rather than the movement state in adjacent timestamp. We refer to these phenomena as *discontinuous dependency (D2)*, which makes accurate spatial-temporal feature extraction extremely challenging. Current trajectory prediction methods do not fully account for this property that the trajectories of traffic agents are usually not first-order continuous due to the frequent starting and stopping motions.

Main Results: In order to model the discontinuous dependency between traffic agents, we present a new trajectory prediction approach (D2-TPred). In our formulation, we construct a spatial dynamic interaction graph (SDG) for different traffic agents in one frame. Each traffic agent is regarded as a graph node and we compute appropriate edges to model its interactions with other changing neighboring agents determined by visual scope, distance, and lane index as well as discontinuous dependencies in terms of their relative positions. Moreover, a behavior dependency graph (BDG) is computed for each agent to model the discontinuities with respect to their behaviors at previous time instances, rather than only adjacent timestamp. Specifically, to avoid the key behavioral features such as acceleration, deceleration, or turning direction, may be filtered by forget gates or the error will be accumulated in sequential prediction by RNN network, the way of dependency information passing between adjacent frames is replaced by a GAT (graph attention network) [31], and the behavior dependency is modeled along the edges in the BDG. The SDG and BDG are used as part of a graph-based network for trajectory prediction.

We also present a new dataset for vehicle trajectory prediction, VTP-TL. Our dataset consists of traffic pattern at urban intersections with different traffic rules, such as crossroads, T-junctions intersections, and roundabouts, containing 2D coordinates of vehicle trajectory and more than 1000 annotated vehicles at each traffic intersection. The novel components of our work include:

1. We propose a novel trajectory prediction approach, D2-TPred, that accounts for various discontinuities in the vehicle trajectories and pairwise interactions near traffic lights and intersections.
2. We present two types of data structure to improve the performance of graph-based networks to model dynamic interactions and vehicle behaviors. SDG is used to model spatial interactions by reconstructing appropriate sub-graphs for dynamic agents with constantly changing neighbors in each frame. BDG is used to model the dynamically changing behaviors dependency of current state on previous behaviors. The usage of SDG and BDG improves the prediction accuracy by 22.45% and 29.39% in ADE and FDE.
3. We present a new dataset VTP-TL that corresponds to traffic video data near traffic lights and intersections. This includes 150 minutes of video clips at 30fps corresponding to challenging urban scenarios. They are captured using drones at 70 – 120 meters above the traffic intersections.

2 Related Work

A brief overview of prior work on graph neural networks, interaction models, and motion pattern dependency is given.

Graph Neural Networks: Graph Neural Network (GNN) [58] can model social or other interactions between agents. Prior trajectory prediction methods based on GNN can be divided into two categories. The first is based on undirected graphs, which utilize the graph structure to explicitly construct interactions and assign the same weight for each pair of nodes, e.g., STUGCN [55], Social-STGCNN [29]. The second is based on Graph attention networks (GAT) [31], which introduces an attention mechanism into the undirected graph to calculate asymmetric influence weights for interactive agents. The GAT-based approaches, such as Social-BiGAT [43], STGAT [23], EvolveGraph [24], and SGCN [38], can flexibly model asymmetric interactions to compute spatial-temporal features and improve the prediction accuracy. Meanwhile, EvolveGraph [24] and SGCN [38] introduce graph structure inference to generate dynamic and sparse interaction. Different from these methods, we directly construct one directed graph according to interactive objects determined by the visual scope, distance, and traffic rules, and use GATs to represent the asymmetric interactions among agents.

Social Interaction Models: Social interactions and related information are used by traffic agents to make reasonable decisions to avoid potential collisions. Social force-based methods [13,33,49] use different types of force to model acceleration and deceleration forces. Social pooling based approaches [2,12,1,40] try to integrate motion information of neighbors within a radius. GNN-based techniques [42,43,24,29,41,23,46] use graph structures to directly model the interactions among near and far agents. These methods assume that the underlying agents interact with all other agents in a predefined or nearby regions. They do not account for those neighbors need to be pruned, especially moving along the opposite lanes.

Motion Models: Motion models are used to infer the motion information as part of trajectory prediction. Early studies have focused on forecasting future trajectories based on a linear model, constant velocity model, or constant acceleration model [52]. However, these simple models can not handle complex traffic patterns. Furthermore, LSTM-based approaches [2,1,23,39] and graph structure-based approaches [54,55,29,38] are proposed to model the motion trajectories. Other techniques take into account driver behavior patterns [7,3]. Giuliani et al. [17] perform precise trajectory prediction using transformer networks. Here, the states of an agent in temporal sequence are regarded as nodes to construct directed graph, further achieve the direct influence between discontinuous time-stamps, rather than only adjacent one.

3 D2-TPred

In this section, we present our novel learning-based trajectory prediction algorithm, which involves the influence of traffic lights on motion behaviors, and the architecture is shown in Figure 2.

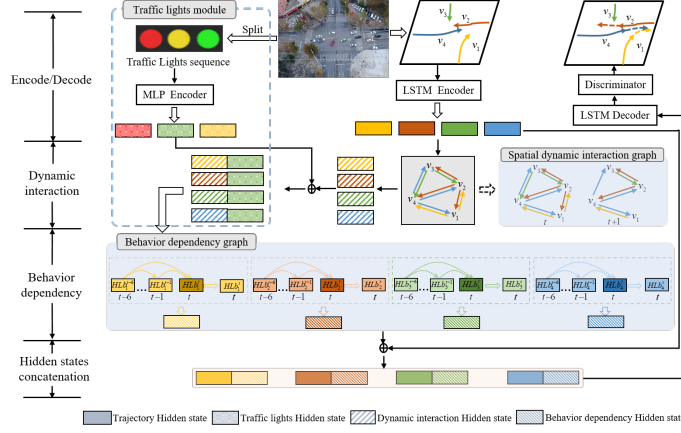


Fig. 2. Architecture of our proposed D2-TPred model. The spatial dynamic interaction graph is used to represent the dynamic interactions and we reconstruct the sub-graphs. The behavior dependency graph learns the movement features by estimating the effect of agent behavior. The discriminator is used to refine the predicted trajectories. The traffic light module is used to predict the trajectories at urban intersections.

3.1 Problem Formulation

Given spatial coordinate and traffic light state of N agents in each scenario, we aim to predict the most likely trajectories of these agents in the future. At any time t , the state of the i th agent Sq_i at time t can be denoted as $Sq_i^t = (Fid, Aid, x_i^t, y_i^t, Lid, pa_i^t, f_i^t, mb_i^t, lid_i^t, ls_i^t, lt_i^t)$, where $p_i^t = (x_i^t, y_i^t)$ represents the position coordinate and the other symbols represent the corresponding traffic light information described with more detail in Section 3.3. According to the inputs of all agents in the interval $[1 : t_{obs}]$, our method can predict their position at next moment $t_{pred} \in [t_{obs} + 1 : T]$. Different from the ground truth trajectory $Lq_i^{t_{pred}}$, $\hat{Lq}_i^{t_{pred}}$ notates the predicted trajectory.

3.2 Spatio-Temporal Dependency

Spatial Dynamic Interaction Graph. Unlike prior methods [23,29], we reconstruct the sub-graphs to model all the interactions in each frame. We illustrate our approach to model discontinuous dependency by highlighting one scenario with 7 vehicles and appropriate trajectories in Figure 3. Similar to [19], the visual area of a subject is treated as frustum, where different visual ranges are set between road and intersection by considering the characteristics of human visual system. At time t_1 , v_2 , v_3 , and v_7 are located in the visual area of neighborhood of v_1 . However, the motion behavior of v_1 is not affected by v_7 , which moves in the opposite lane. Hence, we construct sub-graph G_1 corresponding to the interactions among vehicles v_1 , v_2 , and v_3 , and sub-graph G_3 for vehicles v_5 and v_6 . Moreover, for vehicles v_4 and v_7 without nearby neighbors, we compute

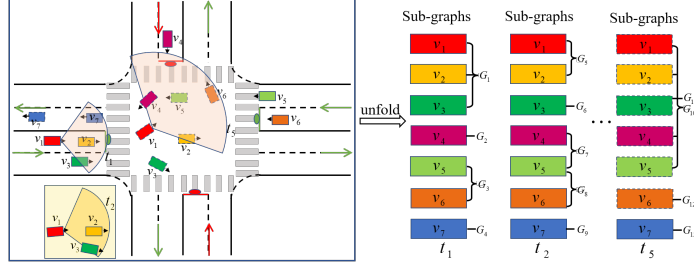


Fig. 3. The spatial dynamic interaction graph (SDG). The left part of the figure shows the scene from time t_1 to t_5 , and the right part represents the reconstructed interaction sub-graphs at different time instances.

sub-graphs G_2 and G_4 , respectively. Based on these sub-graphs, the intermediate states of these vehicles are updated. Since the interactions between the vehicles change dynamically, vehicle v_1 is not affected by vehicle v_3 at time t_2 . Even though they are within the same interaction region determined by distance, the influence of vehicle v_3 on vehicle v_1 is not the same between adjacent frames. In this manner, we reconstruct the corresponding sub-graphs G_5 , G_{13} to represent these varying interactions between the vehicles.

Considering the asymmetry of interactions among agents, we use a self-attention mechanism into these constructed directed graphs to model the spatial interactions. For agent i at time t , we first determine its interactive objects j according to the visual scope θ , distance d , and lane index Lid , and the corresponding matrix V , D , and L respectively.

$$\begin{cases} V[i, j] = 1, & \text{if } \theta(\vec{i}\vec{j}) \in \theta, \\ D[i, j] = 1, & \text{if } \|p_j^t - p_i^t\|_2 \leq d, \\ L[i, j] = 1, & \text{if } Lid_i^t = Lid_j^t, \\ R = V \times D \times L \end{cases} \quad (1)$$

where R filled with 0 and 1 represents adjacency matrix among agents, and we further construct sub-graph based on it. We then calculate the spatial state hs_i^t by integrating the hidden states h_j^t from interactive objects.

$$e_i^t = \Phi(p_i^t, W_p), \quad h_i^t = LSTM(h_i^{t-1}, e_i^t, W_l), \quad (2)$$

$$hs_i^t = \sum_{j \in \mathcal{N}_i^t} (a_{ij}^t R_{ij}^t) h_j^t, \quad (3)$$

where $\Phi(\cdot)$ is an embedding function, and e_i^t is the state vector of agent i at time t . Similar to method [14], a_{ij}^t represents the attention coefficient of agent j to i at timestamp t , W_p and W_l are embedding matrix and LSTM cell weight.

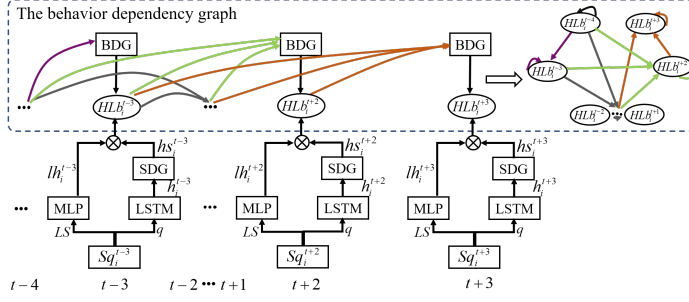


Fig. 4. The behavior dependency graph (BDG). The lower part refers to the encoding process of trajectories and traffic light signals. The upper part describes behavior dependency, where segments with the same color refer to a temporal dependency graph.

Behavior Dependency Graph. To avoid the key behavioral features may be filtered by forget gate of RNN network in process of information passing, we model the discontinuous dependency from previous behaviors to the current state by using GATs, rather than only adjacent timestamps. Specifically, for a given vehicle, its states updated by SDG are regarded as nodes. We model discontinuous dependency in the temporal sequence as edges and construct a directed graph, where the behavior information is transferred along the directed edges. The detailed architecture of the BDG for a given agent is shown in Figure 4.

Specifically, for agent i , we use directed segments with the same color to constitute an unfolded BDG, and different colors represent the behavior dependency graphs at different time instances. The BDG uses the state hs_i^t generated by the SDG. Its current state is updated and embed into the behavior dependency graph at the next time instance, where dependency weights among nodes are calculated by using a self-attention mechanism. As shown in the dashed box of Figure 4, the motion state of agent i at current moment $t+2$ is governed by the previous behaviors at time $t+1$, t , $t-1$, $t-2$, $t-3$, and $t-4$, etc., whereas the next instance $t+3$ is governed by $t+2$, $t+1$, t , $t-1$, $t-2$, and $t-3$. In this way, the updated hidden state hb_i^t for agent i at time t is calculated as follows:

$$a_i^{tt'} = \text{softmax}\left(\frac{\exp(\text{LeakyReLU}^*(\beta^T [Whs_i^t \| Whb_i^{t'}]))}{\sum_{t'} \exp(\text{LeakyReLU}^*(\beta^T [Whs_i^t \| Whb_i^{t'}]))}\right),$$

$$hb_i^t = \sum_{t' \in (t-k) \text{ and } t' \geq 0}^t a_i^{tt'} (hb_i^{t'}, hs_i^t),$$
(4)

where k represents the experimental estimated time window whose quantitative results have the lowest prediction error by obtaining more effective behavior feature. β^T is the weight vector of a single-layer feedforward neural network. t' denotes a specific time instance in the previous frames from $t-k$ to t .

3.3 Trajectory Prediction near Traffic Lights

In this section, we present two prediction schemes for the vehicle trajectory prediction. The first scheme considers the discontinuous constraints on vehicles' behaviors caused by the alternation of traffic light states, where the traffic lights are regarded as indicator signals with fixed position and alternating states. The second scheme is designed for scenarios without traffic lights, which is described in detail in the supplementary material <https://github.com/VTP-TL/D2-TPred>.

Given the observed sequence: $Sq_i^t = (Fid, Aid, x, y, Lid, pa, f, mb, lid, ls, lt)$, which is divided by the vehicle trajectory $q = (Fid, Aid, x, y, Lid, pa, f, mb)$ and the corresponding traffic light states sequence $LS = (Fid, lid, ls, lt)$. Fid , Aid , and Lid are the index of frame, vehicle, and lane where the vehicle located, respectively. lid_i^t is the traffic light index. pa_i^t describes whether vehicle v_i is within the influence area of corresponding traffic light. f_i^t indicates whether vehicle v_i is closest to the parking line in the influence area. mb_i^t represents the movement behavior of one agent, such as turning-left, turning-right, or going-straight. ls_i^t and lt_i^t respectively describe the state and duration of traffic light. We take into account that the vehicle trajectory is continuous and the traffic light states sequence is periodic and discontinuous. Therefore, the two different encoders, LSTM and MLP, are utilized to handle them and compute the corresponding hidden states h_i^t and $lh_i^t = MLP(LS_i^t, W_M)$, respectively. In SDG, we use GAT to integrate influencing features from nearby interacting agents and then compute the updated state hs_i^t of agent i . In terms of behavior dependency, we first concatenate the state hs_i^t (Eq. 4) and traffic lights state lh_i^t as input $\tilde{H}L_i^t$, and then use these results to construct BDG. Based on the BDG, we can model discontinuous constraints of traffic lights on the movement behaviors of vehicles, as shown in Figure 4. In this stage, hidden state $HLLb_i^t$ is computed as a weighted sum of $HLLb_i^{1:(t-1)}$, where the dependency weights are calculated by a self-attention mechanism. The resulting equations:

$$\tilde{H}L_i^t = \text{integrate}(hs_i^t, lh_i^t, W_l), \quad HLLb_i^t = GAT(\tilde{H}L_i^t, HLLb_i^{1:t}, W_{\tilde{\theta}}), \quad (5)$$

where $\text{integrate}(\cdot)$ is concatenation operation. W_l and $W_{\tilde{\theta}}$ are the embedding weights. To augment behavior feature and avoid the feature loss filtered by forget gates in process of sequence, the intermediate state are generated by integrating the state $HLLb_i^t$ and original state h_i^t . The predicted position is given by:

$$\hat{L}q_i^{t_{pred}} = \sigma(D_{LSTM}([HLLb_i^t, h_i^t], W_d)). \quad (6)$$

where D_{LSTM} and W_d are the decoder based LSTM and corresponding weight respectively. $\sigma(\cdot)$ represents a linear layer. Our method is also GAN-based model integrating a discriminator D_{cls} into the predicted approach, which utilizes LSTM and MLP to respectively encode the complete trajectory $([Sq_i^{T_{obs}}, \hat{L}p_i^{T_{pred}}])$ and traffic light sequence LS , and then concatenate them as the input into the *Discriminator* to output a real/fake probability $probL_i$ by a linear network.

$$probL_i = D_{cls}(LSTM([Sq_i^{t_{obs}}, \hat{L}q_i^{t_{pred}}], W_l), MLP(LS_i, W_M)) \quad (7)$$

Table 1. VTP-TL vs other state-of-the-art traffic datasets. *Size* represents the number of annotated frames. E_V and B_V represent the egocentric vision and bird’s view.

Datasets	Location	View	Night	Road type	Size	Traffic lights
CityScapes [10]	Europe	E_V	×	urban	25K	×
Argoverse [9]	USA	E_V	✓	urban	22K	×
INTERACTION [45]	International	B_V	×	urban	-	×
ApolloScape [50]	China	E_V	✓	urban + rural	144K	×
TRAF [6]	India	E_V	✓	urban + rural	72K	×
D2-city [57]	China	E_V	×	urban	700K	×
inD [5]	Germany	B_V	×	urban	-	×
Lyft Level5 [21]	USA	E_V	×	urban	46K	×
nuScenes [20]	USA/Singapore	E_V	✓	urban	40K	×
Waymo [37]	USA	E_V	✓	urban	200K	✓
Waterloo [44]	Canada	B_V	×	urban	-	✓
IDD [16]	India	E_V	×	urban + rural	10K	×
METEOR [35]	India	E_V	✓	urban + rural	2027K	×
VTP-TL	China	B_V	✓	urban	270K	✓

For each vehicle, we calculate displacement error by variety loss in [1]. The model predicts multiple trajectories K , and chooses the trajectory with the lowest distance error between them and ground-truth trajectory as the model output.

$$Loss_{variety} = \min_K \left\| Lq_i - \hat{L}q_i^K \right\|_2. \quad (8)$$

Through considering the best trajectory, the loss encourages network to cover the space of outputs that conform to the past trajectory.

3.4 VTP-TL Dataset

Although plenty of datasets have been constructed to evaluate the performance of trajectory prediction (Table 1), they rarely contained the important attributes of traffic lights except for Waymo and Waterloo. More details and their differences are described in our supplementary materials. For our new traffic dataset, VTP-TL, we use drones to hover at 70 to 120 meters above the traffic intersections in an urban setting, as statically as possible, to record vehicle trajectories passing through the area with a bird’s-eye view in the daytime corresponding to non-rush hours, rush hours, and during the evening. The dataset contains more than 150 minutes video clips, over 270k annotated frames, and more than 4 million bounding boxes for traffic vehicles at three typical urban scenarios: crossroads, T-junctions, and roundabouts intersections. Specifically, according to these recorded videos including traffic lights when they are visible, we infer the invisible state in the videos since the pattern of traffic lights are fixed. Therefore, we manually mark the related attributes of traffic light signals as discrete index, such as the index, state, duration, and position coordinate, and high definition maps such as lanes, crosswalks, and stop lines. Compared with Waymo, the traffic lights are annotated as independent objects, and modeled as agents with fixed position and changeable states in our prediction framework.

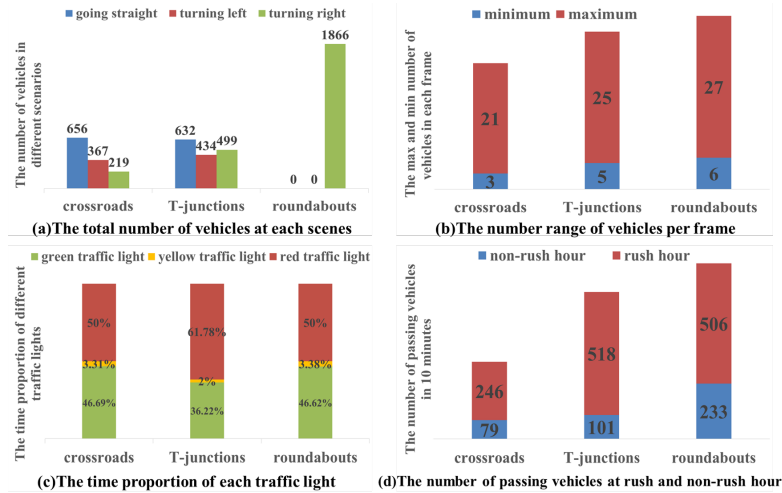


Fig. 5. We highlight the traffic light states and vehicles behaviors in various videos in VTP-TL. The detailed descriptions are given in the appendix.

The center of bounding box is regarded as the position coordinates of vehicle. Finally, we obtain a new dataset of vehicles trajectory prediction containing over 1288 vehicles driving straight, 801 vehicles turning left, and 2584 vehicles turning right. Our dataset is divided into training, validation, and testing sets at a ratio of 4:1:1, and down-sampled to 3 frames per second for experiments.

We also perform statistical analysis on the VTP-TL dataset, and the corresponding results are shown in Figure 5. The number of vehicles with different motion behaviors at each urban intersection are shown in Figure 5(a). Considering different traffic rules at various urban intersections, we also count the number range of passing vehicles per frame as Figure 5(b). Meanwhile, to ensure the effective passing of vehicles, different cycle time for traffic lights is set at different intersections (shown in Figure 5(c)). In Figure 5(d), we count the number of vehicles in the daytime of the rush and non-rush hours in 10 minutes. These descriptions can fully represent a large number of vehicles' behaviors are constrained by traffic lights. The user identifiers and exact date of publication have been masked off to protect privacy. The dataset would only be available for research purposes. More details are given in the supplementary materials.

4 Experimental Evaluation

In our experiments, the dimension of the embedding layer and the hidden state are set as 16 and 32, respectively. We also set the fixed input dimension as 64 and use the attention layer of 64. During training, the Adam optimizer is applied with a learning rate of 0.01 and batch size of 64.

Table 2. Quantitative results of prediction performance on traffic datasets. The ADE/FDE are calculated for each dataset. The bold fonts correspond to the best results with the lowest error among predicted 20 possible trajectories for each agent, except INTER (INTERACTION) where the lowest error among predicted 6 possible trajectories. For Waymo, we implement those baseline methods according to their open source code, while the other experimental values of comparison methods are all described in open papers. - denotes methods have not been validated on those datasets.

Method	ApolloScape	Method	SDD	INTER	Method	Waymo
TPNet [15]	2.23/4.70	EvolveGraph [24]	13.9/22.9	-	SGAN [1]	6.01/11.40
CS-LSTM [12]	2.14/11.70	Goal-Gan [11]	12.2/22.1	-	Social-LSTM [2]	4.05/7.59
G-LSTMS [8]	1.12/2.05	SimAug [26]	10.27/19.71	-	STGAT [23]	1.68/3.70
SGAN [1]	3.98/6.75	LB-EBM [30]	8.87/ 15.61	-	SGCN [38]	1.02/2.26
TraPHic [6]	1.28/11.67	DESIRE [25]	19.3/34.1	0.32/0.88		
NLNI [56]	1.09/1.55	HEAT [28]	-	0.19 /0.66		
AI-TP [53]	1.16/2.13	TNT [18]	-	0.21/0.67		
GRIP++ [47]	1.25/2.34	MultiPath [51]	-	0.30/0.99		
D2-TPred	1.02/1.69	D2-TPred	8.24 /15.89	0.29/ 0.62	D2-TPred	0.85/1.89

Evaluation Datasets. We evaluate proposed model on four traffic datasets, ApolloScape [50], SDD [34], INTERACTION [45], and Waymo [37]. In addition, we also report the experiments on our new dataset VTP-TL.

Evaluation Metrics. We use the same evaluating metrics as [23,41,22]. *Average displacement error* (ADE) represents the average square error between the predicted trajectory and the ground truth trajectory for all agents at all frames. *Final displacement error* (FDE) represents the mean distance between the predicted path and ground truth trajectory for all agents at the final frame.

Comparable methods. *Social-LSTM* [2] models spatial interaction by pooling mechanism. *CS-LSTM* [12], *TraPHic* [6] and *GRAPH-LSTM* [8] combine CNN with LSTM to perform trajectory prediction. *SGAN* [1], *SGCN* [38], and *Goal-Gan* [11] use GAN to model spatial interactions and physical attentions. *Social Attention* [42], *AI-TP* [53], *Trajectron++* [41], *GRIP++* [47], *NLNI* [56], *EvolveGraph* [24], and *STGAT* [23] integrate graph structure and attention mechanism to extract spatial-temporal interaction features. *TPNet* [15] and *DESIRE* [25], integrate scene context into prediction framework. *NMMP* [22] models the directed interaction with the neural motion message passing strategy. *SimAug* [26] is trained only on 3D simulation data to predict future trajectories. *LB-EBM* [30] is a probabilistic model with cost function defined in the latent space to account for the movement history and social context for diverse human trajectories. *HEAT* [28], *TNT* [18], and *MultiPath* [51] are used on INTERACTION and reported in [18].

4.1 Quantitative Evaluation

We have performed the detailed quantitative evaluation. On traffic datasets ApolloScape, SDD, INTERACTION, Waymo, and VTP-TL, the quantitative results of prediction performance for D2-TPred and other trajectory prediction methods are shown in Table 2 and Table 3.

Table 3. Quantitative results on VTP-TL dataset. We compare with the baseline methods and compute the ADE and FDE metrics by using 8 time steps to predict 12 future frames. +TL represents that traffic light states is embedded into the trajectory prediction system. The bold fonts correspond to the best results, which are the lowest error among predicted 20 possible trajectories for each agent.

Metrics	Comparable models (in pixels)					
	Social Lstm	Social Attention	SGAN	STGAT	Trajectron++	NMMP
ADE	54.328	43.648	37.63	28.279	39.01	35.15
FDE	112.635	97.614	75.35	61.762	118.37	70.35
Metrics	Comparable models+TL (in pixels)					
	Social Lstm	Social Attention	SGAN	STGAT	Trajectron++	NMMP
ADE	45.04	34.460	31.56	21.245	35.456	32.33
FDE	78.52	75.825	65.67	43.620	114.365	66.35

Traffic datasets without traffic lights: Taking benefits from the SDG and BDG to extract spatio-temporal features, our method achieves competitive performance in the datasets shown in Table 2. Specifically, the performance of our method significantly outperforms comparative methods on Apolloscape. In SDD dataset with a large number of different scenarios, we observe the lowest error on ADE and the third-lowest error on FDE, as well as the lowest error on the FDE on INTER (INTERACTION). Moreover, we also achieve the best performance on Waymo Open Motion dataset by observing 8 frames to predict the next 12 frames. These demonstrate that our model can effectively capture the dynamic changeable interaction features and behavior dependency in complex traffic scenarios. More experimental results on other datasets such as ETH-UCY, Argoverse, nuScenes and inD are described in our supplementary materials.

VTP-TL dataset with traffic lights: In this section, we describe D2-TPred+TL, which introduces traffic light states into D2-TPred approach. In Table 3, we evaluate our model against comparable methods, and all these methods against themselves with traffic lights. The experimental results show our method outperforms all other methods on the VTP-TL dataset in terms of ADE and FDE. Notably, compared with STGAT with the lowest prediction error, the ADE and FDE of D2-TPred+TL are reduced by 20.45% and 20.78%. This illustrates we can effectively model constraints of traffic lights on motion behaviors.

4.2 Ablation Studies

We present the ablation studies on VTP-TL with traffic light. This not only demonstrates the significance of each component but also highlights the benefits of modeling discontinuity due to traffic lights on vehicle movement behavior.

Evaluation of the SDG and BDG: To show the effectiveness of the SDG and BDG, we compare $S_G + B_B + TL_M + D$, $S_S + B_L + TL_M + D$ with $S_S + B_B + TL_M + D$ in Table 4. $S_S + B_B + TL_M + D$ can reduce ADE by 13.93% and 15.85%, and FDE by 17.34% and 22.46%, respectively. This directly illustrates that the SDG and BDG can effectively capture discontinuous dependency in spatial-temporal space to further improve the accuracy of prediction trajectories.

Table 4. The ablation results on VTP-TL dataset. **S** denotes spatial interaction achieved by GAT (S_G) or SDG (S_S). **B** denotes behavior dependency achieved by LSTM (B_L) or BDG (B_B). **TL** denotes traffic light encoder as LSTM (TL_L) or MLP (TL_M). **D** denotes the discriminator. The bold fonts correspond to the best results.

Setting	S		B		TL		D	Metrics
	GAT	SDG	LSTM	BDG	LSTM	MLP	D	ADE / FDE
$S_G + B_L + TL_M + D$	✓		✓			✓	✓	21.792 / 48.936
$S_G + B_B + TL_M + D$	✓			✓		✓	✓	19.635 / 41.804
$S_S + B_L + TL_M + D$		✓	✓			✓	✓	20.082 / 44.560
$S_S + B_B + TL_L + D$		✓		✓	✓		✓	17.896 / 37.629
$S_S + B_B + TL_M$		✓		✓		✓		18.626 / 39.598
$S_S + B_B + TL_M + D$		✓		✓		✓	✓	16.900 / 34.553

Evaluation of the discriminator: We introduce a discriminator to refine the predicted trajectories. By comparing $S_S + B_B + TL_M$ with $S_S + B_B + TL_M + D$ in Table 4, the performances of the latter are increased by 9.26% and 12.74% in ADE and FDE, respectively. Moreover, the discriminator contributes to improving the accuracy of predicted trajectory.

Evaluation of different encoders: Due to the distinctive characteristics of traffic light states, we use the MLP and LSTM to encode them. By comparing $S_S + B_B + TL_L + D$ and $S_S + B_B + TL_M + D$ in Table 4, utilizing MLP to capture features of traffic light states can be further improved by 5.56% and 8.17% on ADE and FDE, respectively. This illustrates that a discontinuous sequence may not be suitable for being encoded by LSTM with strong context correlation.

Evaluation of the Function of Traffic Lights: For traffic lights, we compare the methods+TL with the corresponding baseline methods. The former directly uses the VTP-TL dataset, and the latter uses a dataset that consists of *Fid*, *Aid*, *x*, and *y* attributes split from the VTP-TL dataset. As shown in Table 3, it can further increase the performance by 8.02% to 24.87% and 3.38% to 30.29% in ADE and FDE, respectively. Therefore, we can clearly validate the necessity of traffic lights in trajectory prediction at urban intersections.

4.3 Qualitative Evaluation

In Figure 6, the images of first two columns show the qualitative results derived from the Argoverse and Apolloscape. It can be seen our method without traffic lights also predicts acceptable future paths at urban intersections.

In the third column, we show the qualitative results on the VTP-TL dataset. For the first row, the current traffic light state is red on the vertical road. We only show five vehicles' trajectories, where vehicle v_1 drives straight, v_2 turns right under the red traffic light, v_3 goes straight under the green light, v_4 and v_5 are not in the influence area of traffic light signals. For v_1 , the predicted trajectory of our method is closest to the ground truth. Although the trajectories of v_2 , v_3 , v_4 , and v_5 are not affected by traffic light signals, our method also predicts acceptable trajectories. The next two images show the predicted trajectory at T-junctions and roundabout intersections, where the states of vehicles located

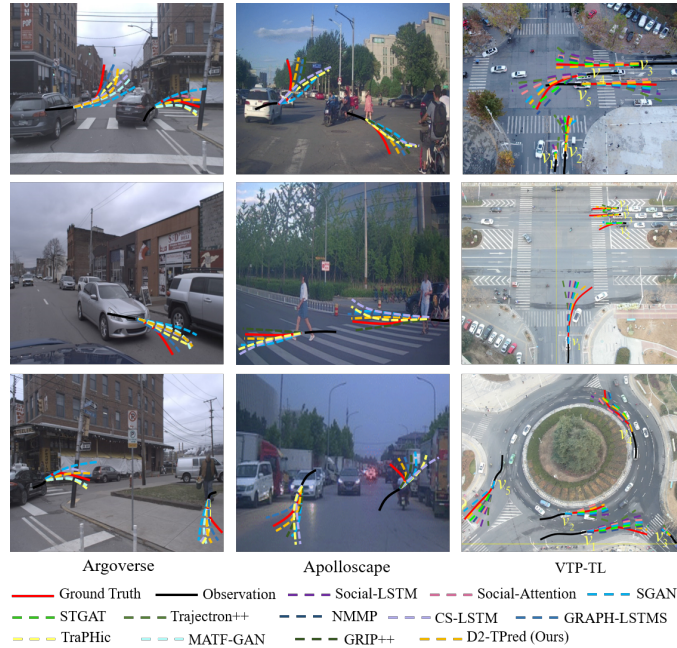


Fig. 6. The visualization results at urban intersections on traffic datasets and VTP-TL dataset. Note that the compared methods are not the same in different datasets.

in the former are changing from parking to driving under the traffic light states from red to green. This illustrates our model can flexibly respond to the dynamic changes of surrounding agents and traffic light states. Limited by the pages, more results and failure case are listed in supplementary materials.

5 Conclusions

We present *D2-TPred*, a new trajectory prediction approach by taking into traffic lights. The approach can not only model the dynamic interactions by reconstructing sub-graphs for all agents with constantly changing interaction objects (SDG), but also captures discontinuous behavior dependency by modeling the direct effects of behaviors at prior instances on the current state (BDG). Moreover, one new dataset VTP-TL for vehicles trajectory prediction with traffic lights is also released. Based on it, we describe two trajectory forecasting schemes and obtain competitive performance against other state-of-the-art.

Acknowledgments. This work was supported in part by the National Natural Science Foundation of China with Grant No.61772474 and 62036010, Zhengzhou Major Science and Technology Project with Grant No.2021KJZX0060-6. We thank all the reviewers for their valuable suggestions.

References

1. Agrim, G., Justin, J., Li, F.F., Silvio, S.: Social gan: Socially acceptable trajectories with generative adversarial networks. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2255–2264 (2018)
2. Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., Savarese, S.: Social lstm: Human trajectory prediction in crowded spaces. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 961–971 (2016). <https://doi.org/10.1109/CVPR.2016.110>
3. Angelos, M., Rohan, C., Dinesh, M.: B-gap: Behavior-guided action prediction for autonomous navigation. arXiv preprint arXiv:2011.03748 (2020)
4. Bai, H., Cai, S., Ye, N., Hsu, D., Lee, W.S.: Intention-aware online pomdp planning for autonomous driving in a crowd. In: 2015 IEEE International Conference on Robotics and Automation (ICRA). pp. 454–460 (2015)
5. Bock, J., Krajewski, R., Moers, T., Runde, S., Vater, L., Eckstein, L.: The ind dataset: A drone dataset of naturalistic road user trajectories at german intersections. In: 2020 IEEE Intelligent Vehicles Symposium (IV). pp. 1929–1934 (2019). <https://doi.org/10.1109/IV47402.2020.9304839>
6. Chandra, R., Bhattacharya, U., Bera, A., Manocha, D.: Taphic: Trajectory prediction in dense and heterogeneous traffic using weighted interactions. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 8475–8484 (2019). <https://doi.org/10.1109/CVPR.2019.00868>
7. Chandra, R., Bhattacharya, U., Mittal, T., Bera, A., Manocha, D.: Cmetric: A driving behavior measure using centrality functions. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 2035–2042 (2020)
8. Chandra, R., Guan, T., Panuganti, S., Mittal, T., Bhattacharya, U., Bera, A., Manocha, D.: Forecasting trajectory and behavior of road-agents using spectral clustering in graph-lstms. *IEEE Robotics and Automation Letters* **5**(3), 4882–4890 (2020)
9. Chang, M.F., Lambert, J., Sangkloy, P., Singh, J., Bak, S., Hartnett, A., Wang, D., Carr, P., Lucey, S., Ramanan, D., Hays, J.: Argoverse: 3d tracking and forecasting with rich maps. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition pp. 8748–8757 (2019)
10. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3213–3223 (2016)
11. Dendorfer, P., Oep, A., Laura, L.T.: Goal-GAN: Multimodal Trajectory Prediction Based on Goal Position Estimation. *Computer Vision – ACCV 2020* (2021)
12. Deo, N., Trivedi, M.M.: Convolutional social pooling for vehicle trajectory prediction. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). pp. 1468–1476 (2018)
13. Dirk, H., Peter, M.: Social force model for pedestrian dynamics. *Physical review E* **51**(5), 4282–4286 (1995)
14. Dzmitry, B., Kyunghyun, C., Yoshua, B.: Neural machine translation by jointly learning to align and translate. arXiv:1409.0473v7 (2014)
15. Fang, L., Jiang, Q., Shi, J., Zhou, B.: TpNet: Trajectory proposal network for motion prediction. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6796–6805 (2020). <https://doi.org/10.1109/CVPR42600.2020.00683>

16. Girish, V., Anbumani, S., Anoop, N., Manmohan, C., CV, J.: Idd: A dataset for exploring problems of autonomous navigation in unconstrained environments. In: 2019 IEEE Winter Conference on Applications of Computer Vision (WACV) pp. 1743–1751 (2021)
17. Giuliari, F., Hasan, I., Cristani, M., Galasso, F.: Transformer networks for trajectory forecasting. In: 2020 25th International Conference on Pattern Recognition (ICPR). pp. 10335–10342 (2020)
18. Hang, Z., Jiyang, G., Tian, L., Chen, S., Benjamin, S., Balakrishnan, V.: Tnt: Target-driven trajectory prediction. arXiv:2008.08294v2 (2020)
19. Hasan, I., Setti, F., Tsesmelis, T., Belagiannis, V., Amin, S., Del Bue, A., Cristani, M., Galasso, F.: Forecasting people trajectories and head poses by jointly reasoning on tracklets and vislets. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **43**(4), 1267–1278 (2021)
20. Holger, C., Varun, K.R.B., Lang, A.H., Sourabh, V.: nusenes: A multimodal dataset for autonomous driving. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 11621–11631 (2020)
21. Houston, J., Zuidhof, G., Bergamini, L., Ye, Y., Ondruska, P.: One thousand and one hours: Self-driving motion prediction dataset. *Conference on Robot Learning (CoRL)* (2020)
22. Hu, Y., Chen, S., Zhang, Y., Gu, X.: Collaborative motion prediction via neural motion message passing. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6318–6327 (2020)
23. Huang, Y., Bi, H., Li, Z., Mao, T., Wang, Z.: Stgat: Modeling spatial-temporal interactions for human trajectory prediction. In: 2019 International Conference in Computer Vision. pp. 6272–6281 (2019)
24. Jiachen, L., Fan, Y., Tomizuka, M., Choi, C.: Evolvegraph: Multi-agent trajectory prediction with dynamic relational reasoning. In: *Proceedings of the Neural Information Processing Systems (NeurIPS)*. pp. 1–18 (2020)
25. Lee, N., Choi, W., Vernaza, P., Choy, C.B., Torr, P.H.S., Chandraker, M.: Desire: Distant future prediction in dynamic scenes with interacting agents. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2165–2174 (2017). <https://doi.org/10.1109/CVPR.2017.233>
26. Liang, J., Jiang, L., Hauptmann, A.: Simaug: Learning robust representations from 3d simulation for pedestrian trajectory prediction in unseen cameras. arXiv preprint arXiv:2004.02022 **2** (2020)
27. Luo, Y., Cai, P., Bera, A., Hsu, D., Lee, W.S., Manocha, D.: Porca: Modeling and planning for autonomous driving among many pedestrians. *IEEE Robotics and Automation Letters* **3**(4), 3418–3425 (2018)
28. Mo, X., Huang, Z., Xing, Y., Lv, C.: Multi-agent trajectory prediction with heterogeneous edge-enhanced graph attention network. *IEEE Transactions on Intelligent Transportation Systems* pp. 1–14 (2022). <https://doi.org/10.1109/TITS.2022.3146300>
29. Mohamed, A., Qian, K., Elhoseiny, M., Claudel, C.: Social-stgcnn: A social spatio-temporal graph convolutional neural network for human trajectory prediction. In: 2020 IEEE Conference on Computer Vision and Pattern Recognition. pp. 14424–14432 (2020)
30. Pang, B., Zhao, T., Xie, X., Wu, Y.N.: Trajectory prediction with latent belief energy-based model. In: the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 11814–11824 (June 2021)
31. Petar, V., Guillem, C., Arantxa, C., Adriana, R.: Graph attention networks. *International Conference on Learning Representations (ICLR)* (2018)

32. Pongsathorn, R., Takahiro, H., Masao, N.: Motion planning and control of autonomous driving intelligence system based on risk potential optimization framework. *International Journal of Automotive Engineering* **7**(1), 53–60 (2016)
33. Ramin, M., Alexis, O., Mubarak, S.: Abnormal crowd behavior detection using social force model. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 935–942 (2009)
34. Robicquet, A., Sadeghian, A., Alahi, A., Savarese, S.: Learning social etiquette: Human trajectory understanding in crowded scenes. In: European Conference on Computer Vision. pp. 549–565 (2016)
35. Rohan, C., Mridul, M., Rahul, K., Rishitha, P., Chandrababu, N., Alok, J., Dinesh, M.: Meteor: A massive dense & heterogeneous behavior dataset for autonomous driving. arXiv preprint arXiv:2109.07648 (2021)
36. Sadeghian, A., ., V.K., Sadeghian, A., Hirose, N., Rezatofighi, H., Savarese, S.: Sophie: An attentive gan for predicting paths compliant to social and physical constraints. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1349–1358 (2020)
37. Scott, E., Shuyang, C., Benjamin, C., Chenxi, L., Hang, Z., Sabeek, P., Yuning, C., Ben, S., Charles, Q., Yin, Z.: Large scale interactive motion forecasting for autonomous driving : The waymo open motion dataset. arXiv preprint arXiv:2104.10133 (2021)
38. Shi, L., Wang, L., Long, C., Zhou, S., Zhou, M., Niu, Z., Hua, G.: Sgcnet: sparse graph convolution network for pedestrian trajectory prediction. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2021)
39. Song, X., Chen, K., Li, X., Sun, J., Hou, B., Cui, Y., Zhang, B., Xiong, G., Wang, Z.: Pedestrian trajectory prediction based on deep convolutional lstm network. *IEEE Transactions on Intelligent Transportation Systems* **22**(6), 3285–3302 (2021)
40. Song, Y., Bisagno, N., Hassan, S.Z., Conci, N.: Ag-gan: An attentive group-aware gan for pedestrian trajectory prediction. In: 2020 25th International Conference on Pattern Recognition (ICPR). pp. 8703–8710 (2021)
41. Tim, S., Boris, I., Punarjay, C., Marco, P.: Trajectron++: Multi-agent generative trajectory forecasting with heterogeneous data for control. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* pp. 683–700 (2020)
42. Vemula, A., Muellering, K., Oh, J.: Social attention: Modeling attention in human crowds pp. 1–7 (2018)
43. Vineet, K., Amir, S., Roberto, M.M., Lan, R.: Social-bigat: Multimodal trajectory forecasting using bicycle-gan and graph attention networks. In *Advances in Neural Information Processing Systems (NeurIPS)* pp. 137–146 (2019)
44. of Waterloo, U.: uwaterloo (2021), <http://wiselab.uwaterloo.ca/waterloo-multi-agent-traffic-dataset/>
45. Wei, Z., Liting, S., Di, W., Haojie, S., Aubrey, C., Maximilian, N., Julius, K., Hendrik, K., Christoph, S., Arnaud, d.L.F., Masayoshi, T.: Interaction dataset: An international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps. arXiv preprint arXiv:1910.03088 (2019)
46. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Yu, P.S.: A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems* pp. 4–24 (2019)
47. Xin, L., Xiaowen, Y., Chuah, M.C.: Grip++: Enhanced graph-based interaction-aware trajectory prediction for autonomous driving. arXiv preprint arXiv:1907.07792 (2020)

48. Xu, Y., Ren, D., Li, M., Chen, Y., Fan, M., Xia, H.: Tra2tra: Trajectory-to-trajectory prediction with a global social spatial-temporal attentive neural network. *IEEE Robotics and Automation Letters* **6**(2), 1574–1581 (2021)
49. Yuexin, M., Dinesh, M., Wenping, W.: Autorvo: Local navigation with dynamic constraints in dense heterogeneous traffic. *arXiv preprint arXiv:1804.02915* (2018)
50. Yuexin, M., Xinge, Z., Sibio, Z., Ruigang, Y., Wenping, W., Dinesh, M.: Trafficpredict: Trajectory prediction for heterogeneous traffic-agents. *Proceedings of the AAAI Conference on Artificial Intelligence* **33**(01), 6120–6127 (2019)
51. Yuning, C., Benjamin, S., Mayank, B., Dragomir, A.: Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. *arXiv:1910.05449* (2019)
52. Zernetsch, S., Kohnen, S., Goldhammer, M., Doll, K., Sick, B.: Trajectory prediction of cyclists using a physical model and an artificial neural network. In: *2016 IEEE Intelligent Vehicles Symposium (IV)*. pp. 833–838 (2016)
53. Zhang, K., Zhao, L., Dong, C., Wu, L., Zheng, L.: Ai-tp: Attention-based interaction-aware trajectory prediction for autonomous driving. *IEEE Transactions on Intelligent Vehicles* pp. 1–1 (2022). <https://doi.org/10.1109/TIV.2022.3155236>
54. Zhao, L., Song, Y., Zhang, C., Liu, Y., Wang, P., Lin, T., Deng, M., Li, H.: T-gcn: A temporal graph convolutional network for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems* **PP**(99), 1–11 (2019)
55. Zhao, Z., Liu, C.: Stugcn:a social spatio-temporal unifying graph convolutional network for trajectory prediction. In: *2021 6th International Conference on Automation, Control and Robotics Engineering (CACRE)*. pp. 546–550 (2021)
56. Zheng, F., Wang, L., Zhou, S., Tang, W., Niu, Z., Zheng, N., Hua, G.: Unlimited neighborhood interaction for heterogeneous trajectory prediction. In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. pp. 13148–13157 (2021). <https://doi.org/10.1109/ICCV48922.2021.01292>
57. Zhengping, C., Guangyu, L., Tracy, L., Bo, J., Xuefeng, S., Xinsheng, Z., Ying, L., Guobin, W., Yan, L., Jieping, Y.: D²-city: A large-scale dashcam video dataset of diverse traffic scenarios. *arXiv preprint arXiv:1904.01975* (2019)
58. Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., Sun, M.: Graph neural networks: A review of methods and applications. *AI Open* **1**, 57–81 (2020). <https://doi.org/https://doi.org/10.1016/j.aiopen.2021.01.001>