Multi-Vehicle Trajectory Prediction at Intersections using State and Intention Information

Dekai Zhu[1*], Qadeer Khan[1*] and Daniel Cremers[1]

***Abstract*— Traditional approaches to prediction of future trajectory of road agents rely on knowing information about their past trajectory. This work rather relies only on having knowledge of the current state and intended direction to make predictions for multiple vehicles at intersections. Furthermore, message passing of this information between the vehicles provides each one of them a more holistic overview of the environment allowing for a more informed prediction. This is done by training a neural network which takes the state and intent of the multiple vehicles to predict their future trajectory. Using the intention as an input allows our approach to be extended to additionally control the multiple vehicles to drive towards desired paths. Experimental results demonstrate the robustness of our approach both in terms of trajectory prediction and vehicle control at intersections. The complete training and evaluation code for this work is available here: `https://github.com/Dekai21/Multi_Agent_Intersection`.**

***Index Terms*— Trajectory prediction, Multiple vehicles, Neural network, Deep learning**

I. Introduction

Over the past decade, deep learning has made tremendous strides towards the ultimate goal of achieving full driving autonomy [1]. Self-driving vehicles deploy a suite of different sensors such as RADAR, GPS, IMU, LIDAR, cameras or their combination for various tasks such as object detection, classification, localization and navigation [2], [3], [4], [5], [6]. Among them, vision based sensors (Cameras, Lidar etc.) have been demonstrated to be most promising in achieving at par human driving performance. This is because these sensors are closest to emulating the traits of human vision in perceiving the driving environment. Coupled with other sensors, they have been successful in various tasks such as emergency braking [7], [8], lane keeping [9], [10], pedestrian detection, object tracking [11], [12] etc. However, such line-of-sight sensors mounted on ego-vehicles are primarily concerned with tasks involving single vehicles and therefore have several limitations of their own:

- They can only partially observe an environment due to limited field of view, occlusions etc. Hence, they may not be feasible for executing maneuvers at hustling areas such as traffic intersections. This is important since a sizable fraction of vehicle collisions occur at traffic intersections [13] which also tend to be more severe [14].
- Each vehicle has an independent sensor and a separate processing setup. Therefore, the combined computational power needed for all the vehicles would be high. Moreover, these resources occupy space within the ego-vehicle and may even require cooling.
- Simulated engines have played a crucial role in testing and evaluating autonomous driving algorithms. However, sensor data such as images rendered in simulation may not be a true reflection of reality. Hence, this domain shift would preclude deployment in the real world.

To overcome the issue associated with partial observability at critical areas such as intersections, a camera can be deployed in a Birds-Eye-View (BEV) manner simultaneously observing all agents in the scene as depicted in Figure 1. Such top-down images are commonplace for trajectory prediction [15], [16]. The state of the agents can be captured with a camera permanently mounted on a high infrastructure [17], [18] or using drone imagery with up to centimeter accuracy precision [19] using vision based object detection algorithms. This state information for each vehicle can then be used to predict the future trajectory or sequence of control actions. Note that each vehicle can also aggregate information about other agents before taking the appropriate action. Using accurate state information rather than ego-vehicle mounted sensors such as RGB cameras has 2 additional advantages: 1) The computational burden on the resources can be relieved since images with many pixels being processed independently on each vehicle is no longer necessary. 2) The domain shift problem caused by the rendering of images in simulation not matching reality should no longer be a concern. This is because we are using the state (location, orientation etc.) of the vehicles as an abstraction to represent information about them. Hence, with this abstraction it would be possible to train a model on one domain and test on another as we demonstrate in the Experiments.

Figure 1 further shows that the state information of all vehicles along with their desired intention to go straight, turn left or right is passed to a Multi-vehicle Trajectory Prediction (MTP) module. The MTP module predicts the future trajectory for each vehicle based on this provided information. Within the MTP, the future trajectory prediction is in turn done by the aggregation module which has shared weights across all the vehicles. This allows the model to handle an arbitrary number of vehicles in the scene. Note that to make a prediction for a particular vehicle, the aggregation module not only takes information about that specific vehicle but also considers information of other vehicles through message passing [20]. This provides each vehicle a holistic overview

[1] Computer Vision Group , CIT, Technical University of Munich.
This work was funded by the Munich Center for Machine Learning
[*] These authors contributed equally

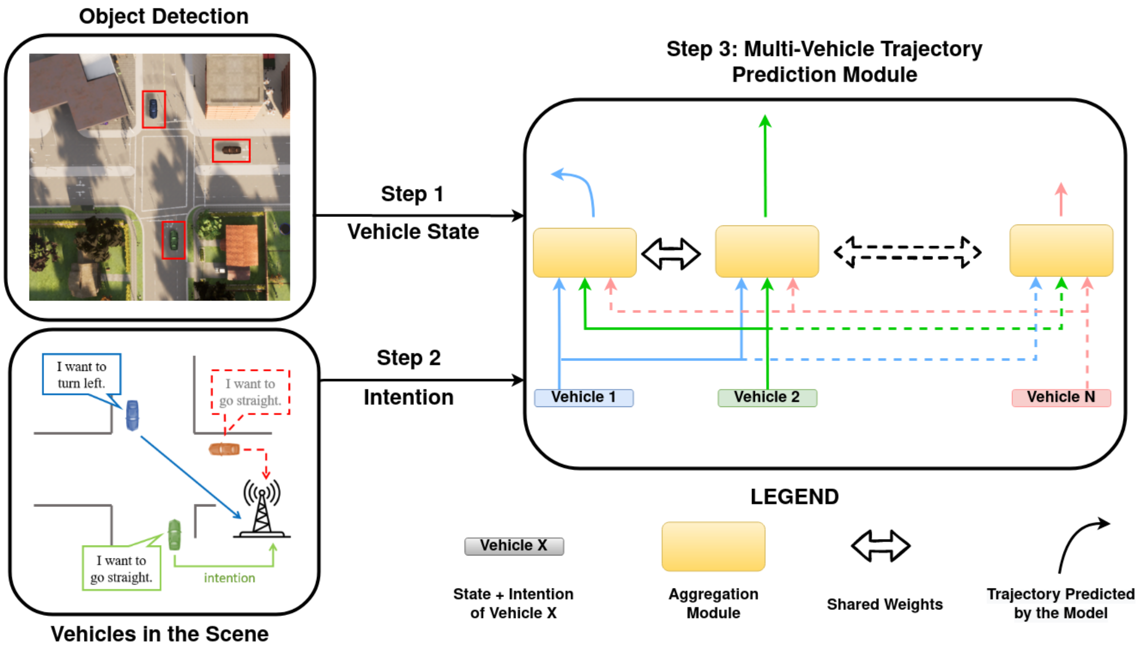

Fig. 1: **Multi-vehicle Trajectory Prediction Framework:** *Step 1:* Object detection is done on a top-down image of an intersection to extract out the state information of each vehicle in the scene. This information is sent to the Multi-vehicle Trajectory Prediction (MTP) module. *Step 2:* Meanwhile, each vehicle in the scene also sends its intention to the MTP. The intention information informs the MTP whether a certain vehicle intends to turn left, right or keep going straight at the upcoming intersection. *Step 3:* The MTP then passes this combined state and intention information to the aggregation sub-module to predict the future trajectory for each vehicle. Note that the trajectory prediction for each vehicle is not only dependent on its own state and intention information but also considers that of other vehicles too. The focus of this work is on the MTP module, where we show how the future trajectory of multiple vehicles can be predicted simultaneously using their state and intent information.

of the environment, thereby making an informed trajectory prediction. In contrast, Figure 2 shows the implications of not aggregating information from other vehicles when making trajectory predictions. To this end, the contributions of this work are summarized below:

1) We demonstrate that our approach of using only the state and intention information outperforms the approach of using past trajectory information.
2) Our model has the ability to predict the future trajectory of an arbitrary number of vehicles. It aggregates information from other vehicles; thereby giving better predictions.
3) We show that the model can be trained on one platform and tested on another.
4) Our approach of predicting the future trajectory can easily be extended to also control multiple vehicles simultaneously at intersections.
5) We have also released the entire codebase for training and testing our method. The code can be found here: https://github.com/Dekai21/Multi_Agent_Intersection.

Note that the primary emphasis of this work is the MTP module in Step 3 of Figure 1, where we show how the future trajectory of multiple vehicles can be predicted simultaneously using their state and intent information. Step 1, regarding retrieving the BEV information of the vehicles and Step 2 regarding transmission of intention information to the MTP is touched upon in the related work Section II. Based on this, the following assumptions are made:

1) Access to the state information and intention of the vehicles is available.
2) In case of control, the vehicles are capable of receiving the control commands to execute the correct maneuvers at intersections.

II. Related Work

Wireless Vehicle Communication:
Vehicle to Everything/Infrastructure (V2X/V2I) allows for wireless communication with the vehicles [21]. V2X/V2I offers various advantages over the usual line of sight sensors placed on ego-vehicles [22]. It facilitates reliable data transmission [23] even among non-line-of-sight vehicles in the immediate vicinity to give prior warnings of impending traffic jams [24], emergency braking [25], risky overtaking [26]. In our framework, it would be needed to transmit state information from and control commands to the vehicles. However, the emphasis of our work is multi-vehicle trajectory prediction and not vehicle communication. Therefore, we assume that the intention information is known by utilizing different trajectory datasets/platforms in our experiments.

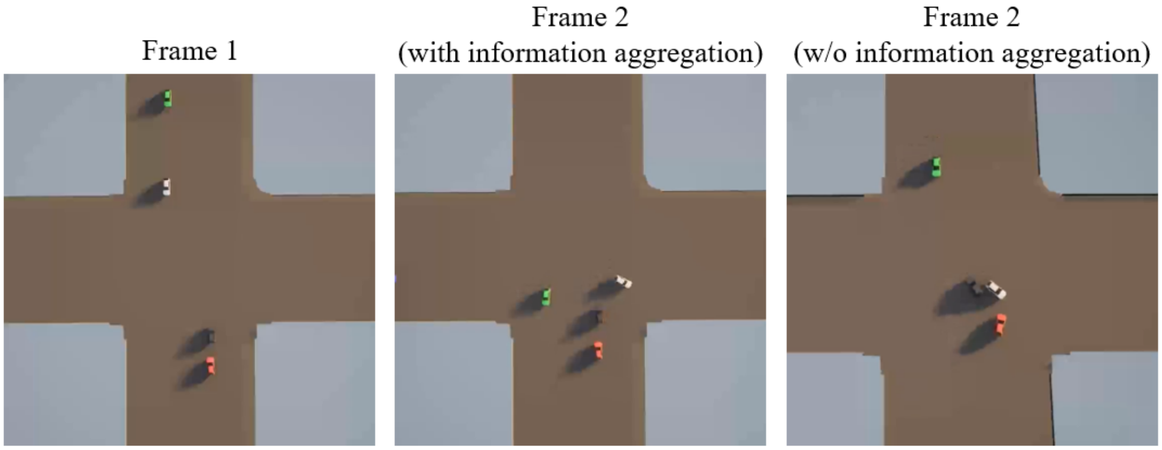

Fig. 2: Figure depicts the importance of aggregating information. *Left:* describes an initial scene comprising of 4 vehicles. The white vehicle is the first one arriving the intersection from the top and intends to turn to its left. Meanwhile, the brown and red vehicles arriving from the bottom intend to turn left and go straight respectively. *Middle:* With information aggregation, the brown and red vehicles wait until the white vehicle has left the intersection. *Right:* With no information aggregation, the brown and red vehicles start moving earlier and crash into the white vehicle.

Trajectory Datasets:
There are multiple real world datasets which provide trajectory information of various road participants from a BEV perspective. For e.g. [17], [18] collect data from top of buildings, while [15], [16] collect from drone imagery. However, these datasets contain limited proportion of trajectories comprising of vehicles in the scene. This would not be enough in terms of quantity and accuracy to train data driven learning based algorithms. We therefore train and test on the real world inD dataset [19], which addresses these limitations. Note that our approach of trajectory prediction can also be extended to control multiple vehicles. This falls under the purview of embodied agent evaluation [27] and is an emerging topic in the area of deep learning. However, none of the datasets described above provide the facility to conduct an online evaluation [28]. We therefore use the Simulation of Urban Mobility (SUMO) platform [29]. In the context of this work, SUMO allows creation and control of various scenarios at intersection for e.g. the number/intention of vehicles, the priority of the roads, structure of the intersection etc. After training on SUMO, we then evaluate the online control of the vehicles on a completely different platform i.e. Car Learning to Act (CARLA) [30]. CARLA provides the option to pass the steering and acceleration/throttle commands to maneuver multiple vehicles in the scene.

Multi-agent trajectory prediction:
Future trajectory prediction of agents using information about the social interaction between them is being used for both pedestrians [31], [32], [33], [34] and vehicles [35], [36], [37], [38]. Many such methods utilize information about the past trajectory of vehicles to make inferences about the future [39], [40]. In [41], [42] the output is probabilistic, while being multimodal in [43], [44] particularly at points where a road splits into multiple directions. However, only one of the multiple alternatives would be valid if the vehicle intends to traverse a certain direction. Our method in contrast does not require information about the past trajectory but rather only the current state of the vehicle. Also, the predictions of the future trajectory is unique as our model is conditioned on the intention of the vehicle. [45] showed that being aware of the intention of other vehicles improves merging at T-intersections. Knowledge of intention allows our task of trajectory prediction to be extended to additionally control the vehicles to reach desired targets. This is done by applying model predictive control to determine the appropriate throttle and steering angle such that the vehicle follows the predicted trajectory. We are not aware of any previous approaches that take only the current state and intention for future trajectory prediction and control of multiple vehicles. A recent work by [46], does use state and intent information but only for the task of maintaining a longitudinal safety distance between the front and rear vehicles. Moreover, their approach uses a rule based approach, whereas our approach is data-driven by training a neural network.

Multi-agent Control:
Controlling a single vehicle at an intersections is a complicated task [47]. This is further aggravated when interaction with other agents also needs to be handled [48]. The work of [49], [50], [51], [52] control the flow of multiple vehicles to minimize traffic congestion and collision at intersections. However, this is done by controlling the traffic lights. Our network on the other hand deals with controlling the individual vehicles at intersections that are void of traffic lights. [53], [54] handle multiple agents using a leader guided formation control. In our work, all vehicles are independently controlled. Other approaches to control the individual vehicles involve solving an optimization problem [55], [56]. Our approach in contrast is learning

based. [57], [58], [59], [60] uses reinforcement learning (RL) for multi agent prediction/control. However, RL methods tend to be heavily data-inefficient [61]. Our framework, on the other hand uses imitation learning complemented with an additional collision cost to prevent vehicle-to-vehicle collision when controlling multiple vehicles simultaneously.

III. Framework

In this section we describe the details of the Multi-Vehicle Trajectory (MTP) module depicted in Fig.1. It takes the state and intention information of each of the N vehicles in the scene as input and predicts their future trajectory for T timesteps ahead. We summarize the components of our framework as follows:

Input:
The information input to the MTP about each vehicle is represented by the vector $X_k \in \mathbb{R}^6$, $k = 1,2,...N$. X_k in turn comprises of 2 components: 1) State and 2) the intention of the vehicle. *State:* The state of vehicle k is in turn represented by a vector $\in \mathbb{R}^3$, described by its orientation ($\theta_k \in \mathbb{R}$) and location $S_k \in \mathbb{R}^2$ on the $x-y$ plane. *Intention:* of vehicle k represented by $I_k \in \mathbb{R}^3$ is a one hot encoded vector describing whether the vehicle intends to go either left, right or keep going straight at the upcoming intersection.

Input Transformation:
This input vector X_k for each vehicle is then passed through a series of L Multi-Layer Perceptron (MLP) layers with trainable parameters. The output of MLP layer l for each vehicle k is a latent representation given by $X_k^l \in \mathbb{R}^l$ and is specified by the following equation:

$$X_k^l = \begin{cases} X_k & l = 0 \\ \sigma(\mathbf{W}^l X_k^{l-1} + \mathbf{b}^l) & 0 < l \leq L \end{cases} \tag{1}$$

where $\mathbf{W}^l \in \mathbb{R}^{l\times(l-1)}$ and $\mathbf{b}^l \in \mathbb{R}^l$ are the trainable parameters of the MLP layer l, while σ is the ReLU non-linear activation function.

Information Aggregation:
Note that the output of the last MLP layer L for vehicle k is X_k^L and is only dependent on the latent representation of the same vehicle in the previous layer. In order to make an informed prediction of the future trajectory of a vehicle, it would be prudent to not only consider latent information about itself but also the other vehicles too. Therefore, information aggregation is done through message passing in the successive layers $l = L+1, L+2, ...L_F$. This produces a new latent representation of each vehicle given by the following equation [62]:

$$X_k^l = \begin{cases} X_k^L & l = L \\ \sigma(\mathbf{W}_s^l X_k^{l-1} + \mathbf{W}_o^l \sum\limits_{p=1,p\neq k}^{N} X_p^{l-1}) & L < l < L_F \\ \mathbf{W}_s^l X_k^{l-1} + \mathbf{W}_o^l \sum\limits_{p=1,p\neq k}^{N} X_p^{l-1} & l = L_F \end{cases} \tag{2}$$

where $\mathbf{W}_s^l$, $\mathbf{W}_o^l \in \mathbb{R}^{l\times(l-1)}$ are the trainable parameters of the aggregation layer. The output of each vehicle k in the last layer is $X_k^{L_F} \in \mathbb{R}^{2T}$. It is a prediction of the future trajectory information S_k of the vehicle k for T timesteps ahead. Note that in the experiments, we demonstrate the significance of aggregating information from neighbouring vehicles. Therein, we show that the performance of the trained model significantly deteriorates when the second term in Eq. 2 corresponding to aggregation of information from the neighbouring nodes is removed.

Note that despite having shared weights, each vehicle predicts a unique trajectory, since the input vector given by the state and intention information for each vehicle is different.

Loss Function:
The loss function used to train our model can be decomposed into the imitation loss ($L_{imitation}$) and the collision loss ($L_{collision}$). The imitation loss is the mean of the L_2 distance between the the future trajectory predicted by the model and the ground truth.

$$L_{imitation} = \frac{1}{N}\sum_{k=1}^{N}\sum_{t=1}^{T} |S_k^t - \hat{S}_k^t|_2 \tag{3}$$

where S_k^t and $\hat{S}_k^t$ are respectively the predicted and ground truth state information of vehicle k at timestep t. Meanwhile, if the future trajectory of any 2 vehicles (e.g. vehicle i and vehicle j) coincide within a certain safety distance threshold λ at the same time instance t, then a collision cost proportional to the excess is added as part of the collision loss:

$$L_{collision} = \sum_{i,j} L_{collision_{i,j}} \tag{4}$$

$$L_{collision_{i,j}} = \begin{cases} 0 & if \min\limits_{t} |S_i^t - S_j^t|_2 > \lambda \\ \lambda - \min\limits_{t} |S_i^t - S_j^t|_2 & otherwise \end{cases} \tag{5}$$

where $1 \leq i < j \leq N$ and $1 \leq t \leq T$. The purpose of the collision loss is to mitigate the propensity of vehicle-to-vehicle collision at intersections. We demonstrate the importance of this component of the loss function in the experiments.

Vehicle Control:
Note that our Multi-Vehicle Trajectory module can be extended to also control the individual vehicles. For this, we model the car with the bicycle model [63] and apply model predictive control (MPC) to optimize for the acceleration (a) and steering angle (δ) such as to follow a selected J number

of points on the predicted trajectory of the vehicle. MPC has demonstrated to be of better performance compared to other controllers [64], [65]. The equation of motion considering the bicycle model are given by:

$$\dot{x} = v \cdot cos\theta;\ \dot{y} = v \cdot sin\theta;\ \dot{v} = a;\ \dot{\theta} = v\frac{\tan\delta}{L} \tag{6}$$

where L is the wheelbase and v is the velocity of the vehicle. Meanwhile, the cost function minimized during optimization is given by:

$$\min_{a,\delta}\sum_{i=1}^{J}[(x_{\mathrm{i}}-\hat{x}_i)^2+(y_{\mathrm{i}}-\hat{y}_i)^2+(\theta_{\mathrm{i}}-\hat{\theta}_i)^2] \tag{7}$$

Data Augmentation:
Note that when it comes to online vehicle control, training merely on the recorded data may not be enough. This is because the parameters controlling the car may cause the ego-vehicle to diverge from the expected trajectory. This deviation from the norm would cause the ego-vehicle to reach scenarios not seen by the model during training, such as the lane of the oncoming traffic or the road boundaries. Since, such scenarios are not present in the training set, the prediction of future trajectory by the model would be incorrect causing the control parameters to further deviate the car from the normal trajectory such that it eventually crashes into the side barrier. Therefore, to prevent these collisions with the barrier, we additionally augment the original recorded data by adding some noise to the position of the car. The output future trajectory is then determined using model predictive control described by Eq. 6 and 7. However, the only difference is that, the optimization is to be done only for the final point on the trajectory, rather than on the J points on the known trajectory. Experiments show that inclusion of this augmentation reduces collisions with the barrier during vehicle control.

IV. Experiments

To measure the performance of our framework, we conduct both an offline and online evaluation. Offline evaluation is an assessment of future trajectory prediction of a trained model. For this we use the real world inD dataset [19].

Note that our approach of trajectory prediction is also capable of being extended to control the driving of individual vehicles at intersections. However, offline evaluation may not necessarily reflect the true driving quality. In fact, [28] showed that 2 models with similar offline metrics can have drastically different performance when deployed in a live setting. For this, online evaluation wherein the agents can actively interact with the environment is necessary. Therefore, we use the CARLA [30] platform for online evaluation with the model trained on a different platform i.e. SUMO [66]. The SUMO-CARLA co-simulation facilitates this evaluation.

A. *Offline Evaluation:*

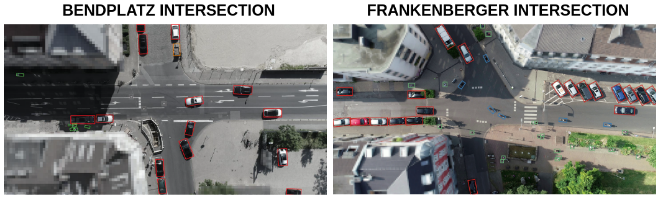

Fig. 3: A snaphot of the Bendplatz and Frankenburg intersections in Germany available in the inD dataset [19]

The Bendplatz and Frankenburg intersections from the inD dataset shown in Figure 3 have been used for offline evaluation. For each intersection, 3 track files for training and 1 for validation are randomly selected. Each track file contains 20 minutes of track records collected during different times. 4 commonly used offline evaluation metrics are used for comparison, namely: Average Displacement Error (ADE) , Final Displacement Error (FDE), Miss Rate (MR) and Collision Rate (CR) [39], [67]. For the interested reader, mathematical formulation and interpretation of these metrics, along with further information regarding the inD dataset used in the experiments is provided in the supplementary file[1].

Apart from our model, 3 additional models were trained for purpose of comparison. Description of which are described below:

Past Trajectory (VectorNet):
This model is adapted from the approach of [39]. Like our approach, it retrieves information about the surrounding vehicles. It uses an attention based mechanism for this purpose. However, this approach additionally uses information of not just the current state of the vehicle but also the past trajectory in order to better ascertain the future trajectory. Our model in contrast uses the intention of where the vehicle desires to go rather than the past trajectory.

No information aggregation:
The architecture of this model is similar to our model, except that a vehicle does not aggregate information from other vehicles in the environment. This is done by preventing message passing among the vehicles for trajectory prediction. Moreover, only the imitation learning loss is used for training.

No collision cost:
This is also similar to our approach, except that this model is trained without the additional collision cost we introduced into our imitation learning paradigm.

Ours:
This model is trained using the framework described in Section III. The model takes information about the state and intention of each vehicle in the scene and predicts the future trajectory for each based on this available information. Note that the model is capable of making each vehicle aggregate information of other vehicles via message passing. This holistic representation of the environment ought to facilitate an informed trajectory prediction that minimizes collisions between the multiple agents. The model is trained with both

[1] https://github.com/Dekai21/Multi_Agent_Intersection/tree/master/supplementary

the imitation and collision loss functions. However, note that data augmentation meant for online vehicle control is not done here.

The result of offline evaluation for all the 4 models are given in Table.I and Table.II.

B. Online Evaluation/Control

Online evaluation of the driving quality is done on the CARLA platform. However, the model is trained on data from SUMO. The intersection is created such that the vertical road (top-bottom) has higher priority over the horizontal road (left-right). The metric used for evaluation of online driving quality is the *Distance Collision Ratio (DCR)*. It is an online metric describing the distance covered by the agents before either a vehicle-to-vehicle (V2V) or vehicle-to-barrier (V2B) collision occurs. It is mathematically described as the total distance driven by all the vehicles at an intersection over the total number of V2V or V2B collisions that occur. A higher value of this metric is better.

$$DCR = \frac{1}{C} \sum_{k=1}^{N} \sum_{t=1}^{T_k - 1} \sqrt{(x_{t+1,k} - x_{t,k})^2 + (y_{t+1,k} - y_{t,k})^2} \quad (8)$$

where C is either the number of V2V or V2B collisions. Meanwhile, T_k is the number of timesteps it takes for a vehicle k to cross an intersection. Generally, it is larger for vehicles taking a left turn as opposed to those taking a right turn due to the difference in the length of the circumference of the respective curvatures.

Models used for comparison in this online evaluation are the same as described in Subsection IV-A for offline evaluation. The only difference is that 2 additional models are trained with data augmentation to enhance robustness to deviations caused by imprecise predictions. The first model is trained with data augmentation but no collision loss and the other model is trained with both augmentation and collision loss. DCR metric for V2V and V2B collision for all these models are described in Table III. For purpose of reproduciblity, the inference code for online control and the details of the SUMO-CARLA co-simulation setup are provided in the following repository: https://github.com/Dekai21/Multi_Agent_Intersection#run-the-inference-code.

C. Discussion:

In this subsection we elaborate some findings from the results.

Significance of aggregation:
As can be seen, the model with no aggregation of information from other vehicles under-performs our model. This is because, intersections are locations where plenty of interaction among multiple vehicles is expected to happen. Therefore, with no aggregation, an agent only receives information about itself and is oblivious to the state, intention and behaviour of the other vehicles. Hence, it cannot holistically look at the entire scene before taking an informed decision about its own trajectory prediction. In case of online evaluation on CARLA, we observed something interesting. Most crashes occurred not within the intersection but rather just before the vehicle enters the intersection on the non-priority road. This is because in the training set, these vehicles yield the right of way to those on the priority road by slowing down or even stopping completely before entering intersection. This is to allow the vehicles on the priority road to pass without hindrance. Only when there is no hindrance to other vehicles, the vehicle on the non-priority road moves in to the intersection. However, such situations are very rare compared to the number of samples where the vehicle on the non-priority road sits stationary. Hence, without receiving knowledge about other agents, the model memorizes to always remain stationary before entering the intersection from the non-priority road. This blocks the non-priority road and prevents other vehicles from passing. In an ideal world, if a vehicle is blocking a road, the other vehicles approaching this choke point will be expected to slow down to prevent a crash. However, these vehicles are also oblivious to the presence of the blocking vehicle and attempt to drive through it causing a crash. This particularly lowers the DCR metric especially for V2V collisions.

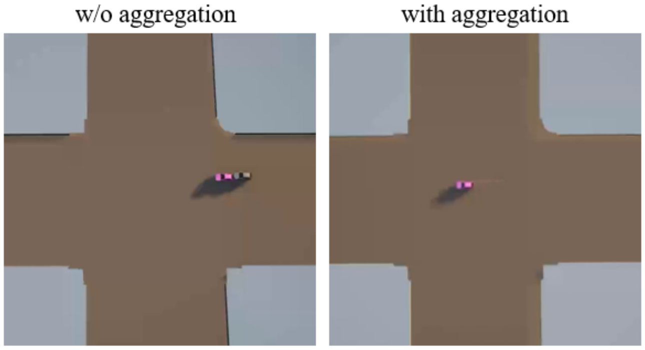

Fig. 4: Shows an example of the implications of not using aggregation in comparison to our model which uses aggregation at an intersection on the CARLA simulator. The horizontal lane (left-right) is the non-priority road, while the vertical lane (top-bottom) is the priority road.

Figure 4 demonstrates the consequences of not aggregating information. As described earlier, this leads to collisions among the vehicles before they enter the intersection due to the first stationary vehicle. On the right side of the same figure, an example scenario of our model which uses aggregation is presented. The pink vehicle desiring to go straight moves into the intersection as it is aware that there is no other vehicle at the intersection.

Contribution of Collision Cost:
It can be observed that our model which uses the collision cost penalty during training performs better than the model trained without it. The effect is even more pronounced on the online metric particularly when it comes to preventing V2V collisions. Note that the DCR metric for V2V drops

TABLE I: Results of trajectory prediction at the Bendplatz Intersection in the InD Dataset. (Lower metric values are better)

Model	Past Traj.	Intention	Message Passing	Collision Cost	ADE	FDE	MR	CR	MR+CR
Past trajectory	✓		✓		3.800	7.515	0.816	0.043	0.859
No info. aggregation		✓			1.341	2.619	0.230	0.127	0.357
No collision cost		✓	✓		1.110	2.193	0.172	0.101	0.273
Our model		✓	✓	✓	1.099	2.126	0.157	0.075	**0.232**

TABLE II: Results of trajectory prediction at the Frankenburg Intersection in the InD Dataset. (Lower metric values are better)

Model	Past Traj.	Intention	Message Passing	Collision Cost	ADE	FDE	MR	CR	MR+CR
Past trajectory	✓		✓		2.192	4.437	0.513	0.092	0.605
No info. aggregation		✓			1.958	3.924	0.411	0.147	0.558
No collision cost		✓	✓		1.752	3.518	0.341	0.112	0.453
Our model		✓	✓	✓	1.850	3.623	0.359	0.072	**0.431**

TABLE III: Results of Online Evaluation on CARLA. (Higher metric values are better)

Model	Past Traj.	Intention	Message Passing	Collision Cost	MPC Aug.	DCR (V2V)	DCR (V2B)
Past trajectory	✓		✓			99.1	158.6
No info. aggregation		✓				168.7	607.4
No collision cost & augmentation		✓	✓			722.2	515.9
No augmentation		✓	✓	✓		925.2	341.3
No collision cost		✓	✓		✓	753.6	1256.0
Our model		✓	✓	✓	✓	**3915.0**	**1957.5**

significantly when this loss component is removed from the training. The utility of the collision cost is that it has the ability to make slight modifications to correct the trajectory of the vehicles if it senses a potential collision thereby providing it with the ability to evade other vehicles. The supplementary material contains a video demonstrating the implications when collision cost is not used as opposed to our approach.

Importance of Data Augmentation:
Note that we introduced data augmentation to prevent the vehicle from deviating and crashing into road barriers during online evaluation. Comparing the performance of the model trained without data augmentation shows that the DCR metric is significantly reduced particularly for V2B collisions. Our model in contrast was trained with data samples at deviated positions from the normal trajectory. Hence, even if the model were to end up at divergent positions during online inference, it would know the corrective action to take to bring the vehicle back on track. This prevents crashes with the barrier or other vehicles if they are in the way.

Past Trajectory information:
Recall that the model in [39] uses past trajectory information of a vehicle in order to predict the future trajectory. Hence, such models have a probabilistic interpretation, wherein the precise future trajectory tends to be fuzzy and begins to become more precise by the time the vehicle reaches well into the intersection. In contrast, since our model is provided with information about the intention of the vehicle, the predictions are unique and much more accurate as can be seen from the results. This intention allows our approach

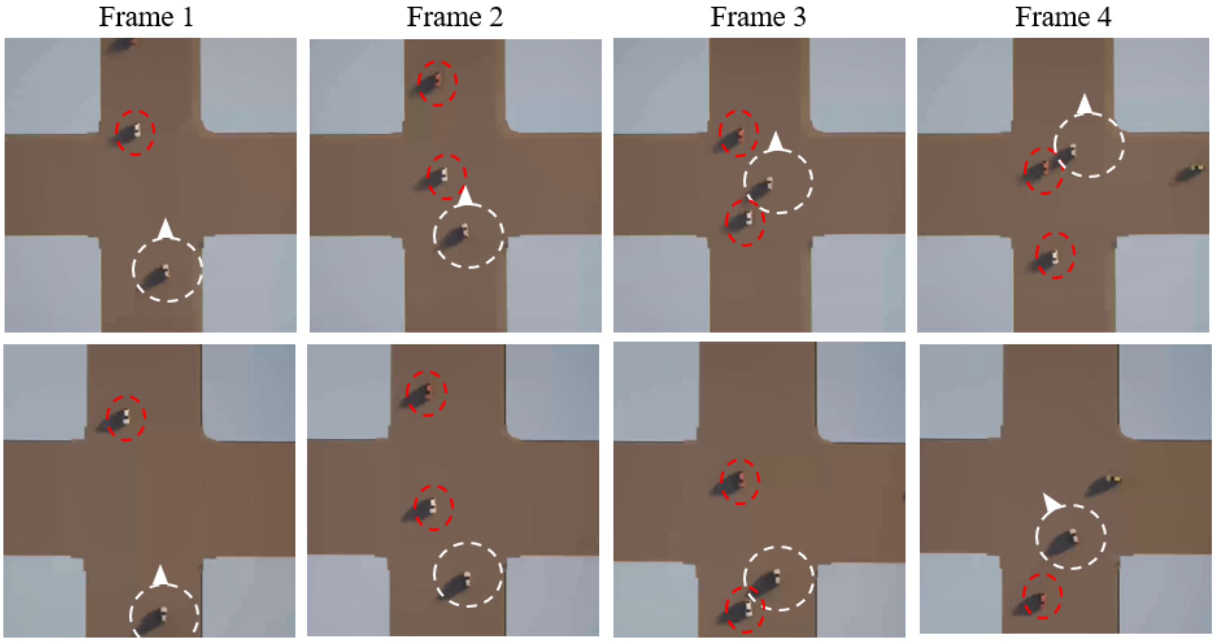

Fig. 5: Demonstrates how intention can be used to control the behaviour and interaction among the vehicles. In the first row of images, the white circled vehicle coming from the bottom desires to go straight. It keeps moving without yielding to any other vehicle. In the second row, the intention of the same vehicle is modified to turn left. In this case, the white vehicle slows down to yield to the red circled vehicles which are moving straight. The white vehicle only starts executing the left turn once the red vehicles have passed. The arrow on the white circle represents the direction of motion. There is no arrow in case the white circled vehicle is stationary. Note that the vehicle coming from the right intends to turn right, so it is not a hindrance when the white circled vehicle intends to turn left.

to be extended to vehicle control. Figure 5 shows that by manipulating the intention, the interaction among the vehicles is adjusted accordingly. This flexibility in changing the behaviour is only possible due to the capability derived from using intention of the vehicle at the input. Not only are the offline trajectory predictions more accurate (see Table I and II) but the online control is also more robust (see Table III) in comparison to using past trajectory information.

Domain Adaptation:
Note that our model trained only on data from the SUMO platform to predict the future trajectory can also be used to control the vehicle on a completely different platform. In this case, it is the CARLA platform. Note that data from CARLA was not available to the model during training. The reason for this successful adaptation of the model to different domains is because we are using the state information of the vehicle as the representation. This representation remains consistent across different platforms/domains. Hence, the model is immune to the source of origin of this representation i.e. CARLA or SUMO. Other representations such as images have difficulty in switching between different domains, weather/lighting conditions etc. For e.g. a control model trained on images from a sunny weather condition would have difficulty controlling the vehicle in a rainy weather condition even though the domains may be the same [68].

Note that the entire code for training and conducting both offline along with online evaluation is contained in the following repository: https://github.com/Dekai21/Multi_Agent_Intersection.

V. Conclusion

In this paper, we demonstrated how the trajectory for multiple vehicles can be predicted simultaneously at intersections. This is done by utilizing their state and intention information. This allowed extending the approach to additionally controlling the vehicles to move towards desired directions. Aggregating information of other vehicles further facilitated each vehicle to make better informed decisions. Our framework is also capable of being trained on one domain while being tested on another domain, data of which was not seen during training.

References

[1] SAE-International, "Taxonomy & definitions for terms related to driving automation systems for on-road motor vehicles," *SAE International*, 2021.

[2] Kadir Korkmaz, "Producing the location information with the kalman filter on the gps data for autonomous vehicles," in *2017 25th Signal Processing and Communications Applications Conference (SIU)*, 2017.

[3] Shaojiang Zhang, Yanning Guo, Qiang Zhu, and Zhiyuan Liu, "Lidar-imu and wheel odometer based autonomous vehicle localization system," in *Chinese Control And Decision Conference (CCDC)*, 2019.

[4] Hermosa Almeida et al., "Autonomous navigation of a small-scale ground vehicle using low-cost imu/gps integration for outdoor applications," in *IEEE International Systems Conference (SysCon)*, 2019.

[5] Ankith Manjunath et al., "Radar based object detection and tracking for autonomous driving," in *IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM)*, 2018.

[6] Myeon-gyun Cho, "A study on the obstacle recognition for autonomous driving rc car using lidar and thermal infrared camera," in *Eleventh International Conference on Ubiquitous and Future Networks (ICUFN)*, 2019.

[7] Heong-tae Kim and Bongsob Song, "Vehicle recognition based on radar and vision sensor fusion for automatic emergency braking," in *2013 13th International Conference on Control, Automation and Systems (ICCAS 2013)*, 2013, pp. 1342–1346.

[8] Jinghua Guo, Ping Hu, and Rongben Wang, "Nonlinear coordinated steering and braking control of vision-based autonomous vehicles in emergency obstacle avoidance," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 11, pp. 3230–3240, 2016.

[9] Farzeen Munir et al., "Ldnet: End-to-end lane marking detection approach using a dynamic vision sensor," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, 2022.

[10] Zeng Li, Shaosong Li, Zheng Li, Gaojian Cui, and Xiaodong Wu, "Lane keeping of intelligent vehicle under crosswind based on visual navigation," in *2018 5th International Conference on Information Science and Control Engineering (ICISCE)*, 2018, pp. 290–294.

[11] Aleksandr Kim, Aljosa Osep, and Laura Leal-Taixe, "Eagermot: 3d multi-object tracking via sensor fusion," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021.

[12] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer, "Trackformer: Multi-object tracking with transformers," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022.

[13] R. Marzoug, N. Lakouari, H. Ez-Zahraouy, B. Castillo Téllez, M. Castillo Téllez, and L. Cisneros Villalobos, "Modeling and simulation of car accidents at a signalized intersection using cellular automata," *Physica A: Statistical Mechanics and its Applications*, vol. 589, pp. 126599, 2022.

[14] Gang-Len Chang and Hua Xiang, "The relationship between congestion levels and accidents," Tech. Rep., STATE HIGHWAY ADMINISTRATION, 2003.

[15] Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese, "Learning social etiquette: Human trajectory understanding in crowded scenes," in *Computer Vision – ECCV 2016*, Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, Eds., Cham, 2016, pp. 549–565, Springer International Publishing.

[16] Dongfang Yang et al., "Top-view trajectories: A pedestrian dataset of vehicle-crowd interaction from controlled experiments and crowded campus," in *2019 IEEE Intelligent Vehicles Symposium (IV)*, 2019.

[17] Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski, "Crowds by example," *Computer Graphics Forum*, vol. 26, 2007.

[18] S. Pellegrini, A. Ess, K. Schindler, and L. van Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," in *2009 IEEE 12th International Conference on Computer Vision*, 2009.

[19] Julian Bock, Robert Krajewski, Tobias Moers, Steffen Runde, Lennart Vater, and Lutz Eckstein, "The ind dataset: A drone dataset of naturalistic road user trajectories at german intersections," in *2020 IEEE Intelligent Vehicles Symposium (IV)*, 2020, pp. 1929–1934.

[20] Justin Gilmer et al., "Neural message passing for quantum chemistry," in *International conference on machine learning*. PMLR, 2017.

[21] Meriem Houmer, Mariyam Ouaissa, and Mariya Ouaissa, "Secure authentication scheme for 5g-based v2x communications," *Procedia Computer Science*, vol. 198, pp. 276–281, 2022.

[22] Fabian de Ponte Müller, Estefania Munoz Diaz, and Ibrahim Rashdan, "Cooperative positioning and radar sensor fusion for relative localization of vehicles," in *2016 IEEE Intelligent Vehicles Symposium (IV)*, 2016, pp. 1060–1065.

[23] Jean-Philippe Vasseur and Adam Dunkels, "Chapter 22 - smart cities and urban networks," in *Interconnecting Smart Objects with IP*, Jean-Philippe Vasseur and Adam Dunkels, Eds., pp. 335–351. Morgan Kaufmann, Boston, 2010.

[24] Ignacio Llatser, Thomas Michalke, Maxim Dolgov, Florian Wildschütte, and Hendrik Fuchs, "Cooperative automated driving use cases for 5g v2x communication," in *2019 IEEE 2nd 5G World Forum (5GWF)*, 2019, pp. 120–125.

[25] Feng Zhao and Leonidas J. Guibas, "8 - applications and future directions," in *Wireless Sensor Networks*, Feng Zhao and Leonidas J. Guibas, Eds., The Morgan Kaufmann Series in Networking, pp. 291–306. Morgan Kaufmann, San Francisco, 2004.

[26] Ruoqi Deng, Boya Di, and Lingyang Song, "Cooperative collision avoidance for overtaking maneuvers in cellular v2x-based autonomous driving," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 4434–4446, 2019.

[27] Peter Anderson, Angel X. Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, and Amir R. Zamir, "On evaluation of embodied navigation agents," *CoRR*, vol. abs/1807.06757, 2018.

[28] Felipe Codevilla, Antonio M Lopez, Vladlen Koltun, and Alexey Dosovitskiy, "On offline evaluation of vision-based driving models," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 236–251.

[29] Pablo Alvarez Lopez et al., "Microscopic traffic simulation using sumo," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 2575–2582.

[30] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, 2017.

[31] Haowen Tang, Ping Wei, Jiapeng Li, and Nanning Zheng, "Evostgat: Evolving spatiotemporal graph attention networks for pedestrian trajectory prediction," *Neurocomputing*, vol. 491, pp. 333–342, 2022.

[32] Yusheng Peng, Gaofeng Zhang, Jun Shi, Benzhu Xu, and Liping Zheng, "Srai-lstm: A social relation attention-based interaction-aware lstm for human trajectory prediction," *Neurocomputing*, vol. 490, pp. 258–268, 2022.

[33] Hao Zhou, Dongchun Ren, Huaxia Xia, Mingyu Fan, Xu Yang, and Hai Huang, "Ast-gnn: An attention-based spatio-temporal graph neural network for interaction-aware pedestrian trajectory prediction," *Neurocomputing*, vol. 445, pp. 298–308, 2021.

[34] Fang Zheng et al., "Unlimited neighborhood interaction for heterogeneous trajectory prediction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[35] Xiaoyu Mo, Zhiyu Huang, Yang Xing, and Chen Lv, "Multi-agent trajectory prediction with heterogeneous edge-enhanced graph attention network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 9554–9567, 2022.

[36] Hengbo Ma, Yaofeng Sun, Jiachen Li, and Masayoshi Tomizuka, "Multi-agent driving behavior prediction across different scenarios with self-supervised domain knowledge," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, 2021.

[37] Tianyang Zhao, Yifei Xu, Mathew Monfort, Wongun Choi, Chris Baker, Yibiao Zhao, Yizhou Wang, and Ying Nian Wu, "Multi-agent tensor fusion for contextual trajectory prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[38] Xin Li et al., "Grip: Graph-based interaction-aware trajectory prediction," in *IEEE Intelligent Transportation Systems Conference*, 2019.

[39] Jiyang Gao et al., "Vectornet: Encoding hd maps and agent dynamics from vectorized representation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[40] Xiaosong Jia et al., "Multi-agent trajectory prediction by combining egocentric and allocentric views," in *Proceedings of the 5th Conference on Robot Learning*. 2022, PMLR.

[41] Jiachen Li et al., "Interaction-aware multi-agent tracking and probabilistic behavior prediction via adversarial learning," in *International conference on robotics and automation (ICRA)*. IEEE, 2019.

[42] Hengbo Ma, Jiachen Li, Wei Zhan, and Masayoshi Tomizuka, "Wasserstein generative learning with kinematic constraints for probabilistic interactive driving behavior prediction," in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 2477–2483.

[43] NN Sriram, Buyu Liu, Francesco Pittaluga, and Manmohan Chandraker, "Smart: Simultaneous multi-agent recurrent trajectory prediction," in *European Conference on Computer Vision*. Springer, 2020.

[44] Francesco Marchetti, Federico Becattini, Lorenzo Seidenari, and Alberto Del Bimbo, "Multiple trajectory prediction of moving agents with memory augmented networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2020.

[45] Volkan Sezer, Tirthankar Bandyopadhyay, Daniela Rus, Emilio Frazzoli, and David Hsu, "Towards autonomous navigation of unsignalized intersections under uncertainty of human driver intent," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 3578–3585.

[46] Hao M. Wang, Sergei S. Avedisov, Onur Altintas, and Gábor Orosz, "Multi-vehicle conflict management with status and intent sharing under time delays," *IEEE Transactions on Intelligent Vehicles*, pp. 1–14, 2022.

[47] Zi-jia Wang, Xue-mei Chen, Pin Wang, Meng-xi Li, Han Zhang, et al., “A decision-making model for autonomous vehicles at urban intersections based on conflict resolution,” *Journal of advanced transportation*, vol. 2021, 2021.

[48] Qiang Ge, Qi Sun, Zhen Wang, Shengbo Eben Li, Ziqing Gu, Sifa Zheng, and Lyuchao Liao, “Real-time coordination of connected vehicles at intersections using graphical mixed integer optimization,” *IET Intelligent Transport Systems*, vol. 15, no. 6, pp. 795–807, 2021.

[49] Bo Liu and Zhengtao Ding, “A distributed deep reinforcement learning method for traffic light control,” *Neurocomputing*, vol. 490, pp. 390–399, 2022.

[50] Zhengyi Ge, “Reinforcement learning-based signal control strategies to improve travel efficiency at urban intersection,” in *2020 International Conference on Urban Engineering and Management Science (ICUEMS)*, 2020, pp. 347–351.

[51] Maheen Firdous, Fasih Ud Din Iqbal, Nouman Ghafoor, Nauman Khalid Qureshi, and Noman Naseer, “Traffic light control system for four-way intersection and t-crossing using fuzzy logic,” in *2019 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, 2019, pp. 178–182.

[52] Mengyu Guo, Pin Wang, Ching-Yao Chan, and Sid Askary, “A reinforcement learning approach for intelligent traffic signal control at urban intersections,” in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 4242–4247.

[53] Sathishkumar Moorthy and Young Hoon Joo, “Distributed leader-following formation control for multiple nonholonomic mobile robots via bioinspired neurodynamic approach,” *Neurocomputing*, vol. 492, pp. 308–321, 2022.

[54] Shude He, Rourou Xu, Zhijia Zhao, and Tao Zou, “Vision-based neural formation tracking control of multiple autonomous vehicles with visibility and performance constraints,” *Neurocomputing*, vol. 492, pp. 651–663, 2022.

[55] Michael W Levin and David Rey, “Conflict-point formulation of intersection control for autonomous vehicles,” *Transportation Research Part C: Emerging Technologies*, vol. 85, pp. 528–547, 2017.

[56] Maximilian Kloock et al., “Distributed model predictive intersection control of multiple vehicles,” in *IEEE intelligent transportation systems conference (ITSC)*. IEEE, 2019.

[57] Di Wang, Hongbin Deng, and Zhenhua Pan, “Mrcdrl: Multi-robot coordination with deep reinforcement learning,” *Neurocomputing*, vol. 406, pp. 68–76, 2020.

[58] Praveen Palanisamy, “Multi-agent connected autonomous driving using deep reinforcement learning,” in *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020, pp. 1–7.

[59] Xianjie Zhang, Yu Liu, Xiujuan Xu, Qiong Huang, Hangyu Mao, and Anil Carie, “Structural relational inference actor-critic for multi-agent reinforcement learning,” *Neurocomputing*, vol. 459, pp. 383–394, 2021.

[60] David Simões, Nuno Lau, and Luís Paulo Reis, “Multi-agent actor centralized-critic with communication,” *Neurocomputing*, vol. 390, pp. 40–56, 2020.

[61] Nelson Vithayathil Varghese and Qusay H. Mahmoud, “A survey of multi-task deep reinforcement learning,” *Electronics*, vol. 9, 2020.

[62] Christopher Morris et al., “Weisfeiler and leman go neural: Higher-order graph neural networks,” in *AAAI conference on artificial intelligence*, 2019.

[63] Danwei Wang and Feng Q, “Trajectory planning for a four-wheel-steering vehicle,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2001.

[64] Jia Liu et al., “Simulation performance evaluation of pure pursuit, stanley, lqr, mpc controller for autonomous vehicles,” in *IEEE International Conference on Real-time Computing and Robotics*, 2021.

[65] Moveh Samuel et al., “Lane keeping maneuvers using proportional integral derivative (pid) and model predictive control (mpc),” *Journal of Robotics and Control (JRC)*, vol. 2, no. 2, pp. 78–82, 2021.

[66] Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wießner, “Microscopic traffic simulation using sumo,” in *The 21st IEEE International Conference on Intelligent Transportation Systems*. 2018, IEEE.

[67] Wei Zhan et al., “INTERACTION Dataset: An INTERnational, Adversarial and Cooperative moTION Dataset in Interactive Driving Scenarios with Semantic Maps,” *arXiv:1910.03088 [cs, eess]*, 2019.

[68] Q. Khan, P. Wenzel, D. Cremers, and L. Leal-Taixé, “Towards generalizing sensorimotor control across weather conditions,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.