Prediction of Autonomous Vehicle Trajectories in Turnaround Scenarios

George Daoud [1,2,*], Mohamed El-Darieby [1], and Khalid Elgazzar[1]
[1]Faculty of Engineering and Applied Science, Ontario Tech University,Oshawa, L1G 0C5, ON, Canada
[2]Faculty of Engineering, Assiut University, Assiut, Egypt
George.Daoud@OntarioTechU.ca, Mohamed.El-Darieby@OntarioTechU.ca, khalid.elgazzar@ontariotechu.ca
*corresponding author

Abstract—Trajectory prediction of road agents plays important role in road traffic safety. Both autonomous vehicles and roadside units can benefit from it. Other than the prediction accuracy, many design constraints should be taken into consideration like the simplicity of the model. A simple model will be executed using the limited available resources of IoT units in real-time. In this paper, a model based on a Graph Neural Network is proposed that uses both the spatial and temporal information for trajectory prediction in a fully context-aware environment. The model solves the normal issues of Deep Graph Neural Networks like over-smoothing. It also predicts the trajectory of all the surrounding agents collectively without the need to iterate it. Also, the computational workload can be distributed among road agents using their communication capabilities to cooperate in predicting the trajectories.

Keywords–Road Safety; Spatial-Temporal model; Vehicle Trajectory Prediction; Deep Graph Neural Network

1. Introduction

The prediction of the trajectory of autonomous vehicles (AV) is a major component of the autonomous vehicles technology stack [1]. AV trajectory prediction helps a vehicle determine what's the expected driving actions of surrounding vehicles. Other components in the stack include sensing where sensors (cameras, lidar, radar, etc.) collect raw data about the car surroundings. The perception component of the stack extracts information for such raw data. For example, information about surrounding objects (e.g., other vehicles, pedestrians, signs, ... etc.) and their relative distance from the ego vehicle. The perception component can accurately enclose such objects (a.k.a. road agents) in 2D or even 3D coordinates. A third component of the stack is a localization module that accurately places road agents on a high-definition map. Other components of the stack include route planning and vehicle control (e.g. drive by the wire).

AV trajectory prediction has recently gained much more focus; especially, after a few reported AV accidents in 2021. Trajectory prediction will help to increase the safety of AVs and other vehicles for travelers, pedestrians, and other road users [2]. Statistics show that in the US alone, the number of fatalities and injuries due to vehicle crashes changed from 36k and 2.74M in 2019 [3] to 38.7K and 2.28M in 2020 [4], respectively. In the US, car accidents among other unintentional injuries are ranked the third leading cause of death in 2019 [5]. Worldwide death count due to road crashes soars up to 1.35M in 2018 [6]. Reducing traffic accidents has also an economic impact as the estimated worldwide cost of traffic road accidents from 2015 to 2030 is 1.35 trillion dollars [7].

Vehicle trajectory prediction can also play a significant role in road network operations and management [8] used in Intelligent Transportation Systems (ITS). For example, trajectory prediction can estimate the possibility of a road accident (and /or traffic buildup) in the nearest future (e.g., a few seconds ahead). Trajectory predictions can be deployed on roadside units (RSU) [9] that have been recently deployed on roads in various countries. An RSU is a unit with sophisticated communication and computation capabilities. It typically collects information about road agents [10], and possibly relays such data to other destinations on the Internet. It can also apply trajectory prediction to determine control actions to optimize traffic. Control actions typically involve giving warnings and recommendations to road agents; including, for example, varying the maximum travel speed along a road or across lanes of the road. This system is a discrete feedback control system in which the computational speed is a crucial factor in determining system stability & efficiency.

For such a real-time application, both the accuracy and the computational time of the prediction algorithm are crucial. Extensive deep-learning neural networks have shown promising results in trajectory predictions. With extensive datasets, the neural network is trained to identify patterns of movement of road agents; it relates movement information of a particular time step (current) to those of previous time steps (historical). Once the neural network reaches an acceptable accuracy, the trained network would have encoded in its architecture enough information to be able to use real-time information to estimate the trajectory of the surrounding road agents in the nearest future.

In the past decade, datasets related to the trajectory prediction problem were collected and published. These datasets were collected by AVs driven on the road for thousands of hours, by drones, and by fixed cameras and sensors deployed on the side of the road. In this paper, we use LevelxData datasets collected by drones that hover over a road segment and capture videos that monitor traffic flow. The videos are processed to generate the dataset. LevelxData includes various datasets (HighD [11], InD [12], RounD [13], and ExiD [14]) collected for various traffic scenarios. For example, HighD captures free-flowing road/highway traffic while the traffic flows at intersections are covered by InD. Turnaround traffic scenarios are captured in RounD and on- and off-ramping scenarios are considered in

2767-6684/23/$31.00 ©2023 IEEE
DOI 10.1109/DSA59317.2023.00089

the exiD dataset.

In this paper, we create and propose a deep neural network model based on Graph Convolution Networks (GNN) that can predict the trajectory of road agents in a fully-aware context. The proposed model predicts the trajectory of all surrounding vehicles at the same time. The model can be distributed and executed in parallel between the road agents. The model overcomes various issues of generic Deep Neural Network models such as over-smoothing [15]. The model is trained using the RounD dataset [13]. The RounD dataset is collected over different 24 turnaround locations in Germany. This dataset contains many driving patterns that can be learned by a deep neural network. Due to the complexity of entering, driving through, and exiting a turnaround, prediction trajectory can be used to intelligently control the turnaround and evaluate its safety. Fig. 1 shows how a drone is used to collect the data from a turnaround. The dataset contains information about 6.6 driving hours sampled with a rate of 25 samples per second. It also contains almost 28k different road agents distributed among seven different classes; bicycle, bus, car, motorcycle, trailer, truck, and van. Also, an image of the captured scene is provided with an accuracy of about a pixel per 10 cm.

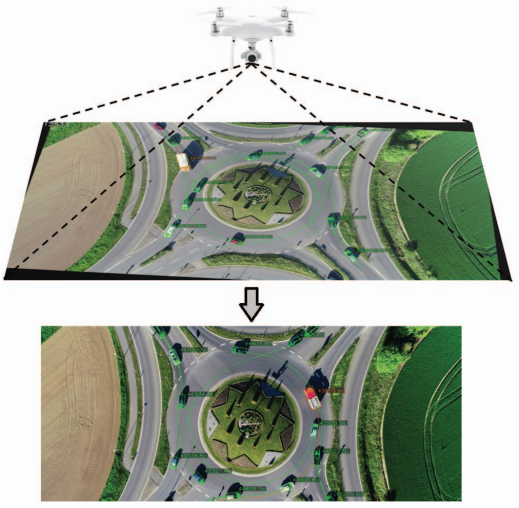

Figure 1: Turnaround scene Captured by a drone for RounD dataset [13]

2. Related work

Trajectory prediction methods in the literature can be categorized into three main classes based on how much vehicular context is considered. The context takes into consideration the surroundings around the car.

2.0..1. Context-free methods

that use only the history of the ego vehicle to learn the driving pattern of the driver. Traditional machine learning models, such as the Monte Carlo method [16], Bayesian networks [17], Hidden Markov Models (HMM) [18], and Conditional Random Fields [19], were applied to deal with the problem. Also, different types of Recurrent Neural Networks (RNN), for example, Long Short-Term Memory (LSTM) with manual feature engineering [20], [21] as well as an RNN combined with conditional variational auto-encoders (CVAE) [22] were used to solve this problem. These methods are basically designed for time-sequence prediction which is a different, but related, problem. These methods are not the best choice for the trajectory prediction problems because a driving action doesn't depend only on the driver's history but also on the driving context. For example, drivers tend to slow down in crowded scenes. These methods would simply perform poorly in the prediction performance.

2.0..2. Context-aware methods

with these methods, the historical trajectory data of both the ego vehicle and surrounding vehicles are considered. Thus, the interaction between vehicles on the road can be encoded in ML models. These methods are then considered spatial-temporal methods. However, not all context information is considered with these methods, for example, map data is not considered in these ML models. In general, it can be divided into two categories:

Graph-based models where a scene (or a frame) at a given time step is represented as a graph in which nodes represent the road agents and the interaction between them is represented by edges. An edge between two agents exists when the distance between them is less than a certain threshold. Vehicular history is captured through several frames and is encoded as a list of graphs. Graph Machine Learning techniques, such as Graph Neural Networks (GNN) and Graph Transformer [23], [24], are applied to learn and estimate the trajectory. Another technique is to use a CVAE-based model to process both the edges and the nodes of the graph [25].

On the other hand, *matrix-based models* use matrix notations to capture road agents' features according to their locations. Then, machine learning models are then applied to learn and predict behavior. For example, Meng et al. [26] project the output of LSTM into a 2D grid. While spatial attention mechanism [27] and social Pooling mechanisms [28], [29] were utilized in processing such grids, Zhao et al. [30] use Graph Convolution Networks (GCN) in parallel to increase the prediction accuracy. By stacking 2D matrices at different time steps, a 3D matrix is generated to capture historical information. A GRU-based encoder-decoder [31] or an encoder-decoder model combined with a CNN [32] are used to process such 3D matrix. However, stacking features into a matrix opens the door for new questions related to the order of stacking and the fixed dimension of the matrix. Others have tried using a CNN followed by LSTM [33] to process the matrix information and the temporal information of the 3D matrix.

Because of the lack of map information, the prediction quality of context-aware models produces acceptable results in only simple and fixed scenes; for example, along a highway with straight lanes. In addition, these models typically lack a standard architecture for spatial-temporal modeling of data and

hence they typically use domain-specific datasets. This leaves various models for various domains.

2.0..3. Fully context-free methods

where map information and historical data of the ego and surrounding vehicles are included in the model. A typical input to such models is a list of images that consists of a) map images that describe the road, lanes, traffic light, and other stationary information, b) a set of images containing the ego agent locations, c) a set of images containing the surrounding road agents. All images except the map image are binary images. The images are rotated, translated, and cropped such that the ego agent will be at a certain location and orientation at the current time step ($t = 0$). Models that accept this type of input are called Bird's Eye View (BEV) models as they are looking at the scene from high above similar to a bird. For example, the Lyft level 5 dataset [1] enabled the proposal of many such models. All these models used the semantic map generators provided by the Lyft toolkit. Fig. 2 shows an example of converting a scene in the RounD dataset, that we study in this paper, into a BEV set of images for 15 history frames.

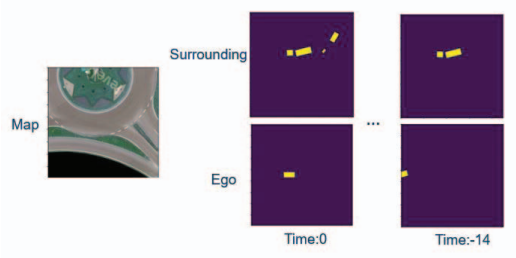

Figure 2: The input to a BEV

To take advantage of the benefits of the recently well-designed CNN architectures, CNN-based models are usually used to process BEV semantic images to estimate the trajectory of the ego agent. Because these models use semantic images, the map information can be input into the model to construct a fully context-aware model in a fixed dimension format. Although this enables processing complex and general scenes, the dimension of the set of images fed as input to the models is much higher than the dimension of the actual input data. This results in relatively higher computational power and time requirements for these models. Examples include using ResNet [34], EfficientNet [35], and convolutional-based CVAE [36] as a backbone of models. These convolutional layers are typically followed by an LSTM-based encoder-decoder network [37] or a transformer-based encoder followed by an RNN-based decoder [38].

It's worthy to note that there is a loss of information due to converting numerical data into semantic images (discretizing locations into pixels and encoding information into the color of the pixel). This limits the accuracy of such prediction models. To remedy this loss of information, models have been proposed to process the original numerical data in parallel to the CNN processing. The output of these models concatenates the output of both types of processing. LSTM encoders [34], CVAE [36], [39], [40], and LSTM Encoder-Decoder [41], [42] were used in the numerical branch of the models.

3. Problem Statement

Let A_t to be the set of all road agents, i.e., vehicles and pedestrian, that belongs to a frame sampled at time t. Also, let $s_t^{(i)} \in \mathbb{R}^d$ to be the feature vector of the ith vehicle ($i \in A_t$) at time t which includes but is not limited to the location of the agent, $\boldsymbol{x}_t^{(i)} = (x_t^{(i)}, y_t^{(i)})$. Given the history of the ego vehicle ($s_t^{(i)}$) and the surrounding road agents ($\boldsymbol{s}_t$) for t_h periods (frames) including the current state at t_0, it's required to estimate the vehicle position for a horizon of t_f. This can be expressed by a function $\mathcal{F}$ such that

$$\begin{aligned}(\hat{\boldsymbol{x}}_{t_0+t_f}^{(i)}, \dots, \hat{\boldsymbol{x}}_{t_0+1}^{(i)}) = \\ \mathcal{F}(s_{t_0}^{(i)}, \dots, s_{t_0-t_{h-1}}^{(i)} | \boldsymbol{s}_{t_0}, \dots, \boldsymbol{s}_{t_0-t_{h-1}}, \eta)\end{aligned} \tag{1}$$

where η represents the driving scene information and $\hat{\boldsymbol{x}}_t^{(i)}$ is the estimation value of $\boldsymbol{x}_t^{(i)}$. The goal is to find the function $\mathcal{F}$ that minimizes the Square Error (SE) of the predicted values as given by (2).

$$\begin{aligned}argmin_{\mathcal{F}} \sum_{i,t} \| \mathcal{F}(s_{t_0}^{(i)}, \dots, s_{t_0-t_{h-1}}^{(i)} | \boldsymbol{s}_{t_0}, \dots, \boldsymbol{s}_{t_0-t_{h-1}}, \eta) \\ - (\boldsymbol{x}_{t_0+t_f}^{(i)}, \dots, \boldsymbol{x}_{t_0+1}^{(i)}) \|^2\end{aligned} \tag{2}$$

4. Methodology

To estimate the trajectory of road agents, we propose an ML model based on the Graph convolution layers that process a heterogeneous graph generated from scenes of the RounD dataset. In this section, we discuss graph generation and ML model construction.

4.1. Graph Generation

We propose to generate and use a directed heterogeneous graph consisting of a single type of nodes and two types of edges. The nodes represent road agents. The edges are spatial and temporal edges that encode the relation between different road agents within the frame and between consequent frames. To generate a graph, we propose a general technique that can be easily applied to any dataset and doesn't require any special requirements. We use the proposed techniques with the RounD dataset as an example. The technique follows the steps shown in Fig. 3, as follows:

1) At each frame (time step), road agents are represented as the nodes of a sub-graph.
2) Spatial edges are created to connect nodes at the same time step (frame) if the distance between the agents is less than 30 m. Spatial edges are bidirectional; this implies that if there is a spatial edge from node i to node j, there must be another edge from node j directed towards node i.
3) Temporal edges are generated to connect the same vehicle in two consequent frames. They are always directed from the frame at time $(t-1)$ to that at time t.

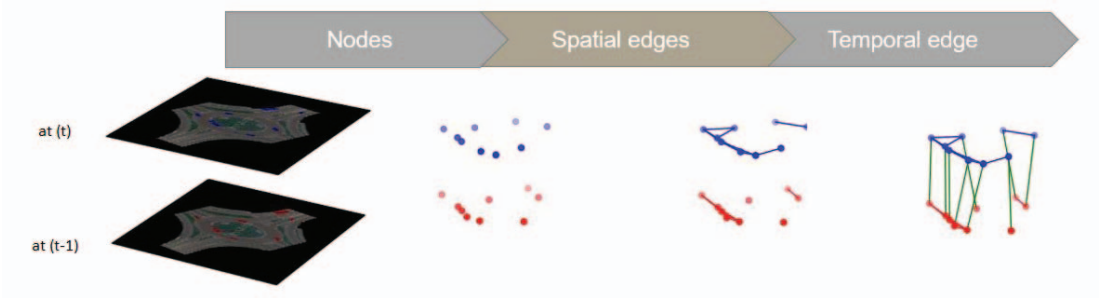

Figure 3: Steps to create the heterogeneous graph where red nodes and edges are agents and their spatial interaction at time $(t-1)$, blue nodes and edges are agents and their spatial interaction at time t, and the green edges represent the temporal relation between agents.

4) A feature vector is assigned to each node, as shown in Fig. 4. The vector consists of numerical data as well as a small image representing the relative map and road information visible to the node. The map information is transformed by translation, rotation, and cropping such that the interesting agent (ego) is always at a fixed location and direction in the image. The numeric data can be divided into two categories:
   a) Agent type which is represented as a one-hot encoding to model the seven available types of road agents. The encoding uses six Boolean numbers which are sufficient to represent the classes without having any linear dependencies between them.
   b) Agent dimension which consists of the agent centroid, direction, and dimensions.

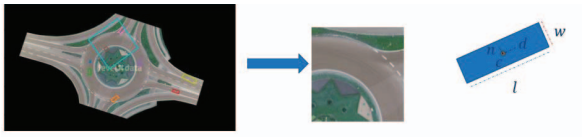

Figure 4: Node features

4.2. The proposed model

We propose a Deep Graph Neural network (DGNN) built with skip connections and attention modules to overcome the over-smoothing problem of DGNN. Data smoothing makes it harder to distinguish between different computational graphs at deeper layers of the model. Thus, the model would fail to process the graph and predict the trajectory. Skip connection is usually used by CCN [43] to increase the speed of training. Attention is implemented by Graph Attention Network (GAT) layers [44] that will intelligently update the embedding of nodes depending not only on the graph structure but also on the data. Both techniques will result in totally different processing of information (passed from one layer to the other) and totally different graph embedding even for the same computational graph structure. The proposed model is shown in Fig. 5.
For every node in the generated graph, an initial embedding $(v^{(0)})$ is calculated by applying a ResNet-18 to the map image. ResNet-18 is used because it's a simple, light, and fast version. As the image has been rotated before, the extracted features from the ResNet layer have to be reoriented and translated in world coordinates as all other numerals. Inspired by a traditional 2D rotation matrix, the features are multiplied by $cos(\theta)$ and $sin(\theta)$ where θ is the heading angle. Then, MLP is applied to provide a sort of linear combination. The output of the MLP is concatenated to the numerical part of the feature vector to produce the initial embedding $(v^{(0)})$ for this graph node, which will be forwarded to the second layer where information will pass through the temporal and spatial edges. A GAT layer followed by a ReLU is used for the spatial edges in order to give a higher score to (and consider more the) surrounding agents because they affect trajectory prediction. The normalized score is used to generate a weighted sum that updates node embedding. This operation is independent of the number of edges (surrounding agents). This will allow the model to generally process any scene with any number of road agents. As, there are two temporal edges per node; the self-loop, and the vehicle at the previous time step, a simple Graph Convolution Network (GCN) layer [45] is capable of propagating the temporal information through the graph and it will be coupled with a ReLU. Then, the spatial and temporal embedding, as well as the skip connection of the numerical features, are concatenated together to form the $(v^{(1)})$ embedding of the second layer. Fig. 5 shows only a single node but this will be repeated in parallel to other nodes in the graph.

At this point, the embedding of all the nodes of the graph at the second layer will be calculated using only the information propagated from nodes one hop away. Then, a similar layer will be applied to produce the embedding $(v^{(2)})$. Similarly, this embedding will have information propagated from nodes two hops away including the spatial and temporal information at the current time steps, as well as the previous two time step. By using $(h-1)$ layers, the information of h time steps in both spatial and temporal edges is propagated into the final embedding $(v^{(h-1)})$. Finally, an MLP will be applied to the final embedding to produce the final prediction which will be $2f$ numerical values; f values for the x-coordinates and f values for the y-coordinates. To prevent over-fitting,

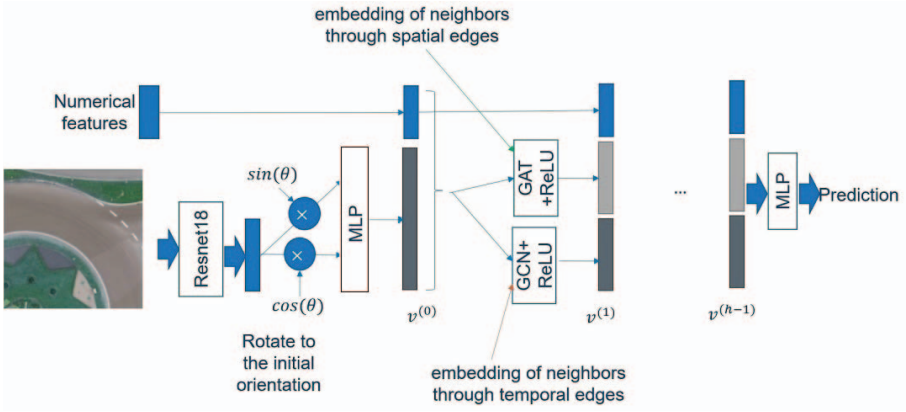

Figure 5: The proposed model

dropout with probability of 0.2 is applied to all GCN and GAT modules.

As the embedding of each layer depends only on the embedding of the previous layers, all road agents can update their embedding collectively. The embedding of the previous layer can be exchanged by some sort of communication between road agents. This is different than BEV models in which each prediction is done individually with the need to iterate among all road agents in the scene with no possibility of balancing the prediction load between road agents.

5. Results and Discussion

The proposed model is applied to the RounD dataset that provides a raw dataset with high non-linearity and uncertainty of the driving actions at turnaround scenarios. First, the graph is generated for each scene of the 24 available scenes. As the original frame rate is 25 frames per second, down-sampling is used to produce a sampling rate of 5 frames per second. The down-sampled data is used to generate the directed graph. Taking the third scene as an example, the generated graph consists of about 55k nodes, 204k spatial edges, and 54k temporal edges. Almost 28K of the nodes have a complete history and future data. Thus, they are considered interesting cases and used in the training and testing processes. The interesting cases are sampled such that 70% of the cases are used for training while the remaining form the testing set. The training set is used to tune the trainable parameters of the model to optimize equation (2) while the testing set is used to determine the best-trained parameters that can be in general without any bias or over-fitting.

The Displacement Error DE is given by equation (3) and is used to evaluate the prediction accuracy for a given interesting case. It calculates the Euclidean distance between the ground truth and the predicted location for vehicle i after a time of ΔT from the current time t_0.

$$DE(i, \Delta T) = \|\boldsymbol{x}^{(i)}_{t_0+\Delta T} - \hat{\boldsymbol{x}}^{(i)}_{t_0+\Delta T}\|_2 \tag{3}$$

Many evaluation metrics are usually used to evaluate the prediction accuracy for the whole dataset. In this paper, Average

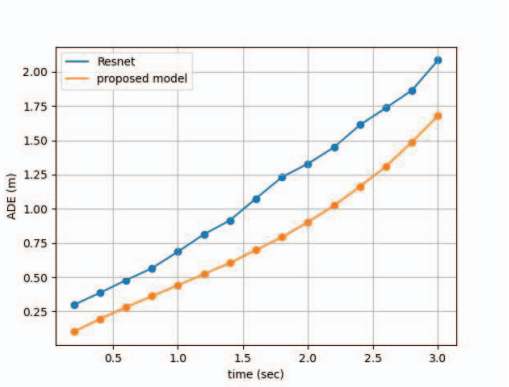

Figure 6: Trajectory prediction results

Displacement Error (ADE), given by equation (4), and Final Displacement Error, given by equation (5), are used to evaluate the performance of the proposed model with a traditional fully context-aware model that uses ResNet-18 as a BEV model to process the semantic maps to predict the vehicle trajectory.

$$ADE(\Delta T) = \sum_i DE(i, \Delta T) \tag{4}$$

$$FDE = ADE(t_f) \tag{5}$$

For both models, the history period (t_h) and the future horizon (t_f) are 3 seconds (15 time-steps). Both models are trained using the same training set and the same stop criterion based on the same testing set is used to control the training loop. The prediction of all the cases is evaluated using ADE and the results are shown in Fig. 6. As shown in the results the ADE of the proposed model has less error than the BEV model for all time steps. Although, the error for both of them increases as the prediction time increases, FDE shown as the final point in the figure is also lower in the case of our model. Table I shows the average value of FDE in meters for each agent type. As the values are close, This illustrates that the model is able to distinguish the different types of traffic agents.

Type	FDE (m)
bicycle	1.742
bus	0.668
car	1.729
motorcycle	1.581
trailer	1.573
truck	1.510
van	1.484

TABLE I: Average FDE for different agent types

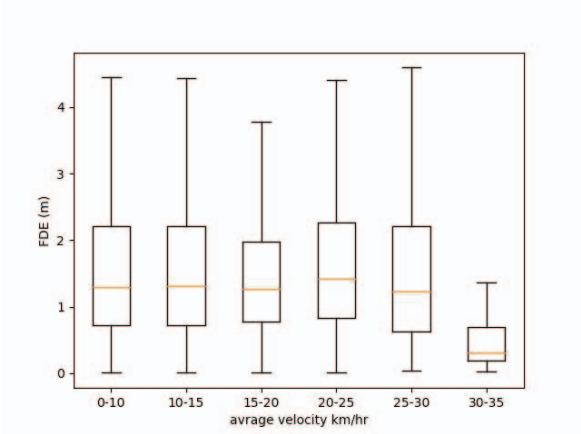

Figure 7: Trajectory prediction results

Also, the model can understand the different dimensions of different road agents. The average FDS value of the trucks, vans, trailers and motorcycles is less than those of others types. A possible reason can be because of they usually are relative slower than other types. Buses also has a very small FDE but this can be due to over-fitting as buses is rarely represented among the dataset ($< 0.1\%$).

Figure 7 is used to prove that the model works well even for aggressive driver. As the type of driving is not included in the dataset and in order not to add our personal prospective into the data to avoid any sort of bias, the FDE of the prediction is divided into separate sets according to the average speed of the agents. A boxplot is drawn to show the first quartile, median, and the third quartile of each set. As shown in the figure, the three values are close in every set while the maximum FDE is small for agents with high speed.

6. Conclusion

Trajectory prediction of road agents plays an important role in maintaining road traffic safety. It's an essential component in the AV stack as well as in real-time road traffic control. The prediction should be fast, simple, accurate, and fully context-aware. Related works use a very high dimensional input as a set of semantic maps in a traditional BEV to predict the trajectory of a single vehicle. In this paper, a model is proposed that can predict the trajectory of all road agents in a turnaround-driving scene. The proposed model is applied to a heterogeneous graph that encodes both the temporal and spatial information in the same structure. It overcomes the over-smoothing problem due to the large number of hops in the graph. The model was able to estimate the trajectory of all the vehicles in the frame at the same time with an error of fewer than 1.7 meters for a horizon of 3 seconds.

An algorithm is also proposed to generate the graph in which each vehicle is represented as a node and the map information locally viewed by the vehicle is placed as a feature for the node. Two types of directed edges are used to represent the temporal and spatial relation between nodes. The model is designed based on Deep Graph Neural Networks.

Although the number of images per node processed by the model is reduced to one image instead of $(2h+1)$ images used by BEV models, It's still needed in future work to propose another numeric presentation of the road map to reduce the model complexity. Future work can also include parameter sharing between layers to reduce the model size and training time.

References

[1] J. Houston, G. Zuidhof, L. Bergamini, Y. Ye, L. Chen, A. Jain, S. Omari, V. Iglovikov, and P. Ondruska, "One thousand and one hours: Self-driving motion prediction dataset," 2020. [Online]. Available: https://arxiv.org/abs/2006.14480

[2] A. R. Alozi and M. Hussein, "Evaluating the safety of autonomous vehicle–pedestrian interactions: An extreme value theory approach," *Analytic Methods in Accident Research*, vol. 35, p. 100230, Sep. 2022. [Online]. Available: https://doi.org/10.1016/j.amar.2022.100230

[3] "Traffic safety facts 2019: A compilation of motor vehicle crash data," National Center for Statistics and Analysis, Washington, DC, Tech. Rep., 2021. [Online]. Available: https://crashstats.nhtsa.dot.gov /api/public/viewpublication/813141

[4] T. Stewart, "Overview of motor vehicle crashes in 2020," National Highway Traffic Safety Administration, US, Tech. Rep., 2022. [Online]. Available: https://crashstats.nhtsa.dot.gov/Api/Public /Publication/813266

[5] J. X. E. A. S.L. Murphy, K. D. Kochanek, "Mortality in the united states, 2020 key findings data from the national vital statistics system." U.S. Department of Health and Human Services, US, Tech. Rep., 2021.

[6] "Global status report on road safety 2018," World Health Organization, Tech. Rep., 2018. [Online]. Available: https://www.who.int/publications/i/item/9789241565684

[7] S. Chen, M. Kuhn, K. Prettner, and D. E. Bloom, "The global macroeconomic burden of road injuries: estimates and projections for 166 countries," *The Lancet Planetary Health*, vol. 3, no. 9, pp. e390–e398, Sep. 2019. [Online]. Available: https://doi.org/10.1016/s2542-5196(19)30170-6

[8] S. S. Musa, M. Zennaro, M. Libsie, and E. Pietrosemoli, "Convergence of information-centric networks and edge intelligence for IoV: Challenges and future directions," *Future Internet*, vol. 14, no. 7, p. 192, Jun. 2022. [Online]. Available: https://doi.org/10.3390/fi14070192

[9] L. Jiang, T. G. Molnár, and G. Orosz, "On the deployment of v2x roadside units for traffic prediction," *Transportation Research Part C: Emerging Technologies*, vol. 129, p. 103238, 2021. [Online]. Available: https://www.sciencedirect.com/science /article/pii/S0968090X21002515

[10] M. N. Tahir, P. Leviäkangas, and M. Katz, "Connected vehicles: V2v and v2i road weather and traffic communication using cellular technologies," *Sensors*, vol. 22, no. 3, p. 1142, Feb. 2022. [Online]. Available: https://doi.org/10.3390/s22031142

[11] R. Krajewski, J. Bock, L. Kloeker, and L. Eckstein, "The highd dataset: A drone dataset of naturalistic vehicle trajectories on german highways for validation of highly automated driving systems," 2018. [Online]. Available: https://arxiv.org/abs/1810.05642

[12] J. Bock, R. Krajewski, T. Moers, S. Runde, L. Vater, and L. Eckstein, "The inD dataset: A drone dataset of naturalistic road user trajectories at german intersections," in *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, Oct. 2020. [Online]. Available: https://doi.org/10.1109/iv47402.2020.9304839

[13] R. Krajewski, T. Moers, J. Bock, L. Vater, and L. Eckstein, "The rounD dataset: A drone dataset of road user trajectories at roundabouts in germany," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, Sep. 2020. [Online]. Available: https://doi.org/10.1109/itsc45102.2020.9294728

[14] T. Moers, L. Vater, R. Krajewski, J. Bock, A. Zlocki, and L. Eckstein, "The exiD dataset: A real-world trajectory dataset of highly interactive highway scenarios in germany," in *2022 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, Jun. 2022. [Online]. Available: https://doi.org/10.1109/iv51971.2022.9827305

[15] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," 2018. [Online]. Available: https://arxiv.org/abs/1812.08434

[16] S. Danielsson, L. Petersson, and A. Eidehall, "Monte carlo based threat assessment: Analysis and improvements," in *2007 IEEE Intelligent Vehicles Symposium*. IEEE, Jun. 2007. [Online]. Available: https://doi.org/10.1109/ivs.2007.4290120

[17] S. Lefevre, C. Laugier, and J. Ibanez-Guzman, "Exploiting map information for driver intention estimation at road intersections," in *2011 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, Jun. 2011. [Online]. Available: https://doi.org/10.1109/ivs.2011.5940452

[18] H. Berndt and K. Dietmayer, "Driver intention inference with vehicle onboard sensors," in *2009 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*. IEEE, 2009. [Online]. Available: https://doi.org/10.1109/icves.2009.5400203

[19] E. Ohn-Bar, A. Tawari, S. Martin, and M. M. Trivedi, "On surveillance for safety critical events: In-vehicle video networks for predictive driver assistance systems," *Computer Vision and Image Understanding*, vol. 134, pp. 130–140, May 2015. [Online]. Available: https://doi.org/10.1016/j.cviu.2014.10.003

[20] A. Khosroshahi, E. Ohn-Bar, and M. M. Trivedi, "Surround vehicles trajectory analysis with recurrent neural networks," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, Nov. 2016. [Online]. Available: https://doi.org/10.1109/itsc.2016.7795922

[21] D. J. Phillips, T. A. Wheeler, and M. J. Kochenderfer, "Generalizable intention prediction of human drivers at intersections," in *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, Jun. 2017. [Online]. Available: https://doi.org/10.1109/ivs.2017.7995948

[22] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. S. Torr, and M. Chandraker, "Desire: Distant future prediction in dynamic scenes with interacting agents," 2017. [Online]. Available: https://arxiv.org/abs/1704.04394

[23] D. Singh and R. Srivastava, "Multi-scale graph-transformer network for trajectory prediction of the autonomous vehicles," *Intelligent Service Robotics*, vol. 15, no. 3, pp. 307–320, May 2022. [Online]. Available: https://doi.org/10.1007/s11370-022-00422-w

[24] L. Li, X. Sui, J. Lian, F. Yu, and Y. Zhou, "Vehicle interaction behavior prediction with self-attention," *Sensors*, vol. 22, no. 2, p. 429, Jan. 2022. [Online]. Available: https://doi.org/10.3390/s22020429

[25] B. Ivanovic, K.-H. Lee, P. Tokmakov, B. Wulfe, R. McAllister, A. Gaidon, and M. Pavone, "Heterogeneous-agent trajectory forecasting incorporating class uncertainty," 2021. [Online]. Available: https://arxiv.org/abs/2104.12446

[26] Q. Meng, B. Shang, Y. Liu, H. Guo, and X. Zhao, "Intelligent vehicles trajectory prediction with spatial and temporal attention mechanism," *IFAC-PapersOnLine*, vol. 54, no. 10, pp. 454–459, 2021. [Online]. Available: https://doi.org/10.1016/j.ifacol.2021.10.204

[27] J. Yan, Z. Peng, H. Yin, J. Wang, X. Wang, Y. Shen, W. Stechele, and D. Cremers, "Trajectory prediction for intelligent vehicles using spatial-attention mechanism," *IET Intelligent Transport Systems*, vol. 14, no. 13, pp. 1855–1863, Dec. 2020. [Online]. Available: https://doi.org/10.1049/iet-its.2020.0274

[28] N. Deo and M. M. Trivedi, "Convolutional social pooling for vehicle trajectory prediction," 2018. [Online]. Available: https://arxiv.org/abs/1805.06771

[29] X. Xu, W. Liu, and L. Yu, "Trajectory prediction for heterogeneous traffic-agents using knowledge correction data-driven model," *Information Sciences*,

vol. 608, pp. 375–391, Aug. 2022. [Online]. Available: https://doi.org/10.1016/j.ins.2022.06.073

[30] Z. Zhao, H. Fang, Z. Jin, and Q. Qiu, "GISNet:graph-based information sharing network for vehicle trajectory prediction," in *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, Jul. 2020. [Online]. Available: https://doi.org/10.1109/ijcnn48605.2020.9206770

[31] J. An, W. Liu, Q. Liu, L. Guo, P. Ren, and T. Li, "DGInet: Dynamic graph and interaction-aware convolutional network for vehicle trajectory prediction," *Neural Networks*, vol. 151, pp. 336–348, Jul. 2022. [Online]. Available: https://doi.org/10.1016/j.neunet.2022.03.038

[32] Z. Sheng, Y. Xu, S. Xue, and D. Li, "Graph-based spatial-temporal convolutional network for vehicle trajectory prediction in autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 10, pp. 17 654–17 665, Oct. 2022. [Online]. Available: https://doi.org/10.1109/tits.2022.3155749

[33] C. Ju, Z. Wang, C. Long, X. Zhang, and D. E. Chang, "Interaction-aware kalman neural networks for trajectory prediction," in *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, Oct. 2020. [Online]. Available: https://doi.org/10.1109/iv47402.2020.9304764

[34] A. Trabelsi, R. J. Beveridge, and N. Blanchard, "Motion prediction performance analysis for autonomous driving systems and the effects of tracking noise," 2021. [Online]. Available: https://arxiv.org/abs/2104.08368

[35] S. Mandal, S. Biswas, V. E. Balas, R. N. Shaw, and A. Ghosh, "Motion prediction for autonomous vehicles from lyft dataset using deep learning," in *2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA)*. IEEE, Oct. 2020. [Online]. Available: https://doi.org/10.1109/iccca49541.2020.9250790

[36] X. Huang, S. G. McGill, J. A. DeCastro, L. Fletcher, J. J. Leonard, B. C. Williams, and G. Rosman, "Diversitygan: Diversity-aware vehicle motion prediction via latent semantic sampling," 2019. [Online]. Available: https://arxiv.org/abs/1911.12736

[37] X. Huang, G. Rosman, I. Gilitschenski, A. Jasour, S. G. McGill, J. J. Leonard, and B. C. Williams, "Hyper: Learned hybrid trajectory prediction via factored inference and adaptive sampling," 2021. [Online]. Available: https://arxiv.org/abs/2110.02344

[38] M. Bhat, J. Francis, and J. Oh, "Trajformer: Trajectory prediction with local self-attentive contexts for autonomous driving," 2020. [Online]. Available: https://arxiv.org/abs/2011.14910

[39] Z. Zhong, Y. Luo, and W. Liang, "STGM: Vehicle trajectory prediction based on generative model for spatial-temporal features," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 10, pp. 18 785–18 793, Oct. 2022. [Online]. Available: https://doi.org/10.1109/tits.2022.3160648

[40] J. Kim, R. Mahjourian, S. Ettinger, M. Bansal, B. White, B. Sapp, and D. Anguelov, "Stopnet: Scalable trajectory and occupancy prediction for urban autonomous driving," 2022. [Online]. Available: https://arxiv.org/abs/2206.00991

[41] C. Kim, H.-S. Yoon, S.-W. Seo, and S.-W. Kim, "STFP: Simultaneous traffic scene forecasting and planning for autonomous driving," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, Sep. 2021. [Online]. Available: https://doi.org/10.1109/iros51168.2021.9636255

[42] H. Kim and D. H. Shim, "Vehicle trajectory prediction with convolutional neural network and sequence-to-sequence," in *Lecture Notes in Mechanical Engineering*. Springer Singapore, 2021, pp. 109–114. [Online]. Available: https://doi.org/10.1007/978-981-16-4803-8_13

[43] L. Wang, B. Yin, A. Guo, H. Ma, and J. Cao, "Skip-connection convolutional neural network for still image crowd counting," *Applied Intelligence*, vol. 48, no. 10, pp. 3360–3371, Feb. 2018. [Online]. Available: https://doi.org/10.1007/s10489-018-1150-1

[44] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," 2017. [Online]. Available: https://arxiv.org/abs/1710.10903

[45] H. Cai, V. W. Zheng, and K. C.-C. Chang, "A comprehensive survey of graph embedding: Problems, techniques and applications," 2017. [Online]. Available: https://arxiv.org/abs/1709.07604