Multi-modal Trajectory Prediction for Autonomous Driving with Semantic Map and Dynamic Graph Attention Network

Bo Dong, Hao Liu, Yu Bai, Jinbiao Lin, Zhuoran Xu, Xinyu Xu, Qi Kong
JD Logistics, JD.com, China
{dongbo5,liuhao163,baiyu9,linjinbiao,xuzhuoran,xinyu.xun,Qi.Kong}@jd.com

Abstract

Predicting future trajectories of surrounding obstacles is a crucial task for autonomous driving cars to achieve a high degree of road safety. There are several challenges in trajectory prediction in real-world traffic scenarios, including obeying traffic rules, dealing with social interactions, handling traffic of multi-class movement, and predicting multi-modal trajectories with probability. Inspired by people's natural habit of navigating traffic with attention to their goals and surroundings, this paper presents a unique dynamic graph attention network to solve all those challenges. The network is designed to model the dynamic social interactions among agents and conform to traffic rules with a semantic map. By extending the anchor-based method to multiple types of agents, the proposed method can predict multi-modal trajectories with probabilities for multi-class movements using a single model. We validate our approach on the proprietary autonomous driving dataset for the logistic delivery scenario and two publicly available datasets. The results show that our method outperforms state-of-the-art techniques and demonstrates the potential for trajectory prediction in real-world traffic.

1 Introduction

Autonomous driving is believed [6] to have a tremendous positive impact on human society. To ensure a high degree of safety even in uncertain or dynamically changing environments, an autonomous vehicle should be able to anticipate the future trajectories of the surrounding agents (*e.g.* vehicles, pedestrians, and cyclists) in advance and plan a plausible path in response to the behaviour of other agents such that the probability of collision is minimized. However, the motion trajectory of the surrounding agents is often hard to predict without explicitly knowing their intention. In this case, we need to utilize other useful information to improve safety and efficacy of the planned path of the ego-vehicle, including the observed current status of notable surrounding agents, possible physically acceptable routes in the current traffic scenario, and possible interaction outcomes with their likelihoods. Unfortunately, several challenges still exist that prevents us from utilizing this information to achieve reliable trajectory prediction. In this paper, five main challenges in trajectory prediction for autonomous driving are summarized and discussed as follows:

Considering surrounding traffic environments. In real-world traffic scenarios, the movement of traffic must obey traffic rules, and avoid surrounding obstacles in the meantime. That useful information can be found in the high definition (HD) map.

Dealing with social interactions. To avoid the collision, the trend of interacting with surrounding traffic agents needs to be captured. However, interactions between different types of traffic are very different, *e.g.* the interaction between pedestrians is different from the interaction between a car and a pedestrian.

Machine Learning for Autonomous Driving Workshop at the 34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

Table 1: Comparison of challenges handled in different methods in trajectory prediction.

Methods	Traffic Environments	Social	Multi-class	Multi-modal	Probability
Social LSTM [1]		✓			
Social GAN [7]		✓		✓	
PECNet [12]		✓		✓	
Argoverse [4]	✓				
Trajectron++ [21]		✓	✓	✓	
Multipath [2]	✓			✓	✓
DGAN (ours)	✓	✓	✓	✓	✓

Handling traffic of multi-class movement. The movement patterns of different types of traffic need to be considered for autonomous driving, including cars, buses, trucks, motorcycles, bicycles, and pedestrians. In this paper, those types of traffic are divided into three categories, namely vehicles (cars, buses, and trucks), cyclists (motorcycles and bicycles) and pedestrians.

Predicting multi-modal trajectories with probability. In reality, people may follow several plausible ways when navigating crowd and traffic. To avoid potential collisions, the most probable future movements should be considered.

Probability awareness. The probability value of each possible path of surrounding obstacles is a considerable factor in the planning and control of the autonomous driving car.

State-of-the-art methods only solve some, but not all, challenges at once as shown in Table 1. In this paper, we present a multi-modal trajectory prediction method to tackle all these challenges, which models the dynamic social interactions among agents using Graph Attention Network (GAT) [23] and semantic map. The contributions of our proposed method are summarized as follows:

- The proposed method is designed to achieve multi-modal predictions with considering traffic environments, dealing with social interactions, and predicting multi-class movement patterns with probability values, simultaneously.
- In the proposed Dynamic Graph Attention Network (DGAN), Dynamic Attention Zone and GAT are combined to model the intention and habit of human driving in heterogeneous traffic scenarios.
- To capture complex social interactions among road agents, we combine different types of information, including a semantic HD map, observed trajectories of road agents, and the current status of the traffic.

2 Related Work

Here, we review recent literature on trajectory prediction with social interactions.

RNN-related methods. The recurrent neural network (RNN) [13] and long short term memory (LSTM) [8] have proven to be very effective in time-related prediction tasks. To capture social interactions between pedestrians in crowds, Alexandre *et al*. [1] used a social pooling layer in LSTMs to capture social interactions based on the relative distance between different pedestrians. Chandra *et al*. [3] introduced an LSTM-CNN hybrid method with the weighted horizon and local relative interactions in heterogeneous traffic. However, those previous studies only focus on predicting future trajectories for one class, *e.g.* pedestrians or vehicles.

GAN-related methods. As there are multiple plausible paths that people could take in the future, several methods [7, 9, 14] were proposed using the GAN framework to generate multiple trajectories for a given input. However, to generate multiple results for one target in practice, the generative model should be executed repeatedly with a latent vector randomly sampled from $\mathcal{N}(0, 1)$ as input. Randomly initialised inputs will generate random outcomes, which may lead to large margins between the generated results and the ground truth. To cover the most likely future paths, the number of executions has to be increased.

Methods that encode traffic rules. To predict trajectories that obey traffic rules, several methods used features learned from customised semantic HD map or static-scene images to encode prior knowledge on traffic rules. Chai *et al.* [2] proposed a multipath model to predict parametric distributions of future trajectories with HD map. It regresses offsets for each predefined anchor and

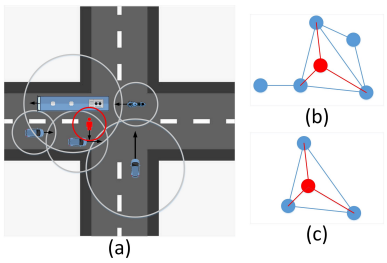

Figure 1: Dynamic attention zone and graph modelling for simulating the interaction pattern in real world traffic scenario.

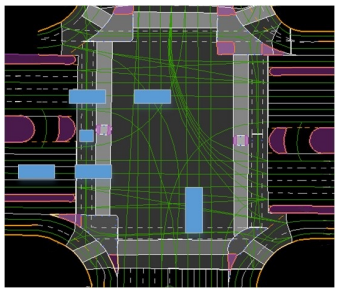

Figure 2: RGB image representation of semantic HD map for encoding the real world traffic environments.

predicts a Gaussian Mixture Model (GMM) at each time step. Meanwhile, with a birds-eye-view (BEV) binary image, probabilities are predicted over the fixed set of K predefined anchor trajectories. Cui *et al*. introduced a multi-modal architecture using a raster image from an HD map with each agent's surrounding content encoded. In [4], lane sequences were extracted from rich maps as reference lines to predict cars' trajectories. Sadeghian *et al.* [19] presented a GAN framework integrating features encoded from the static-camera frames as the traffic rule constraints using the attention mechanism. However, those works only encode car lanes without considering pedestrian crossings, cycle lanes and other static obstacles labeled in the HD map at the same time.

3 Methodology

3.1 Problem Definition

Given a set of N agents in a scenario with their corresponding observed information over a time period T_{ob} from time steps $1, ..., t_{ob}$, our goal is to predict the future trajectories $\hat{\mathbf{Y}} = \{\hat{Y_1}, ..., \hat{Y_N}\}$ of all agents involved in the scenario over a time period T_f from time step $t_{ob}+1, ..., t_f$. N agents belong to multiple c classes, *e.g.* vehicle, cyclist, and pedestrian. Similarly, the ground truth of the future trajectory is defined as $\mathbf{Y} = \{Y_1, ..., Y_N\}$, where $Y_i = \{p_i^t = (x_i^t, y_i^t) | t \in \{t_{ob}+1, ..., t_f\}$, and $i \in \{1, ..., N\}$. There are three different kinds of observed information as inputs to our model, including the semantic map $map^{t_{ob}}$ of the current scenario at time stamp t_{ob}, the traffic state $S_i^{t_{ob}}$ of agent i at current time stamp t_{ob}, and the observed trajectories of all agents $\mathbf{X} = \{X_1, ..., X_N\}$, where $X_i = \{p_i^t = (x_i^t, y_i^t) | t \in \{1, ..., t_{ob}\}\}$.

3.2 Dynamic Graph Attention Network

3.2.1 Dynamic Attention Zone and Graph Modelling

Inspired by the real-world traffic moving pattern, a dynamic attention zone is designed to capture the normal ability of people when interacting with others in traffic. Human beings have the natural sense to choose which surrounding moving agents should be noticed by judging their current status, such as distances, headings, velocities, and sizes. Then, we model each object in the scenario to have an attention circle. Based on the intersection status of the attention circles, we can easily select surrounding agents to have social interactions with. The radius r of the circle is defined as follows:

$$r_i^t = velocity_i^t * T_f + \lambda * length_i, \tag{1}$$

where T_f represents the period of future time for prediction, and λ is a constant value. The $velocity_i^t$ and $length_i$ represent the speed at time t and length of object i, respectively. The attention zone at time t covers all potential future positions over a time period T_f based on the observed speed at the current time step and the length of the agent. If the agent accelerates or decelerates, the region of attention zone will be enlarged or reduced accordingly to predict the future movement for the next time step.

As illustrated in Figure 1.(a), based on the current position and radius of each agent, attention zones of all agents are firstly drawn. Then, the graph of the current scenario at time step t is generated based on the intersection relations of every attention zone.

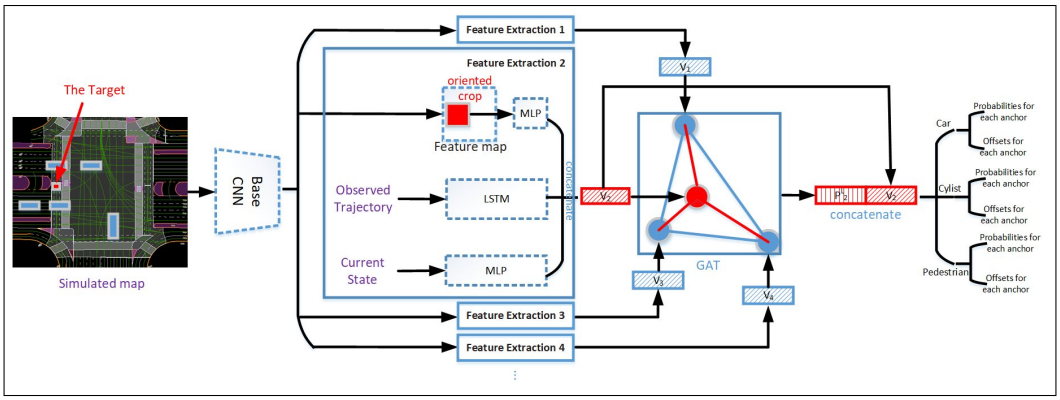

Figure 3: Dynamic Graph Attention Network.

We define G as (V, E), in which $V = \{v_i | i \in \{1, .., N\}\}$ and $E = \{e_{ij} | \forall i, j \in \{1, .., N\}\}$, where V and E denotes the vertexes and edges of the graph G. As shown in Figure 1.(b), the graph represents the relations in the whole scenario, but in Figure 1.(c), we only focus on the partial graph related to the target in red color. The value of e_{ij} will be calculated and updated in the GAT model in section 3.2.3. Each node in V denotes feature embeddings calculated from three different sources including semantic map, observed trajectory, and traffic state.

3.2.2 Feature Extraction

To make the best use of the available information, three types of features are jointly extracted from the semantic map, observed history trajectories, and current moving status.

Semantic Map. In autonomous driving applications, semantic HD map contains valuable traffic rule information. We create an RGB image representation to encode traffic rule information contained in semantic HD map. In the RGB image representation of the semantic HD map (Figure.2), pink regions represent commonly seen un-movable road obstacles, *e.g.* median strips or barriers. Yellow lines represent road boundaries. Grey and white regions represent pedestrian crossings and bicycle lanes. The green lines are the centre lines of lanes. Blue boxes denote movable obstacles (*i.e.* it can move even though it could be stationary) in the current traffic scenario. Dotted white lines and solid white lines are the traffic lane lines and edge lines, respectively. The middle-layer output estimated by the CNN is extracted as the visual feature $V_{map}^{t_{ob}}$ to represent traffic rule information in $map^{t_{ob}}$:

$$V_{map}^{t_{ob}} = CNN(map^{t_{ob}}; W_{cnn}). \tag{2}$$

Observed Trajectory. An LSTM is used to extract joint features from the observed trajectories of all involved agents. Similar to [7], we first embed the location using a single-layer multilayer perceptron (MLP) to get a fixed-length vector e_i^t as the input of the LSTM cell:

$$\begin{aligned} e_i^t &= \phi_{ot}(X_i^t; W_{ot}), \\ V_{oti}^t &= LSTM(V_{oti}^{t-1}, e_i^t; W_{ot}), \end{aligned} \tag{3}$$

where ϕ is an embedding function with a rectified linear unit (ReLU) nonlinearity, and W_{ot} is the embedding weight. The LSTM weight (W_{ot}) is shared between all agents.

traffic state. The traffic state S is very important for capturing extra information to predict the future trajectories, where $S_i^t = (velocity_i^t, acceleration_i^t, heading_i^t, width_i, length_i, c_i)$ represent the velocity, acceleration, heading, width, length, and class of agent i, respectively. A simple MLP is used for encoding to get the embedding feature V_{ts}^t of the traffic state.

$$V_{tsi}^t = \phi_{ts}(S_i^t; W_{ts}), \tag{4}$$

where W_{ts} is the embedding weight of the MLP.

The final embedding feature is defined as $V_i^{t_{ob}}$, which concatenates the three types of embedding calculated from the semantic map, observed trajectory, and agent status at the current time step:

$$V_i^{t_{ob}} = concatenate(V_{map}^{t_{ob}}, V_{oti}^{t_{ob}}, V_{tsi}^{t_{ob}}). \tag{5}$$

3.2.3 Graph Attention Network

The attention mechanism is found to be extremely powerful to draw global dependencies between inputs and outputs [22]. In attention-related methods, the GAT [23] can naturally work with our proposed dynamic attention zone and graph modelling described in section 3.2.1. In the graph, the vertex V_i represents the embedding feature of agent i, and e_{ij} represents the relative weight between an agent i and its neighbour j according to the graph generated from the dynamic attention zone. We use multiple stacked graph attention layers, and for each layer l, W_{gat} is updated during training.

$$\begin{aligned} e_{ij} &= a(W_{gat}V_i^{t_{ob}}, W_{gat}V_j^{t_{ob}}), \\ a_{ij} &= softmax(e_{ij}), \\ P^l(i) &= \sum_{j \in N_i} a_{ij} W_{gat} V_j^{t_{ob}}, \end{aligned} \tag{6}$$

where e_{ij} indicates the importance of node $j's$ feature to node i, a is the shared attentional mechanism described in [23], and P^l is the output of the lth layer by summing the corresponding weighted feature of each j in N_i neighbours of agent i. We define P^L, the output from the last GAT layer L, as the final feature.

Finally, the final feature P^L and the original feature $V_i^{t_{ob}}$ are concatenated as the input of the final MLP layers ϕ_f to predict the future trajectories. We follow the idea of hierarchical classification [17] to calculate the probabilities belonging to class c and anchor k_c.

$$(prob(c)_i, prob(k_c|c)_i), \mu_{ik} = \phi_f(concatenate(P^L, V_i); W_{ac}, W_{or}), \tag{7}$$

where W_{ac} and W_{or} are weights of the MLPs for the two parallel headers, anchor classification and offset regression, respectively; $prob(c)_i$ and $prob(k_c|c)_i$ are the hierarchical probabilities for agent i classified into class c and anchor k_c; and μ_{ik_c} is the predicted future trajectory offset based on the k_c-th anchor for the i-th agent.

3.3 Multi-modal Trajectory Prediction

The proposed method is capable of predicting multiple possible future trajectories with corresponding probability using pre-defined anchor trajectories. In this section, we present the details of multi-modal trajectory prediction.

For the anchor and loss design, we follow the methods described in [2] and [5], respectively. First, all ground-truth future trajectories are normalized in the training dataset. Then, an unsupervised classification algorithm [2] such as the k-means or uniform sampling algorithm, depending on datasets, is applied to obtain a fixed number of anchors with squared distance $dist(Y_i, Y_j)$ between future trajectories.

$$dist(Y_i, Y_j) = \sum_{t=t_{ob}}^{t_f} ||M_i p_i^t - M_j p_j^t||_2^2, \tag{8}$$

where M_i and M_j are transform matrices which transform trajectories into the agent-centric coordinate frame with the same orientation at time step t_{obs}.

However, those unsupervised classification algorithms always generate redundant results for a heavily skewed distribution. In practice, we manually select anchors based on the normalized ground-truth trajectories. For each class c, we extract K_c anchors. In total, we have K anchors for anchor classification and corresponding offset regression.

The final loss consists of anchor classification loss and trajectory offset loss:

$$\mathcal{L}_\theta = \sum_{i=1}^{N} [\mathcal{L}_i^{class} + \alpha \sum_{c=1}^{C} \sum_{k_c=1}^{K_c} I_{k_c=k^*} L(\hat{Y}_{ik_c}, Y_i)]. \tag{9}$$

$L(\hat{Y}_{ik}, Y_i)$ represents the single-mode loss L of the ith agent's k_cth anchor, where:

$$L(\hat{Y}_{ik_c}, Y_i) = \frac{1}{T_f} \sum_{t=t_{ob}+1}^{t_f} \|a_{ik_c}^t + \mu_{ik_c}^t - M_i p_i^t\|_2, \tag{10}$$

where $a^t_{ik_c}$, $\mu^t_{ik_c}$, and p^t_i are points at each time step t of the k_cth anchor, corresponding offset based on the k_cth anchor, and Y_i, respectively.

$\mathcal{L}^{class}_i$ is the hierarchical classification loss [17]:

$$\mathcal{L}^{class}_i = -\sum_{c=1}^{C}\sum_{k_c=1}^{K_c} I_{c=c^*} I_{k_c=k^*_c} \log(prob(c)_i * prob(k|c)_i), \tag{11}$$

where I is the indicator function; c^* is the ground-truth class of the agent i; k^*_c is the index of the anchor trajectory closest to the ground-truth trajectory according to the squared distance function $dist(\hat{Y}_{ik_c}, Y_i)$:

$$k^*_c = \underset{k_c \in \{1,\ldots,K_c\}}{\arg\min}\ dist(\hat{Y}_{ik_c}, Y_i). \tag{12}$$

4 Experiments

In this section, we evaluate the proposed methods on three datasets, including our internal proprietary logistic delivery dataset and two publicly available datasets, the Stanford drone dataset [18], and ETC-UCY datasets. These three datasets all include trajectories of multiple agents with social interaction scenarios and birds-eye-view RGB frames used for semantic maps. The commonly used metrics [1–3, 7], including Average Displacement Error (ADE), Final Displacement Error (FDE), and Minimum Average Displacement Error (minADE$_N$), are used to assess the performances of the proposed trajectory prediction method. minADE$_N$ is the displacement error against the closest trajectory in the set of size N. minADE$_N$ [2] is computed to evaluate the method with the multi-modal property.

4.1 Implementation Details

The proposed learning framework is implemented using PyTorch Library [15]. For the selection of the base CNN model, we follow a similar setting as Multipath [2] method. Firstly, the base CNN model is a Resnet50 network with a depth multiplier of 25%, followed by a depth-to-space operation to restore the spatial resolution of the feature map to 200×200. Then we extract patches of size 11×11 centered on agents locations in this feature map followed by a single-layer MLP as the representation of the traffic rules. Then, the 640-dimension feature embedding is calculated from the feature extraction block, concatenated with 256, 256 and 128-dimensional embeddings from the semantic map, observed trajectory, and current status, respectively. For the dynamic attention zone, we set the parameter λ=0.5. We train one model for each class using baseline methods, and only one model for all classes with our method.

4.2 Logistic Delivery Dataset

Our autonomous driving dataset for the logistic delivery purpose is collected by a vehicle equipped with multiple RGB cameras, Lidar and, radar from several regions in Beijing. We benchmark the performance of the proposed method with these baseline methods, including linear, a basic LSTM, Social LSTM(S-LSTM) [1], Social GAN (S-GAN) [7], and Multipath [2]. For the logistic delivery dataset, we sample time steps every 0.2 (5Hz) from the original data and use 2 seconds of history (10 frames) to predict 3 seconds (15 frames) into the future. This dataset contains around 0.8 million agents. We extract approximately 2 million trajectories and use 90% for training and the rest for testing. We compare our method on ADE, FDE, and minADE$_5$ against different baselines and other state-of-the-art methods. We define ADEv, FDEv, ADEc, FDEc, ADEp, and FDEp representing the ADE and FDE of vehicles, cyclists, and pedestrians, respectively. The experimental results for the logistic delivery dataset are shown in Table 2. As expected, the linear method performs the worst for only predicting straight paths. Our method DGAN with setting 20S (k_c=20 with semantic map) performs the best compared with other methods.

Figure 4 illustrates the original labeled dataset, ground truth trajectories, and the top two generated results with probabilities using our method. We compare with different settings of our method, including using or not using the semantic map (Table 2) and the different number of K (Figure 5). The proposed method using the semantic map performs significantly better than without using it for the

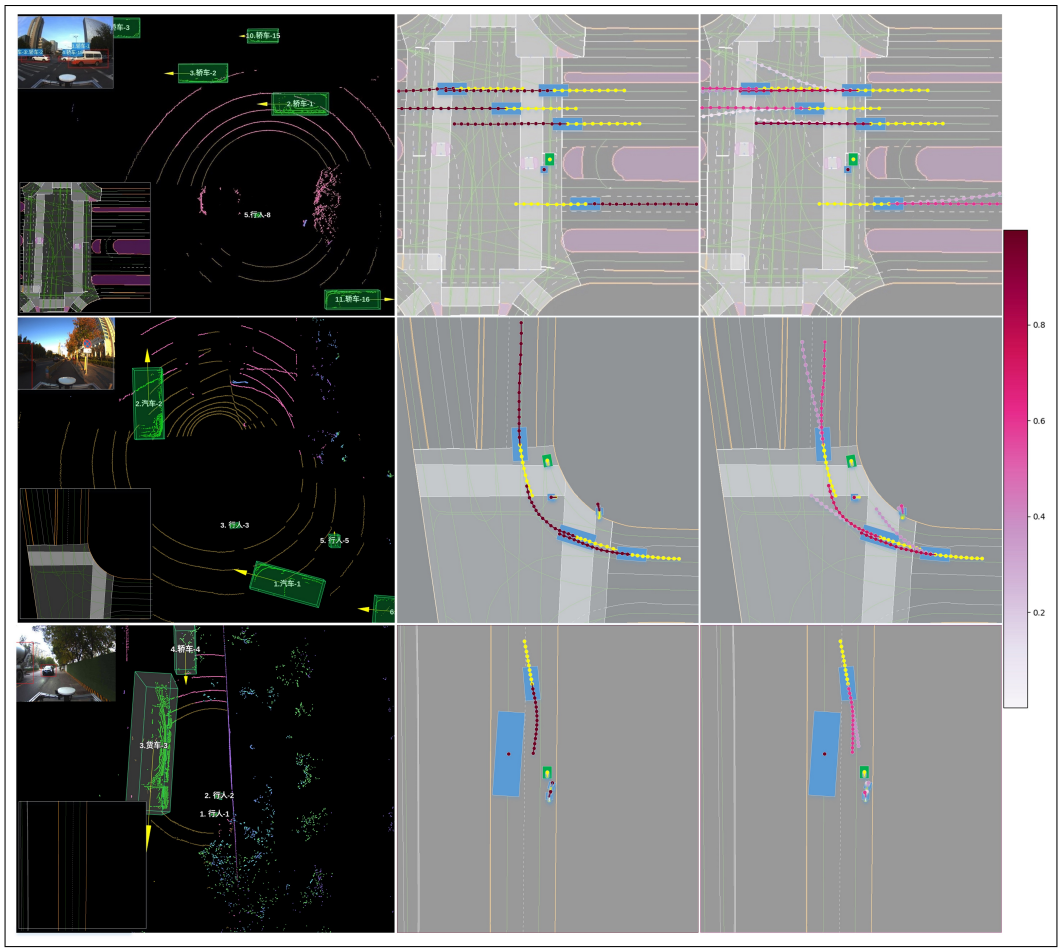

Figure 4: Logistic delivery dataset examples and results using our proposed method DGAN. Left: Logistic delivery dataset example, consisting of three-dimensional cloud points with manually labeled information, front camera image, and semantic map. Middle: observed in dashed yellow and future ground truth trajectories in red. Right: Prediction results using our proposed DGAN method showing up the two most likely future trajectories, and corresponding probabilities encoded in a color map to the right. The green box on the semantic map represents our autonomous driving vehicle, and only agents around it are evaluated using the proposed method.

vehicle and cyclist classes. However, due to the unpredictability of movements of pedestrians and the unavailability of traffic marks in the HD map for pedestrians, the influence of the semantic map is small for the pedestrian class. The results demonstrate that our method can handle complex situations at traffic intersections. It also indicates the predicted trajectory with the maximum probability value is more likely to follow center lines of lanes guiding by the semantic map.

4.3 Stanford Drone Dataset

The Stanford drone dataset [18] is collected by drones in college campus scenarios for trajectory prediction applications, consisting of birds-eye-view videos and labels of multi-class agents, including pedestrians, cyclists, and vehicles. The RGB camera frames encode traffic rule information in a semantic HD map and can serve as input to our method without any modification. For the Stanford drone dataset, we use the direction calculated from positions at the latest two observed time steps as the heading information. We use the length of the labeled bounding box as the length information of the agent. In addition to pedestrians as one class, the largest category in this database, we treat cyclists, skateboarders as one class, and the rest (carts, cars, and buses) as another class. We sample the dataset every 0.4s (2.5Hz) and use five frames of information to predict the trajectory in the next 12 frames. We evaluate the ADE, FDE, and minADE_5 for all agents in the test dataset compared with several state-of-the-art methods, and results are shown in Table 3.

Table 2: Comparison of our proposed method (DGAN) and baselines on our logistic delivery dataset. kS means the method with $K = k$ anchors using our semantic map (the S of kS stands for evaluating with semantic map).

Methods	ADEv	FDEv	ADEc	FDEc	ADEp	FDEp
linear	3.8809	6.7718	3.7221	6.0352	1.5334	3.2096
LSTM	3.2296	5.1659	3.0519	4.8564	1.3536	2.7642
S-LSTM [1]	2.9196	5.0659	2.9519	4.7145	1.2561	2.6018
S-GAN 20VP [7]	2.7276	4.5493	2.7567	4.1431	1.0305	2.2416
Multipath 20S [2]	1.9366	3.2300	1.8573	2.9416	0.9416	1.8603
DGAN 20S (ours)	**1.8398**	**3.0685**	**1.7593**	**2.7945**	**0.9312**	**1.8314**
Methods	minADE_5v	minFDE_5v	minADE_5c	minFDE_5c	minADE_5p	minFDE_5p
S-GAN 20VP [7]	1.6840	2.8835	1.6511	2.6134	0.6645	1.2848
Multipath 20S [2]	1.4595	2.5293	1.1391	2.2136	0.5534	1.1590
DGAN 20 (ours)	1.4697	2.5531	1.1415	2.1918	0.5530	1.1153
DGAN 20S (ours)	**1.4323**	**2.3946**	**1.1309**	**2.1636**	**0.5521**	**1.1134**

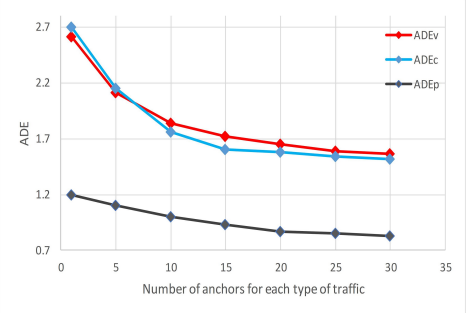

Figure 5: The impact of the number of anchors K_c on the final ADE result for each class.

Table 3: Comparison of our proposed method (DGAN) and other state-of-the-art methods on the Stanford Drone Dataset. Following a similar setting with Multipath [2] method, distance metrics are in terms of pixels in the original resolution.

Methods	ADE	FDE	minADE_5
Linear	26.14	53.24	-
CVAE [10]	30.91	61.40	26.29
DESIRE-SI-IT0 [10]	36.48	61.35	30.78
Social Forces [24]	36.48	58.14	-
S-LSTM [1]	31.19	56.97	-
Multipath μ, Σ [2]	28.32	58.38	17.51
CAR-Net [20]	25.72	51.80	-
DGAN (ours)	**24.53**	**50.78**	**17.28**

4.4 ETH and UCY Datasets

The ETH[16] and UCY[11] datasets for pedestrian trajectory prediction only, include 5 scenes in total, including ETH, HOTEL, ZARA1, ZARA2, and UNIV. The trajectories were sampled every 0.4 seconds. The information in 8 frames (3.2 seconds) is observed and the model predicts the trajectories for the next 12 frames (4.8 seconds). We follow a similar setting with other relevant works [1, 7] for evaluating those two datasets. Results are shown in Table 4.

Table 4: ADE/FDE metrics for several methods on ETH and HCY datasets.

Methods	ETH	HOTEL	UNIV	ZARA1	ZARA2	AVG
Linear	1.33/2.94	0.39/0.72	0.82/1.59	0.62/1.21	0.77/1.48	0.79/1.59
LSTM	1.09/2.41	0.86/1.91	0.61/1.31	0.41/0.88	0.52/1.11	0.72/1.52
S-LSTM [1]	1.09/2.35	0.79/1.76	0.67/1.40	0.47/1.00	0.56/1.17	0.72/1.54
S-GAN [7]	0.81/1.52	0.72/1.61	0.60/**1.26**	0.34/0.69	0.42/0.84	0.58/**1.18**
S-GAN-P [7]	0.87/1.62	**0.67/1.37**	0.76/1.52	0.35/0.68	0.42/0.84	0.61/1.21
Ours	**0.78/1.50**	0.80/1.71	**0.59/1.26**	**0.31/0.64**	**0.39/0.79**	**0.57/1.18**

5 Conclusion

We have introduced a dynamic social interaction-aware model that predicts the future trajectories of agents in real-world settings to solve several challenges simultaneously. In the proposed framework, we use an encoded semantic map, the observed history trajectories, and the current status of agents as the input of the GAT. To generate the graph at the current time step, we use the dynamic attention zone to simulate the intuitive ability of people to navigate roads in real-world traffic. The proposed method is evaluated in different datasets, including our internal logistic delivery dataset and two publicly available datasets. The results demonstrate the potential ability of our method for trajectory prediction in a real-world setting. Through synthetic and real-world datasets, we have shown the benefits of the proposed method over previous methods.

5.1 References

References

[1] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 961–971, 2016.

[2] Yuning Chai, Benjamin Sapp, Mayank Bansal, and Dragomir Anguelov. Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. *arXiv preprint arXiv:1910.05449*, 2019.

[3] Rohan Chandra, Uttaran Bhattacharya, Aniket Bera, and Dinesh Manocha. Traphic: Trajectory prediction in dense and heterogeneous traffic using weighted interactions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8483–8492, 2019.

[4] Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, et al. Argoverse: 3d tracking and forecasting with rich maps. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8748–8757, 2019.

[5] Henggang Cui, Vladan Radosavljevic, Fang-Chieh Chou, Tsung-Han Lin, Thi Nguyen, Tzu-Kuo Huang, Jeff Schneider, and Nemanja Djuric. Multimodal trajectory predictions for autonomous driving using deep convolutional networks. *CoRR*, abs/1809.10732, 2018.

[6] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361. IEEE, 2012.

[7] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[8] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[9] Vineet Kosaraju, Amir Sadeghian, Roberto Martín-Martín, Ian Reid, Hamid Rezatofighi, and Silvio Savarese. Social-bigat: Multimodal trajectory forecasting using bicycle-gan and graph attention networks. In *Advances in Neural Information Processing Systems*, pages 137–146, 2019.

[10] Namhoon Lee, Wongun Choi, Paul Vernaza, Christopher B Choy, Philip HS Torr, and Manmohan Chandraker. Desire: Distant future prediction in dynamic scenes with interacting agents. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 336–345, 2017.

[11] Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. Crowds by example. In *Computer graphics forum*, volume 26, pages 655–664. Wiley Online Library, 2007.

[12] Karttikeya Mangalam, Harshayu Girase, Shreyas Agarwal, Kuan-Hui Lee, Ehsan Adeli, Jitendra Malik, and Adrien Gaidon. It is not the journey but the destination: Endpoint conditioned trajectory prediction. *arXiv preprint arXiv:2004.02025*, 2020.

[13] Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Eleventh annual conference of the international speech communication association*, 2010.

[14] Abduallah Mohamed, Kun Qian, Mohamed Elhoseiny, and Christian Claudel. Social-stgcnn: A social spatio-temporal graph convolutional neural network for human trajectory prediction. *arXiv preprint arXiv:2002.11927*, 2020.

[15] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[16] Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *2009 IEEE 12th International Conference on Computer Vision*, pages 261–268. IEEE, 2009.

[17] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.

[18] Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Learning social etiquette: Human trajectory understanding in crowded scenes. In *European conference on computer vision*, pages 549–565. Springer, 2016.

[19] Amir Sadeghian, Vineet Kosaraju, Ali Sadeghian, Noriaki Hirose, Hamid Rezatofighi, and Silvio Savarese. Sophie: An attentive gan for predicting paths compliant to social and physical constraints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1349–1358, 2019.

[20] Amir Sadeghian, Ferdinand Legros, Maxime Voisin, Ricky Vesel, Alexandre Alahi, and Silvio Savarese. Car-net: Clairvoyant attentive recurrent network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 151–167, 2018.

[21] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. *arXiv preprint arXiv:2001.03093*, 2020.

[22] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[23] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[24] Kota Yamaguchi, Alexander C Berg, Luis E Ortiz, and Tamara L Berg. Who are you with and where are you going? In *CVPR 2011*, pages 1345–1352. IEEE, 2011.