2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)

TPNet: Trajectory Proposal Network for Motion Prediction

Liangji Fang[†⋆] Qinhong Jiang[†⋆] Jianping Shi[†] Bolei Zhou[‡]
[†]SenseTime Group Limited [‡]The Chinese University of Hong Kong
{fangliangji, jiangqinhong, shijianping}@sensetime.com, bzhou@ie.cuhk.edu.hk

Abstract

Making accurate motion prediction of the surrounding traffic agents such as pedestrians, vehicles, and cyclists is crucial for autonomous driving. Recent data-driven motion prediction methods have attempted to learn to directly regress the exact future position or its distribution from massive amount of trajectory data. However, it remains difficult for these methods to provide multimodal predictions as well as integrate physical constraints such as traffic rules and movable areas. In this work we propose a novel two-stage motion prediction framework, Trajectory Proposal Network (TPNet). TPNet first generates a candidate set of future trajectories as hypothesis proposals, then makes the final predictions by classifying and refining the proposals which meets the physical constraints. By steering the proposal generation process, safe and multimodal predictions are realized. Thus this framework effectively mitigates the complexity of motion prediction problem while ensuring the multimodal output. Experiments on four large-scale trajectory prediction datasets, i.e. *the ETH, UCY, Apollo and Argoverse datasets, show that TPNet achieves the state-of-the-art results both quantitatively and qualitatively.*[1]

1. Introduction

Predicting the motion of the surrounding traffic agents, such as vehicles, pedestrians, and cyclists, is crucial for the autonomous driving system to make informative and safe decisions. Traffic agent behaviors tend to be inherently multimodal where there could be multiple plausible intentions for determining their future paths. As illustrated in Fig. 1, Vehicle 1 in green could turn right or go straight under this scenario when only a limited number of observations are received. Moreover the movements of the traffic agents are not only determined by their intentions but also regularized by the nearby traffic rules such as the possible movable areas. For example, vehicles should drive on the road and pedestrians should walk on sidewalks or crosswalks. Thus reliable motion prediction should involve the modeling of the agent's previous trajectory as well as the traffic constraints for the target. Ensuring safe and multimodal predictions is critical for autonomous driving systems.

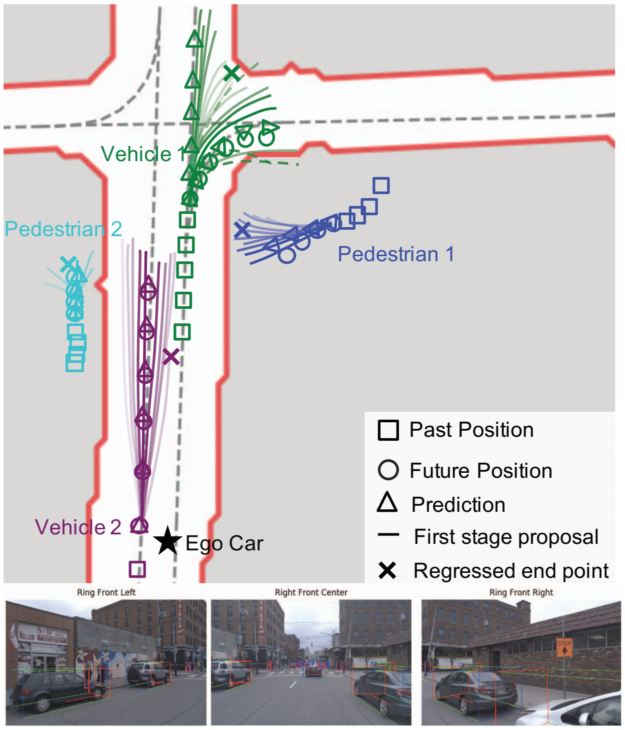

Figure 1. The movements of traffic agents are often regularized by the movable areas (white areas for vehicles and gray areas for pedestrians), while there might be multiple plausible future paths for the agents. Thus it requires the motion prediction systems to be able to incorporate the traffic constraints and output multimodal predictions. Our framework generates the predictions with different intentions under physical constraints for both vehicles and pedestrians.

Early work on motion prediction considers the time-series prediction task by utilizing Kalman Filter based dynamic models [5, 6] or Gaussian mixture models [9], *etc.* However, these models are sensitive to the observation

[⋆] indicates equal contribution.

[1] More information is available at this link.

2575-7075/20/$31.00 ©2020 IEEE
DOI 10.1109/CVPR42600.2020.00683

noise and become unreliable for long-term prediction due to the failure of modeling agent's intention. Recently, many data-driven motion prediction approaches based on deep neural networks have been developed [1, 11, 13, 25, 31, 39, 42, 44, 48]. Most of them attempt to learn the motion patterns by directly regressing the exact future positions or its distributions from the large amount of trajectory data. The multimodal predictions are generated by sampling from the predicted distribution [25, 49, 28]. However, it is difficult for the data-driven approach to provide reasonable multimodal prediction when the distributions of future positions for different intentions are large (*e.g*. turn left and turn right). In order to further ensure predictions complied with the traffic rules, the environment information is often encoded as a semantic map then fed into the neural networks [10, 4]. However, these end-to-end deep networks lack the safety guarantee to make the output prediction strictly follow the traffic rules or semantic map, while it is difficult for them to effectively incorporate the surrounding physical constraints.

In this work, we propose a novel two-stage framework called Trajectory Proposal Network (TPNet) to better handle multimodal motion prediction and traffic constraints. In the first stage, TPNet predicts a rough future end position to reduce the trajectory searching space, and then generates hypothesis as a set of possible future trajectory proposals based on the predicted end point. In the second stage, TPNet performs classification and refinement on the proposals, then outputs the proposal with highest score as the final prediction. Proposals with different intentions can be generated in the first stage to realize diverse multimodal predictions. Prior knowledge such as the movable area constraint is utilized to filtering results of proposals, making this module more effective and transparent. Extensive experimental results have shown that proposing and refining the future trajectories makes the motion prediction more accurate than the ones which directly regress the future positions.

The contributions of this paper are summarized as follows: 1) we propose a unified two-stage motion prediction framework for both vehicles and pedestrians. 2) This framework can incorporate the prior knowledge in the proposal generation process to ensure predictions with multimodal prediction where multiple intentions of the agents are taken into consideration, as well as the compliance of traffic rules and conditions. 3) We achieve the state-of-the-art results on the recent large-scale trajectory prediction datasets ETH [35], UCY [27], ApolloScape [32] and Argoverse [8].

2. Related work

Motion prediction methods can be roughly divided into two categories, the classic methods and the deep learning based methods. Most of the classic motion based algorithms use kinematics equations to model the agent's motion and predict the future location and the maneuver of the vehicle. Comprehensive overview of these approaches can be found in [26, 37]. For future location prediction, statistical models such as polynomial fitting [16], Gaussian processes [23, 43], Gaussian mixture models [9] have been deployed. Kalman Filter based dynamic models [5, 6] have been also wildly used for motion prediction. For maneuver recognition, models like Bayesian networks [41], Hidden Markov models [9, 23], SVMs [3, 33], random forest classifiers [40] are extensively explored. Some of them propose to use scene information to improve prediction [21, 36]. These classical methods model the inherent behaviors based only on the previous movements without considering the uncertainty of driver's decision, thus they can not achieve satisfactory performance in long-term prediction.

Recently many deep learning-based methods have been used for motion prediction [18, 19, 22, 47]. Most of them focus on how to extract useful information from the environment. Convolutional Neural Networks (CNN) Encoder-Decoder is proposed in [46] to extract features from agents' past positions and directions and directly regress the future positions. In [10] the vehicle's location and context information are encoded as binary masks, and a perception RNN is proposed to predict vehicles' location heat-map. The typical pipeline for learning-based prediction methods first encodes the input features, then uses CNN or Long Short-Term Memory(LSTM) [15] to extract features and regress the future locations [2, 24, 34, 45]. However, for these data-driven and deep learning-based methods it is difficult to guarantee the safety and the physical constraints of the prediction. There is another pipeline where the possible trajectory set is first generated based on a lot of motion information (speed, acceleration, angular acceleration, *etc*.) and then optimize the designed cost function to obtain final prediction [16]. However this method heavily relies on the accuracy in the physical measurements, high definition map and the quality of the trajectory set. Different from [16], the proposed TPNet could generate complete proposals only based on trajectory locations. The proposed two-stage pipeline performs further refinement of the proposals which reduces the correlation of the generated proposals and guarantees the diversity of the predictions. Meanwhile, by applying prior knowledge into proposal generation process, our method could take into consideration the physical constraints effectively.

3. Trajectory Proposal Network

To facilitate the safe and multimodal motion prediction, we propose a novel two-stage framework called Trajectory Proposal Network(TPNet). The framework is shown in Fig. 2: In the first stage, base features are extracted from the target agent, then a rough end point is predicted to reduce the proposal searching space. This predicted end point

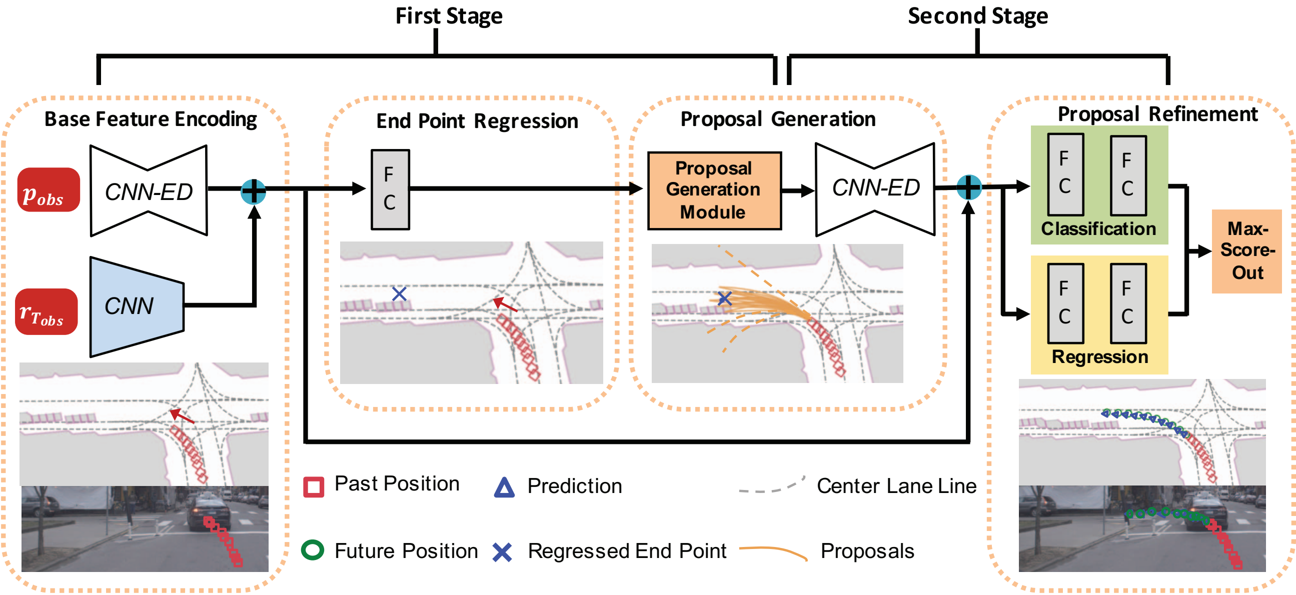

Figure 2. Framework of the Trajectory Proposal Network (TPNet). In the first stage, a rough end point is regressed to reduce the searching space and then proposals are generated. In the second stage, proposals are classified and refined to generate final predictions. The dotted proposals are the proposals that lie outside of the movable area, which will be further punished.

is then utilized to generate proposals. In the second stage, proposals are classified to find the most possible future trajectory and then are refined to ensure the diversity of final predictions.

By monitoring the proposals generated from the proposal generation process in the first stage, the deep learning based prediction method could be more interpretable and flexible. Given the generated proposals, the second stage of the TPNet only needs to choose the most plausible trajectory, which simplifies the prediction problem compared to previous methods of directly regressing the trajectory. Furthermore, it is convenient to debug and explain the possible error predictions by examining the outputs from the two stages respectively.

3.1. Base Feature Encoding Module

The *Base Feature Encoding Module* is designed as an encoder-decoder network due to its flexibility of extending different kinds of input features to the module. The encoder and decoder blocks consist of several convolutional and deconvolutional layers, respectively. The detailed model structure is illustrated in Fig. 2.

The module takes a series of past positions $p_{obs} = \{p_0, p_1, ..., p_{T_{obs}}\}$ in time interval $[0, T_{obs}]$ of the target agent and its surrounding road information $r_{T_{obs}}$ as input, the road information is optional for different dataset. The road information is represented by many semantic elements, *e.g.*, lane line, cross walks, *etc*., and is related to agent's position. For simplicity, we encode the road information as an image and draw targets past positions onto the image, same as [10]. A small backbone ResNet-18 [14] is used to extract features from the road semantic image.

3.2. Proposal Generation

In this section, we introduce the detailed process of Proposal Generation. There are two proposal generation methods depending on whether the road information is utilized or not. The Base Proposal Generation only uses the position information and can be applied to datasets without road information. When combined with road information, the multimodal proposal generation can generate proposals for each possible intentions, ensuring a more compact set of hypotheses.

3.2.1 Problem Definition

In our TPNet, we model the agent trajectory in a limited time as a continuous curve to enable efficiency, flexibility, and robustness. Instead of the traditional representation with discrete point sequence [31, 10] prediction, the continuous curve [16] avoids inefficient combinatorial explosion of future trajectory set and the lack of physical constraint in some combinations. By varying fewer parameters of the curve, we can generate a set of curves flexibly. Curve representation is also robust to noises and could reflect motion tendency and intention.

We choose polynomial curve to represent the trajectory due to its simplicity [16]. To find the best polynomial fit-

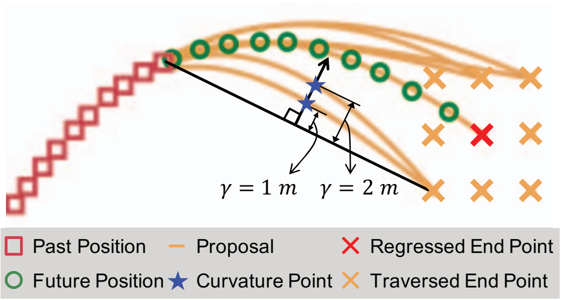

Figure 3. Illustration of proposal generation. Proposals are generated around the end point predicted in the first stage. γ is used to control the shape of the proposal.

ting degree, we conduct experiments with different degrees and calculate the fitting errors for trajectories of time length $T = T_{obs} + T_{pre}$, where T_{obs} is the length for the history observations and T_{pre} is the length for the future predictions. We choose cubic curve with a balance of accuracy and complexity. The average fitting error is $0.048\ m$ for pedestrians on ApolloScape dataset, and $0.068\ m$ for vehicles on Argoverse dataset, which is accurate enough for most cases (detailed analysis can be found in supplemental material).

Since the curve is sensitive to the parameters and difficult to optimize, we propose to use a set of points to represent the curve: two control points, *i.e.*, the end point and curvature point (as shown in Fig. 3), along with the past points. The curvature point reflects curve's crook degree and is determined by a distance variable named γ. γ is defined as the distance between the trajectory curve and the mid-point of the current point and end point as shown in Fig. 3. Encoding curvature point as γ allows to generate curves flexibly with different crook degrees.

3.2.2 Base Proposal Generation

A good proposal generation process should have the ability of generating complete proposals based on less trajectory information. Hence the Base Proposal Generation method generates proposals only based on the trajectory positions, which are the one of the most fundamental and common features provided by almost all trajectory prediction datasets [8, 27, 32, 35]. Given the past positions of an agent, proposals could be generated by varying different control points under curve representation defined in Sec. 3.2.1. Based on the end point p_e predicted in the first stage, possible end points can be generated by enumerating a $N \times N$ grid centered at p_e:

$$p_{ep} = \{(x_e + interval * i, y_e + interval * j)\}_{i,j \in [-N/2, N/2]},$$

where p_{ep} is the possible end points set, (x_e, y_e) is the coordinate of p_e, $interval$ and N are the interval and size of the grid. By varying the values of γ, different curvature points for each possible end point could be generated.

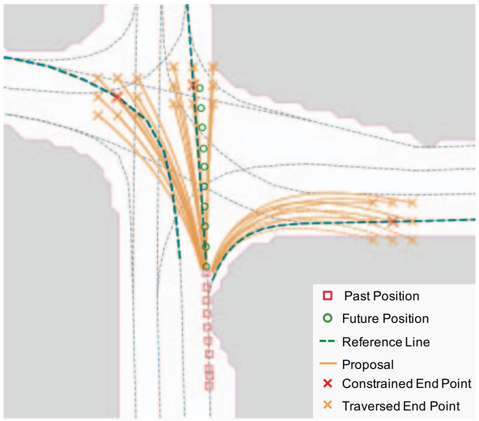

Figure 4. Illustration of multimodal proposal generation using road information. The reference lines indicate the possible center lane lines that the vehicle could dive in. Best viewed in color.

Finally, proposals are generated only based on positions using Eq. 1.

$$proposals = \{f(p_{obs}, p_{ep}^{'}, \gamma)\}, \tag{1}$$

where $f(\cdot)$ is the cubic polynomial fitting function, $p_{ep}^{'} \in p_{ep}$ and $\gamma \in [-2, -1, 0, 1, 2]$.

3.2.3 Multimodal Proposal Generation

Base Proposal Generation has strong dependency on the regressed end point in the first stage which might lead to low diversity of the generated proposals. Multimodal Proposal Generation takes use of road information to generate multiple end points since road has strong constraints on vehicles. Based on the basic elements of the road information (lane lines and their directions, *etc*.) and vehicle's past positions, we can obtain a set of reference lines represent the possible center lane lines the vehicle will reach [8]. Hence Eq. 1 could be extended to generate multiple proposal sets for different reference lines.

Specifically, the relative 1D end position displacement d_{ep} along the reference line is predicted rather than the 2D end point p_e. Then we sample the future end point on each reference line based on the predicted d_{ep} which reduce the dependency on the single regressed end point and ensures the diversity of predictions. Lastly the proposals are generated for each sampled end point using Eq. 1. The process is illustrated in Fig. 4.

3.3. Proposal Classification and Refinement

Given a set of proposals, the *Classification Module* chooses the best proposal while the *Refinement Module* refines the end points and γ of the proposals.

Classification Module. During training, binary class label, denotes good trajectory or not, is assigned to each proposal. We define the average distance between the uniformly sampled points of ground truth and proposal trajectory curves as the criterion of proposal's quality, noted as:

$$AD = \frac{1}{N}\sum_{i=1}^{N} \|p_{gt}^{i} - p_{pp}^{i}\|, \tag{2}$$

where N is the number of sampled points, p_{gt}^{i} and p_{pp}^{i} are the i-th sampled point of ground truth trajectory and proposal, respectively. We assign positive label to a proposal that has an AD lower than a threshold, *e.g.* $1m$. The remaining proposals are potential negative samples. To avoid the overwhelming impact of too many negative samples, we adopt uniform sampling method to maintain the ratio between the negatives and positives as 3:1.

Refinement Module. For proposals refinement, we adopt the parameterization of the 2 coordinates and 1 variate:

$$\begin{cases} t_x = x_e^{gt} - x_e^{pp}, \\ t_y = y_e^{gt} - y_e^{pp}, \\ t_\gamma = \gamma^{gt} - \gamma^{pp}, \end{cases} \tag{3}$$

where (x_e^{gt}, y_e^{gt}) and (x_e^{pp}, y_e^{pp}) are the end point coordinates of ground-truth trajectory and proposal. t_x, t_y and t_γ are the supervised information used during training.

Model Designing. For each proposal, we use the same encoder-decoder module mentioned in Sec. 3.1 to extract features. Then the base features are concatenated with the proposal features. Last two fully connected layers are utilized to classify and refine the proposals, respectively.

3.4. Prior Knowledge

Prior knowledge such as vehicles tend to drive on the road will make the trajectory prediction results more stable and safe. However, DNN-based solutions cannot guarantee these constraints due to the complexity and unexplained nature of the model.

Thanks to the proposal based pipeline, we can use the prior knowledge to filter the proposals explicitly. Combined with historical trajectory and high-definition maps, the polygonal area where the agent can travel in the future is determined, namely movable area. We propose to explicitly constrain the predicted trajectories during inference by decaying the classification scores of the proposals outside of the movable area using Eq. 4:

$$score = score * e^{\frac{-r^2}{\sigma^2}} \tag{4}$$

where r is the ratio that proposal trajectory points outside of the movable area, and σ is the decaying factor.

Compared to abandoning the prediction results outside of the movable area, decaying the classification scores ensures the diversity of the predictions.

3.5. Objective Function

During training, we minimize a multi-task loss as:

$$L = L_{ep}(p_e, p_e^*) + \frac{1}{N}\sum_i L_{cls}(c_i, c_i^*) + \frac{\alpha \sum_i L_{ref}(t_i, t_i^*)}{N_{pos} + \beta N_{neg}}, \tag{5}$$

where p_e and p_e^* are the predicted end point and corresponding ground-truth, c_i and t_i are the predicted confidence and trajectory parameters for each proposal, c_i^* and t_i^* are the corresponding ground-truth labels, α is the weight term. Euclidean loss is employed as the end point prediction loss L_{ep} and the refinement loss L_{ref}. Binary cross entropy loss is employed as the classification loss L_{cls}. Due to the multi-modal property of the future trajectory, we use positive samples along with part of randomly sampled negative samples to calculate the refinement loss and a β to control the ratio of sampled negatives.

4. Experiments

TPNet is evaluated on four public datasets, ETH [35], UCY [27], ApolloScape [31] and Argoverse [8]. **ETH** and **UCY** datasets focus on the pedestrian trajectory prediction. Totally there are five subsets, named ETH, HOTEL, ZARA-01, ZARA-02 and UCY. We follow the same data preprocessing strategy as Social GAN [12]. There are two settings for the length of trajectories, $T_{obs} = T_{pre} = 3.2s$ and $T_{obs} = 3.2s, T_{pre} = 4.8s$. The time interval is set as $0.4s$ for both settings, which results in 8 frames for observation and 8/12 frames for prediction. **ApolloScape** contains bird eye view coordinates of target agents' trajectories along with the trajectories of their surrounding agents. There are three object types need to be predicted, namely vehicle, pedestrian, cyclist. For the length of trajectories, ApolloScape set $T_{obs} = T_{pre} = 3s$ and time interval as $0.5s$, which results in 6 frames for both observation and prediction. **Argoverse** dataset focuses on the prediction of vehicle trajectories. Besides the bird eye view coordinates of each vehicle, Argoverse dataset also provides the high-definition maps. For the length of trajectories, Argoverse set $T_{obs} = 2s, T_{pre} = 3s$ and time interval as $0.1s$. The training, validation and testing sets contain 205942, 39472 and 78143 sequences respectively.

Evaluation Metrics. Average Displacement Error (ADE) and Final Displacement Error (FDE) are the most used metrics in motion prediction. ApolloScape also uses the weighted sum of ADE (WSADE) and weighted sum of FDE (WSFDE) as metrics among different agents types. Argoverse also calculates minimum ADE (minADE), minimum FDE (minFDE) and Drivable Area Compliance (DAC).

- *WSADE/WSFDE*: weighted sum of ADE/FDE among different agents types.

Metric	Dataset	S-LSTM [1]	S-GAN [12]	Liang [30]	Li [29]	SoPhie [38]	STGAT [17]	TPNet-1	TPNet-20
ADE	**ETH**	0.73 / 1.09	0.61 / 0.81	- / 0.73	- / **0.59**	- / 0.70	0.56 / 0.65	0.72 / 1.00	**0.54** / 0.84
	HOTEL	0.49 / 0.79	0.48 / 0.72	- / 0.30	- / 0.46	- / 0.76	0.27 / 0.35	0.26 / 0.31	**0.19 / 0.24**
	UNIV	0.41 / 0.67	0.36 / 0.60	- / 0.60	- / 0.51	- / 0.54	0.32 / 0.52	0.34 / 0.55	**0.24 / 0.42**
	ZARA1	0.27 / 0.47	0.21 / 0.34	- / 0.38	- / **0.22**	- / 0.30	0.21 / 0.34	0.26 / 0.46	**0.19** / 0.33
	ZARA2	0.33 / 0.56	0.27 / 0.42	- / 0.31	- / **0.23**	- / 0.38	0.20 / 0.29	0.21 / 0.33	**0.16** / 0.26
AVG		0.45 / 0.72	0.39 / 0.58	- / 0.46	- / **0.40**	- / 0.54	0.31 / 0.43	0.36 / 0.53	**0.27 / 0.42**
FDE	**ETH**	1.48 / 2.35	1.22 / 1.52	- / 1.65	- / 1.30	- / 1.43	**1.10 / 1.12**	1.39 / 2.01	1.12 / 1.73
	HOTEL	1.01 / 1.76	0.95 / 1.61	- / 0.59	- / 0.83	- / 1.67	0.50 / 0.66	0.48 / 0.58	**0.37 / 0.46**
	UNIV	0.84 / 1.40	0.75 / 1.26	- / 1.27	- / 1.27	- / 1.24	0.66 / 1.10	0.68 / 1.15	**0.53 / 0.94**
	ZARA1	0.56 / 1.00	0.42 / 0.69	- / 0.81	- / **0.49**	- / 0.63	0.42 / 0.69	0.55 / 0.99	**0.41** / 0.75
	ZARA2	0.70 / 1.17	0.54 / 0.84	- / 0.68	- / **0.55**	- / 0.78	0.40 / 0.60	0.43 / 0.72	**0.36** / 0.60
AVG		0.91 / 1.52	0.78 / 1.18	- / 1.00	- / 0.89	- / 1.15	0.62 / **0.83**	0.71 / 1.08	**0.56** / 0.90

Table 1. Comparison with baseline methods on ETH and UCY benchmark for $T_{pre} = 8$ and $T_{pre} = 12$ (8 / 12). Each row represents a dataset and each column represents a method. 20V-20 means that use variety loss and sample 20 times during test time according to [12, 17]. TPNet-20 means we chose the best prediction from proposals with top-20 classification scores.

Metric	Type	S-LSTM	S-GAN	StarNet [50]	TPNet
ADE	**Ped**	1.29	1.33	0.79	**0.74**
	Veh	2.95	3.15	2.39	**2.21**
	Cyc	2.53	2.53	1.86	**1.85**
WSADE		1.89	1.96	1.34	**1.28**
FDE	**Ped**	2.32	2.45	1.52	**1.41**
	Veh	5.28	5.66	4.29	**3.86**
	Cyc	4.54	4.72	3.46	**3.40**
WSFDE		3.40	3.59	2.50	**1.91**

Table 2. Comparison with other methods on the ApolloScape dataset. In the table, Veh, Ped and Cyc indicate agent types of Vehicle, Pedestrian and Cyclist, respectively. Since the ground-truth labels of test set are released, we only report the unimodal result of S-GAN and TPNet.

- *minADE/minFDE*: is the minimum ADE/FDE among multiple predictions (up to K=6) .

- *DAC*: is the ratio of the predicted positions inside the drivable area.

Baselines. Since the multimodal proposal generation and safety guarantee in our proposed method are dependent on high-definition maps, the comparison methods are divided into two groups. The first group consists of methods do not use high-definition maps, including Social LSTM [1] and Social GAN [12]. These baselines are compared on ApolloScape, ETH and UCY dataset. The second group consists of methods that use high-definition maps, including Nearest Neighbor [8] and LSTM ED [8]. These baselines are compared on Argoverse dataset.

- Social LSTM (S-LSTM): uses LSTM to extract features of trajectory and propose social pooling to model social influence for pedestrian trajectory prediction.

- Social GAN (S-GAN): proposes a conditional GAN which takes the trajectories of all agents as input.

- Nearest Neighbor (NN): weighted Nearest Neighbor regression using top-K hypothesized centerlines.

- LSTM ED: LSTM Encoder-Decoder model with road map information as input.

Implementation Details. For network input, road elements within $70m \times 70m$ relative to the target agent is encoded into a semantic map with resolution of 0.5 m/pixel. ResNet-18 [14] is used to extract features of the semantic map. During training, we use data augmentation by randomly rotating and flipping the trajectories. The ratio between negative and positive samples is set to 3:1 and positive AD threshold is set to $3m$ experimentally. We optimize the network using Adam [20] with batch size of 128 for 50 epochs, and learning rate 0.001 with a decay rate 0.9.

4.1. Comparison with Baselines

The effectiveness of the proposed two-stage framework is evaluated on ETH, UCY and Apollo dataset with only target's bird eye view past positions as input in Tab. 1 and Tab. 2. To validate the multimodal prediction and safety guarantee of our proposed method, experiments are conducted on Argoverse dataset as shown in Tab. 3.

Evaluation of Two-stage Framework. The proposed TPNet is compared with the baselines on ETH and UCY datasets in terms of two metrics ADE and FDE in Tab. 1. Following the evaluation methods in S-GAN, we report the results as TPNet-1 and TPNet-20, where TPNet-1 is the prediction with highest classification score while the TPNet-20 result is the best prediction among the predictions with *top-K* classification scores. The results show that the TPNet-1 result already outperforms Social LSTM and the multi-

Methods	ADE	FDE	minADE	minFDE	DAC
NN [7]	3.45	7.88	1.71	3.29	0.87
LSTM ED [7]	2.96	6.81	2.34	5.44	0.90
TPNet	2.33	5.29	2.08	4.69	0.91
TPNet-map	2.23	4.71	2.04	4.23	0.96
TPNet-map-safe	2.23	4.70	2.03	4.22	**0.99**
TPNet-map-mm	**2.23**	**4.70**	**1.61**	**3.28**	0.96

Table 3. Comparison with baseline methods on the Argoverse test set.

modal results of Social GAN. After using the TPNet-20 result, TPNet is competitive with all baselines on all datasets. Note that TPNet only uses the past positions of the target agent while other baselines also utilizing the positions of around agents, which could potentially make our method worse on some datasets.

Then, the performance results of TPNet and the comparison methods on ApolloScape dataset are shown in Tab. 2. From the table we can see that TPNet outperforms the baseline methods on all agent types. Specifically, TPNet performs better on vehicle trajectory prediction and we believe it is because that the curve representation is more friendly to vehicle trajectories.

Evaluation of Multimodal Prediction. TPNet-map-mm in Tab. 3 generates proposals with different intentions based on reference lines mentioned in Sec. 3.2. In the table, TPNet is referred as our method with only past positions as input, TPNet-map as our method with past positions and road semantic map as input. TPNet-map-safe and TPNet-map-mm are referred as using prior knowledge to constrain the proposals and generating multimodal proposals, respectively. In order to evaluate the diversity of the prediction method, Argoverse [8] uses minADE and minFDE as metrics. These two metrics calculate the best ADE and FDE among K number of samples for each target trajectory. After the proposals with different intentions are generated, minADE and minFDE are improved by 60cm and 1m, respectively. Furthermore, the proposed TPNet could generate multimodal predictions even without the use of reference lines. As shown in Tab. 1, the TPNet-20 results on ETH and UCY dataset outperforms the TPNet-1 result by a large margin without the use of reference lines. Because of the proposal generation process, predictions with different intentions could be ensured more effective.

Evaluation of Safety Guarantee. To evaluate the effectiveness of safety guarantee mentioned in Sec. 3.4, we show the experiment results on Argoverse dataset in Tab. 3. Tab. 3 shows that TPNet outperforms the baselines proposed in Argoverse [8] by a large margin, especially on FDE. This indicates that TPNet could generate more accurate end point.

Regression	Classification	ADE	FDE
✗	✗	2.00	4.01
✓	✗	1.85	3.96
✓	✓	**1.75**	**3.88**

Table 4. Ablation Study on the effectiveness of different stages on the Argoverse validation dataset.

Range (m)	Interval (m)	#Anchor	ADE	FDE
6 × 6	1	245	**1.75**	**3.87**
6 × 6	1.5	125	1.78	3.89
6 × 6	3	45	1.84	4.01
10 × 10	1.67	245	**1.75**	3.88
10 × 10	2.5	125	1.76	3.88
10 × 10	5	45	1.84	4.01
20 × 20	3.3	245	1.77	3.93
20 × 20	5	125	1.79	3.98

Table 5. Ablation study on the impact of different grid size for anchor generation on the Argoverse validation dataset.

Furthermore, after taking road semantic map as input, TPNet-map achieves a better results. However the prediction results still may outside the drivable area as the DAC metric still has the room for improvements.

By decaying the classification scores of proposals outside the drivable area using Eq. 4, DAC is improved to 0.99 for TPNet-map-safe which indicates our proposed method could generate more safe prediction results.

4.2. Ablation Study

In this section, we will illustrate the effectiveness of each part of TPNet. We choose Argoverse dataset to do the ablation study for two reasons, 1) the scale of Argoverse dataset is larger than others, 2) Argoverse dataset provides the ground-truth labels for the validation set.

Two-stage Framework. To further validate the effectiveness of modeling trajectory prediction as a two-stage framework, experiments on removing the classification and regression modules step by step are conducted. The results are shown in Tab. 4. By removing the classification and regression simultaneously, the model achieves 4.01 m on FDE metric. The predicted trajectory is obtained by sampling the positions on the curve fitted by the past positions and predicted end position. Then a cascade regressor is utilized to refine the predicted end point and it further improves the FDE by 5 cm as shown in the second row in Tab. 4. Finally the complete two-stage pipeline is experimented and the FDE could be further improved by 8 cm.

Grid Size. The proposed method relies on the quality of generated proposals. The influence of grid size for proposals generation is shown in Tab.5. TPNet will have better results when grid range is set to $6m \times 6m$. As the grid range grows, the performance becomes worse as the searching space becomes larger. And smaller interval size is better.

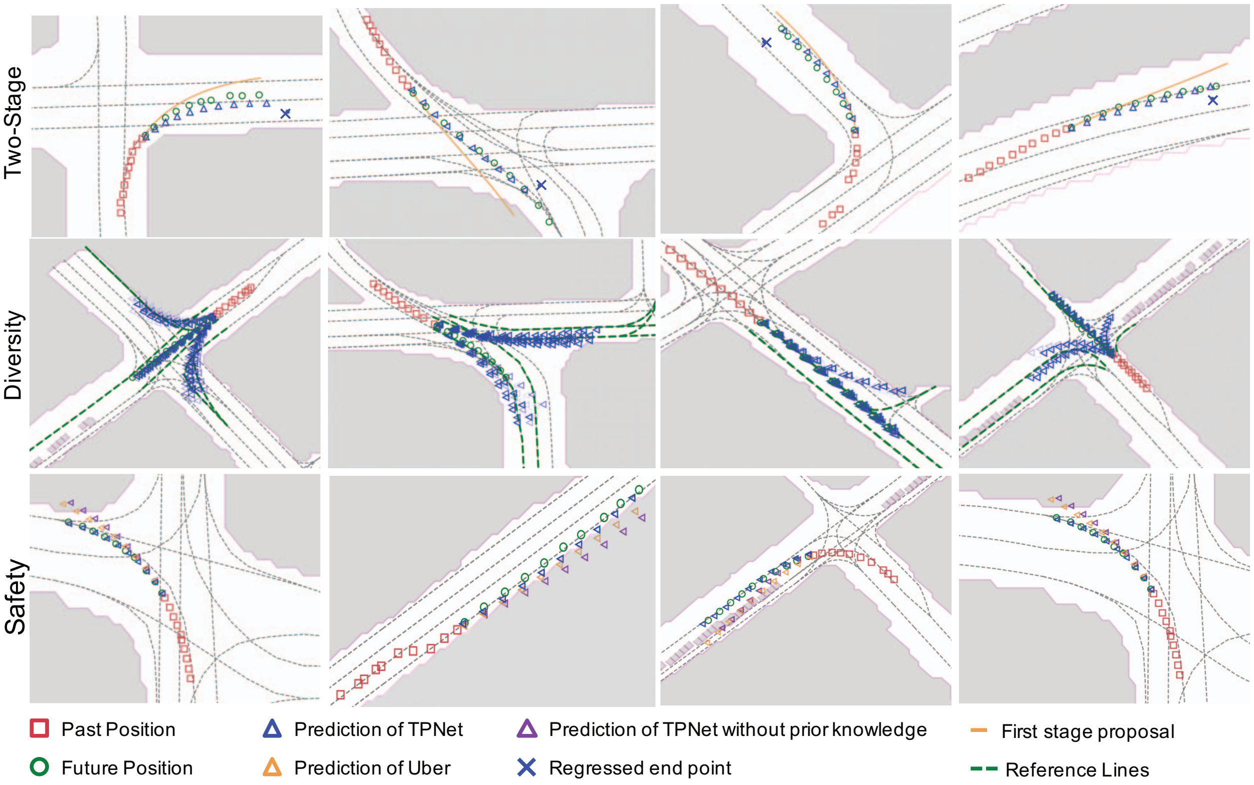

Figure 5. Qualitative results on the effectiveness of each components of TPNet on Argoverse dataset. From top to bottom rows illustrate the effectiveness on the two-stage framework, diversity and safety, respectively. Better viewed in color.

4.3. Qualitative Evaluation

Predicting the motion of traffic agent is challenge because the agent may have different intentions under the same scenario. Furthermore, the possible future paths are not only determined by their intentions but also constrained by the nearby traffic rules. The qualitative results on Argoverse validation set are shown in Fig. 5. Most of the selected scenarios are nearing crossroad. Fig. 5 shows that our method could generate more safe and diverse predictions.

Two-stage Framework. The effectiveness of the proposed two-stage framework is shown in the first row of Fig. 5. The regressed end point might be inaccurate, however the classification and regression processes will refine the prediction results.

Multimodal Output. In the second row of Fig. 5, prediction results under scenarios nearing the crossroad are shown. We can observe multimodal predictions around each possible intentions. Furthermore, the predictions of each intention are also diverse, for example, a vehicle might follow the center lane line or deviate the center lane line.

Safety. In the last row of Fig. 5, we show the results of TPNet (purple triangle), Uber [10] (yellow triangle) and TPNet with safety-guaranteed (blue triangle). Uber [10] encodes the road elements into a raster image and use CNN to regress the future positions. As can be seen in the figure, input the semantic road map to DNN could not ensure the safety of prediction while the proposed decaying function Eq. 4 is more reliable.

5. Conclusion

In this work we propose a two-stage pipeline for more effective motion prediction. The proposed two-stage TPNet first generates the possible future trajectories served as proposals and uses a DNN based model to classify and refine the proposals. Multimodal predictions are realized by generating proposals for different intentions. Furthermore, safe prediction can also be ensured by filtering proposals outside the movable area. Experiments on the public datasets demonstrate the effectiveness of our proposed framework. The proposed two-stage pipeline is flexible to encode prior knowledge into the deep learning method. For example, we can use lamp status which indicates the intention of vehicles to filter the proposals, which will be included in the future work.

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] Florent Altché and Arnaud de La Fortelle. An lstm network for highway trajectory prediction. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 353–359. IEEE, 2017.

[3] Georges S Aoude, Brandon D Luders, Kenneth KH Lee, Daniel S Levine, and Jonathan P How. Threat assessment design for driver assistance system at intersections. In *13th International IEEE Conference on Intelligent Transportation Systems*, pages 1855–1862. IEEE, 2010.

[4] Mayank Bansal, Alex Krizhevsky, and Abhijit Ogale. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. *arXiv preprint arXiv:1812.03079*, 2018.

[5] Alexander Barth and Uwe Franke. Where will the oncoming vehicle be the next second? In *2008 IEEE Intelligent Vehicles Symposium*, pages 1068–1073. IEEE, 2008.

[6] Thomas Batz, Kym Watson, and Jurgen Beyerer. Recognition of dangerous situations within a cooperative group of vehicles. In *2009 IEEE Intelligent Vehicles Symposium*, pages 907–912. IEEE, 2009.

[7] 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.

[8] Ming-Fang Chang, John W Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, and James Hays. Argoverse: 3d tracking and forecasting with rich maps. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[9] Nachiket Deo, Akshay Rangesh, and Mohan M Trivedi. How would surround vehicles move? a unified framework for maneuver classification and motion prediction. *IEEE Transactions on Intelligent Vehicles*, 3(2):129–140, 2018.

[10] Nemanja Djuric, Vladan Radosavljevic, Henggang Cui, Thi Nguyen, Fang-Chieh Chou, Tsung-Han Lin, and Jeff Schneider. Motion prediction of traffic actors for autonomous driving using deep convolutional networks. *arXiv preprint arXiv:1808.05819*, 2018.

[11] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2255–2264, 2018.

[12] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2255–2264, 2018.

[13] Irtiza Hasan, Francesco Setti, Theodore Tsesmelis, Alessio Del Bue, Fabio Galasso, and Marco Cristani. Mx-lstm: mixing tracklets and vislets to jointly forecast trajectories and head poses. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6067–6076, 2018.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

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

[16] Adam Houenou, Philippe Bonnifait, Véronique Cherfaoui, and Wen Yao. Vehicle trajectory prediction based on motion model and maneuver recognition. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4363–4369. IEEE, 2013.

[17] Yingfan Huang, Huikun Bi, Zhaoxin Li, Tianlu Mao, and Zhaoqi Wang. Stgat: Modeling spatial-temporal interactions for human trajectory prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6272–6281, 2019.

[18] Aida Khosroshahi, Eshed Ohn-Bar, and Mohan Manubhai Trivedi. Surround vehicles trajectory analysis with recurrent neural networks. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 2267–2272. IEEE, 2016.

[19] ByeoungDo Kim, Chang Mook Kang, Jaekyum Kim, Seung Hi Lee, Chung Choo Chung, and Jun Won Choi. Probabilistic vehicle trajectory prediction over occupancy grid map via recurrent neural network. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 399–404. IEEE, 2017.

[20] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[21] Kris M Kitani, Brian D Ziebart, James Andrew Bagnell, and Martial Hebert. Activity forecasting. In *European Conference on Computer Vision*, pages 201–214. Springer, 2012.

[22] Puneet Kumar, Mathias Perrollaz, Stéphanie Lefevre, and Christian Laugier. Learning-based approach for online lane change intention prediction. In *2013 IEEE Intelligent Vehicles Symposium (IV)*, pages 797–802. IEEE, 2013.

[23] Christian Laugier, Igor E Paromtchik, Mathias Perrollaz, Mao Yong, John-David Yoder, Christopher Tay, Kamel Mekhnacha, and Amaury Nègre. Probabilistic analysis of dynamic scenes and collision risks assessment to improve driving safety. *IEEE Intelligent Transportation Systems Magazine*, 3(4):4–19, 2011.

[24] Donghan Lee, Youngwook Paul Kwon, Sara McMains, and J Karl Hedrick. Convolution neural network-based lane change intention prediction of surrounding vehicles for acc. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6. IEEE, 2017.

[25] 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.

[26] Stéphanie Lefèvre, Dizan Vasquez, and Christian Laugier. A survey on motion prediction and risk assessment for intelligent vehicles. *ROBOMECH journal*, 1(1):1, 2014.

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

[28] Karen Leung, Edward Schmerling, and Marco Pavone. Distributional prediction of human driving behaviours using mixture density networks. Technical report, Technical report, Stanford University, 2016.

[29] Yuke Li. Which way are you going? imitative decision learning for path forecasting in dynamic scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 294–303, 2019.

[30] Junwei Liang, Lu Jiang, Juan Carlos Niebles, Alexander G Hauptmann, and Li Fei-Fei. Peeking into the future: Predicting future person activities and locations in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5725–5734, 2019.

[31] Yuexin Ma, Xinge Zhu, Sibo Zhang, Ruigang Yang, Wenping Wang, and Dinesh Manocha. Trafficpredict: Trajectory prediction for heterogeneous traffic-agents. *arXiv preprint arXiv:1811.02146*, 2018.

[32] Yuexin Ma, Xinge Zhu, Sibo Zhang, Ruigang Yang, Wenping Wang, and Dinesh Manocha. Trafficpredict: Trajectory prediction for heterogeneous traffic-agents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6120–6127, 2019.

[33] Hiren M Mandalia and Mandalia Dario D Salvucci. Using support vector machines for lane-change detection. In *Proceedings of the human factors and ergonomics society annual meeting*, volume 49, pages 1965–1969. SAGE Publications Sage CA: Los Angeles, CA, 2005.

[34] Seong Hyeon Park, ByeongDo Kim, Chang Mook Kang, Chung Choo Chung, and Jun Won Choi. Sequence-to-sequence prediction of vehicle trajectory via lstm encoder-decoder architecture. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1672–1678. IEEE, 2018.

[35] 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.

[36] Ewoud AI Pool, Julian FP Kooij, and Dariu M Gavrila. Using road topology to improve cyclist path prediction. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 289–296. IEEE, 2017.

[37] Andrey Rudenko, Luigi Palmieri, Michael Herman, Kris M Kitani, Dariu M Gavrila, and Kai O Arras. Human motion trajectory prediction: A survey. *arXiv preprint arXiv:1905.06113*, 2019.

[38] 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.

[39] Amir Sadeghian, Vineet Kosaraju, Ali Sadeghian, Noriaki Hirose, and Silvio Savarese. Sophie: An attentive gan for predicting paths compliant to social and physical constraints. *arXiv preprint arXiv:1806.01482*, 2018.

[40] Julian Schlechtriemen, Florian Wirthmueller, Andreas Wedel, Gabi Breuel, and Klaus-Dieter Kuhnert. When will it change the lane? a probabilistic regression approach for rarely occurring events. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, pages 1373–1379. IEEE, 2015.

[41] Matthias Schreier, Volker Willert, and Jürgen Adamy. Bayesian, maneuver-based, long-term trajectory prediction and criticality assessment for driver assistance systems. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 334–341. IEEE, 2014.

[42] Hang Su, Jun Zhu, Yinpeng Dong, and Bo Zhang. Forecast the plausible paths in crowd scenes. In *IJCAI*, volume 1, page 2, 2017.

[43] Quan Tran and Jonas Firl. Online maneuver recognition and multimodal trajectory prediction for intersection assistance using non-parametric regression. In *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pages 918–923. IEEE, 2014.

[44] Daksh Varshneya and G Srinivasaraghavan. Human trajectory prediction using spatially aware deep attention models. *arXiv preprint arXiv:1705.09436*, 2017.

[45] Long Xin, Pin Wang, Ching-Yao Chan, Jianyu Chen, Shengbo Eben Li, and Bo Cheng. Intention-aware long horizon trajectory prediction of surrounding vehicles using dual lstm networks. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 1441–1446. IEEE, 2018.

[46] Takuma Yagi, Karttikeya Mangalam, Ryo Yonetani, and Yoichi Sato. Future person localization in first-person videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7593–7602, 2018.

[47] Seungje Yoon and Dongsuk Kum. The multilayer perceptron approach to lateral motion prediction of surrounding vehicles for autonomous vehicles. In *2016 IEEE Intelligent Vehicles Symposium (IV)*, pages 1307–1312. IEEE, 2016.

[48] Pu Zhang, Wanli Ouyang, Pengfei Zhang, Jianru Xue, and Nanning Zheng. Sr-lstm: State refinement for lstm towards pedestrian trajectory prediction. *arXiv preprint arXiv:1903.02793*, 2019.

[49] Yu Zhao, Rennong Yang, Guillaume Chevalier, Rajiv C Shah, and Rob Romijnders. Applying deep bidirectional lstm and mixture density network for basketball trajectory prediction. *Optik*, 158:266–272, 2018.

[50] Yanliang Zhu, Deheng Qian, Dongchun Ren, and Huaxia Xia. Starnet: Pedestrian trajectory prediction using deep neural network in star topology. *arXiv preprint arXiv:1906.01797*, 2019.