A Lightweight Goal-Based model for Trajectory Prediction

Amina Ghoul, Kaouther Messaoud, Itheri Yahiaoui, Anne Verroust-Blondet, Fawzi Nashashibi

A Lightweight Goal-Based model for Trajectory Prediction

Amina Ghoul[1], Kaouther Messaoud[2], Itheri Yahiaoui[3], Anne Verroust-Blondet[1] and Fawzi Nashashibi[1]

***Abstract*— We present a lightweight goal-based model for multimodal, probabilistic trajectory prediction for urban driving. Previous conditioned-on-goal methods have used map information in order to establish a set of potential goals and then complete the corresponding full trajectory for each goal. We instead propose two original representations, based on the agent's states and its kinematics, to extract the potential goals. In this paper, we conduct a comparative study between the two representations. We also evaluate our approach on the nuScenes dataset, and show that it outperforms a wide array of state-of-the-art methods.**

I. Introduction

Predicting the future motion of a dynamic agent knowing its past trajectory is crucial in many fields such as advanced surveillance systems and autonomous vehicles. However this task is challenging as it depends on various factors such as the agent's intention, the static environment around the agent, the interaction with other agents and its kinematics. Because of these uncertainties, future motion of agents are inherently multimodal.

Recent works take into account the static environment of agents to predict their trajectories accurately. Most work encodes HD maps using a rasterized bird's eye view image and convolutional layers [1]. These learned map features provide useful context information for motion forecasting. However, the rasterization process can be computationally inefficient and result in information loss. Other studies such as VectorNet [2] or TNT [3] obtain the scene context from graph structures constructed directly from the instance-level semantic map information.

As the agent's intent (goal) carries most of the uncertainty of a trajectory, many goal-based methods have been proposed [4], [5]. For example, TNT [3] defines anchors as the points sampled on the lane centerlines, LaneRCNN [6] takes the lane segments as anchors and predicts a goal for each lane segment. [4] generates a pedestrian's possible goals given the previous trajectories of the humans in the scene. These methods then complete the corresponding full trajectory for each goal.

One limitation of these approaches, is that they sample the possible goals either only from the map information or from the past trajectories of the agents in the scene. We consider however, that the set of possible goals depends on the agent's states (such as its velocity, heading, etc). For example, a fast vehicle should have its farthest potential goals further than the one of a slower vehicle.

Another limitation is that for the task of vehicle trajectory prediction, most of the models only use lane information to extract potential goals. This approach can be problematic if the vehicle is not next to lanes (eg. in a parking lot, or outside of a lane).

To address these limitations, we propose a lightweight goal-based model for multimodal trajectory prediction. We present two novel representations to extract potential goals from.

In this paper, we conduct a comparative study between two representations : a radial grid representation (see Fig. 1) and a kinematic-based representation (see Fig. 2).

Furthermore, our method doesn't use any map information which leads to a simpler and a lighter model, while outperforming a wide array of state-of the-art methods on the nuScenes dataset.

II. Related Work

A. Input Representation

Several state-of-the-art studies [7], [8], [1] represent road information (lane geometry and connectivity, stop lines, crosswalks) and motion history of agents by rasterizing all the scene states in an eye bird view image and deploy a CNN with or without attention network to generate predicted trajectories. However, it is difficult to capture the temporal evolution and the physics of the motion using such a representation. In addition, the rasterization makes a dense representation of the driving scene information which is spatially sparse, this is computationally wasteful representation. As an alternative, recent studies [6], [9] propose to use lane segments information to represent the static scene and agents states. In our work, we propose a lightweight representation based on agents states and potential goals definition.

B. Multi-modal Prediction

In recent years many studies tackle the task of motion prediction using neural network models [10], [11], [12]. To address the multimodality [13] introduced the social LSTM for pedestrian trajectory prediction. They encode the motion of each agent using an LSTM. Then, they extract the interactions between agents by sharing the hidden states between all the LSTMs corresponding to a set of neighboring pedestrians. MHA-JAM [1] applies multi-head attention by considering a joint representation of the static scene and surrounding agents. The authors use each attention head to

This work was carried out in the SAMBA collaborative project, co-funded by BpiFrance in the framework of the Investissement d'Avenir Program.
1: INRIA Paris, France firstname.lastname@inria.fr ;
2: EPFL, Switzerland kaouther.messaoudbenamor@epfl.ch ;
3: CReSTIC, Université de Reims Champagne-Ardenne, Reims, France itheri.yahiaoui@univ-reims.fr

generate a distinct future trajectory to address multimodality of future trajectories.

C. Conditioned-on-Goal Prediction

Several methods such as TNT [3] or LaneRCNN [6] condition each prediction on goals of the driver. Conditioning predictions on future goals makes sense and helps leverage the HD map by restricting goals to be in a certain space. MultiPath [7] and CoverNet [8] chose to quantize the trajectories into anchors, where the trajectory prediction task is reformulated into anchor selection and offset regression. Most of these methods use map information as input of the model and/or to extract anchors. Instead, we are studying the use of two potential goals representations that don't use any map information.

III. Method

A. Problem definition

The goal is to predict the future trajectories of a target agent T $\hat{Y_T} = (\hat{x}_T^t, \hat{y}_T^t)$ from time $t = t_{obs} + 1$ to $t = t_f$. We have as input of our model the track history of the target agent and the n neighboring agents in a scene defined as $\mathbf{X} = [X_1, X_2, ..., X_n]$. Each agent i is represented by a sequence of its states, from time $t = 1$ to $t = t_{obs}$. Each state is composed of a sequence of the agent relative coordinates x_i^t and y_i^t, velocity v_i^t, acceleration a_i^t, heading θ_i^t.

$$X_i^t = (x_i^t, y_i^t, v_i^t, a_i^t, \theta_i^t) \tag{1}$$

The positions of each agent i are expressed in a frame where the origin is the position of the target agent at t_{obs}. The y-axis is oriented toward the target agent's direction of motion and x-axis points to the direction perpendicular to it.

B. Potential goals set representations

Instead of extracting the potential goals from the lane centerlines [3], or generating them only using the social environment of the agent [4], we extract potential goals from two different representations : a **radial grid** and a **kinematics-based** representations.

For the task of pedestrian trajectory prediction, [14] and [15] represent the potential goals set using the pedestrian's current speed and their visual angle to form a radial grid. To the best of our knowledge, this representation has not been used for the task of vehicle trajectory prediction. We use the same grid as [14] for a vehicle (see Fig. 1). The size of the set is defined by the number of speed levels N_s, and number of direction N_d such that $K = N_d \times N_s$. We consider $N_s = 3$ and $N_d = 8$. The potential goals are the center of each grid cell illustrated by the black dots in Fig. 1.

In addition to taking into consideration the current velocity of the target agent, we propose the representation in Fig. 2 that consider its kinematics. Given a steering angle δ and the current velocity of the target agent at t_{obs} v, we calculate the next positions and the heading rates of the target agent (x^t, y^t, θ^t), for $t = t_{obs} + 1, ..., 2 \times t_f$ according to the following equations [16]:

Fig. 1. Radial grid representation (Rep 1).

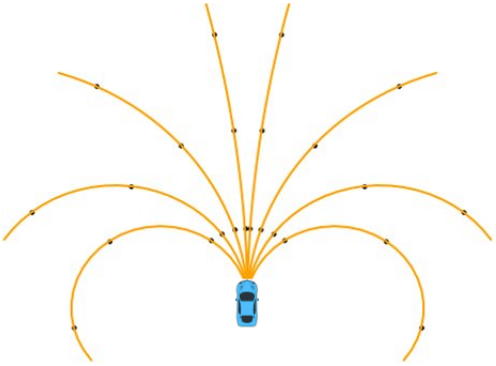

Fig. 2. Kinematics-based representation (Rep 2).

$$x^{t+1} = x^t + dt \times v \times \cos(\theta^t + \delta) \tag{2}$$

$$y^{t+1} = y^t + dt \times v \times \sin(\theta^t + \delta) \tag{3}$$

$$\theta^{t+1} = \theta^t + dt * \delta \tag{4}$$

We obtain N trajectories for each value of δ and we divide them in three to have $K = N \times 3$. We consider $N = 8$ values of steering angles δ between 1 and 17 degrees. Here the potential goals are represented by the black points in Fig. 2. The radial and the kinematic-based representations are built using $N_d = N$ directions.

Along a direction, we consider three potential targets. Each target is defined as the position of the target agent if his velocity was constant and equal to $0.5 \times v_T^{t_{obs}}$, $v_T^{t_{obs}}$, and $2 \times v_T^{t_{obs}}$, respectively, during 6 seconds. Therefore, these two representations are dynamics, as they depend on the current velocity of the target agent.

C. Proposed model

Our model aims at predicting the target agent trajectory by predicting the agent's goal sampled from a representation described in section III-B. We use a multi-head attention-based model proposed by Messaoud *et al.* [17].

For a target agent T at time t, X_T^t is embedded using a fully connected layer to a vector e_i^t and encoded using an LSTM encoder,

$$h_i^t = LSTM(h_i^{t-1}, e_i^t; W_{enc}), \tag{5}$$

W_{enc} are the weights to be learned. The weights are shared between all agents in the scene.

Then we build a social tensor similar to [1]. We define the interaction space of a target vehicle T as the area centered

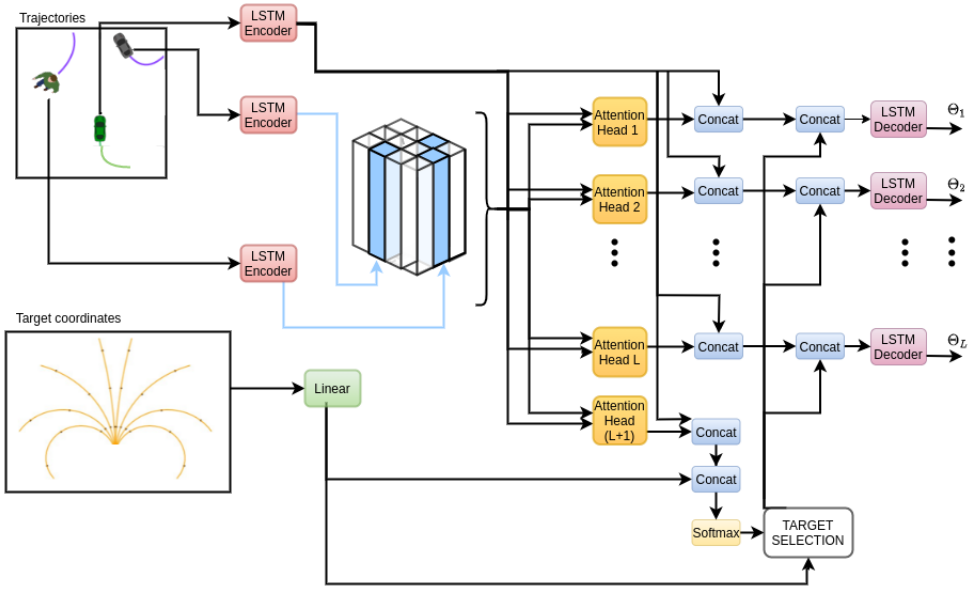

Fig. 3. Proposed model architecture with the representation illustrated in Fig.2. The model takes as inputs the past trajectories of the agents in the scene, as well as the target (or potential goals) coordinates sampled from a representation described in section III-B. Here we illustrate the model with the kinematics-based representation (see in Fig. 2), where the target coordinates correspond to the black points. Our model outputs L trajectories. For more details see section III-C.

on its position at t_{obs} and oriented in its direction of motion. We divide this interaction space into a spatial grid of size (M, N). The trajectory encoder states of the surrounding agents $h_i^{t_{obs}}$ are placed at their corresponding positions in the 2D spatial grid, giving us a tensor F_s of size (M, N, C_h), where C_h is the size of the trajectory encoder state.

We use the multi-head attention mechanism [18] to model the social interactions, where the target vehicle $h_T^{t_{obs}}$. is processed by a fully connected layer to give the query and the social tensor is processed by 1×1 convolutional layer to give the keys and the values.

We consider $L+1$ attention heads where L attention heads are specialized to the L predicted trajectory. This ensures multimodality to our model. And one attention head is used to predict the goal of the target agent (see Fig. 3).

For each attention head, we concatenate the output of the multi-head attention module A_l with the target vehicle trajectory encoder state $h_T^{t_{obs}}$ to give a context representation z_l for $l = 1, ...L + 1$.

$$z_l = Concat(h_T^{t_{obs}}, A_l) \tag{6}$$

Then, we embed the coordinates of the potential goals with a fully connected layer to give the embedding G. We concatenate G with the context representation z_{L+1} associated to the attention head $L+1$. The output is fed to a softmax to give the prediction p_k for each point $k = 1, ..., K$.

$$p_k = Softmax(Concat(z_{L+1}, G_k)) \tag{7}$$

We select the L best scored targets (see Target selection in Fig 3), and we concatenate their embedding to the output of the context representation z_l for $l = 1, ...L$.

Finally, the context vector z_l is fed to an LSTM Decoder which generates the predicted parameters of the distributions over the target vehicle's estimated future positions of each possible trajectory for next t_f time steps,

$$\Theta_l^t = \Lambda(LSTM(h_l^{t-1}, z_l; W_{dec})), \tag{8}$$

where W_{dec} are the weights to be learned, and Λ is a fully connected layer. Similar to [1], we also output the probability P_l associated with each mixture component.

D. Loss function

Our model outputs the means and variances $\Theta_l^t = (\mu_l^t, \Sigma_l^t)$ of the Gaussian distributions for each mixture component at each time step.

The loss for training the model is composed of a regression loss L_{reg} and two classification losses L_{score} and L_{cls}.

L_{reg} is the negative log-likelihood (NLL) similar to the one used in [1] and given by :

$$L_{reg} = -\min_{l} \sum_{t=t_{obs}+1}^{t_{obs}+t_f} log(\mathcal{N}(y^t|\mu_l^t; \Sigma_l^t))). \tag{9}$$

L_{score} is a cross entropy loss defined as :

$$L_{score} = -\sum_{l=1}^{L} \delta_{l*}(l) log(P_l), \tag{10}$$

where δ is a function equal to 1 if $l = l*$ and 0 otherwise.

L_{cls} is also a cross entropy loss defined as :

$$L_{cls} = -\sum_{k=1}^{K} \delta_{k*}(k) log(p_k), \tag{11}$$

TABLE I

COMPARISON BETWEEN THE TWO REPRESENTATIONS ON THE NUSCENES TEST SET (6 SEC HORIZON)

	$MinADE_1$	$MinADE_5$	$MinADE_{10}$	$MinFDE_1$	$MinFDE_5$	$MinFDE_{10}$	$MissRate_{5,2}$	$MissRate_{10,2}$
Rep 1	**2.87**	**1.35**	**1.08**	**6.02**	**2.38**	**1.63**	**0.60**	**0.48**
Rep 2	3.12	1.59	**1.08**	6.70	3.09	1.71	0.63	0.49

TABLE II

COMPARISON WITH STATE-OF-THE-ART METHODS ON THE NUSCENES TEST SET (6 SEC HORIZON)

	$MinADE_1$	$MinADE_5$	$MinADE_{10}$	$MinFDE_1$	$MinFDE_5$	$MinFDE_{10}$	$MissRate_{5,2}$	$MissRate_{10,2}$
MHA-JAM	3.69	1.81	1.24	8.57	3.72	2.22	0.59	0.45
PGP	-	**1.27**	**0.94**	7.17	-	-	**0.52**	**0.34**
P2T	-	1.45	1.16	10.5	-	-	0.64	0.46
GOHOME	-	1.42	1.15	6.99	-	-	0.57	0.47
THOMAS	-	1.33	1.04	6.71	-	-	0.55	0.42
AgentFormer	-	1.86	1.45	-	3.89	2.86	-	-
SG-Net	-	2.1	1.67	-	4.65	3.53	-	-
MHA-LSTM	4.89	2.37	1.81	11.46	5.25	3.83	0.74	0.59
Ours	**2.87**	1.35	1.08	**6.02**	**2.38**	**1.63**	0.60	0.48

TABLE III

COMPARISON BETWEEN FIXED AND DYNAMIC RADIAL GRID REPRESENTATIONS ON THE NUSCENES TEST SET (6 SEC HORIZON)

	$MinADE_1$	$MinADE_5$	$MinADE_{10}$	$MinFDE_1$	$MinFDE_5$	$MinFDE_{10}$	$MissRate_{5,2}$	$MissRate_{10,2}$
Fixed	3.47	1.37	**1.06**	7.62	2.39	**1.59**	0.63	**0.47**
Dynamic	**2.87**	**1.35**	1.08	**6.02**	**2.38**	1.63	**0.60**	0.48

where δ is a function equal to 1 if $k = k*$ and 0 otherwise, k_*^t is the index of the potential goal most closely matching the endpoint of the ground truth trajectory.

Finally, the loss is given by :

$$L = L_{cls} + L_{reg} + L_{score}, \tag{12}$$

E. Implementation details

We use $K = 24$ number of potential goals. The input states are embedded in a space of dimension $C_h = 64$. Similar to [1], our interaction space is 40 m ahead of the target vehicle, 10 m behind and 25 m on each side. We use $L + 1 = 11$ parallel attention operations applied on the vectors projected on different spaces of size d=64. We use a batch size of 64 and Adam optimizer. During training, we help the model by forcing the target selection module to output the closest target from the grid with the highest probability. For the representations of the targets, we use the current velocity of the target agent $v_T^{t_{obs}}$ as explained in III-B. When $v_T^{t_{obs}} = 0$, we replace it with an arbitrary value equal to $0.5\ m.s^{-1}$. The model is implemented using PyTorch [19]. We trained our model for 500 epochs on the nuScenes dataset, which took approximately 4 hours. We used a Nvidia GeForce GTX 1080 Ti. Our model only takes 30 seconds per epoch for training.

IV. Experiments

A. Dataset

We evaluate our model on the nuScenes [20] dataset. It is a large-scale dataset for autonomous driving with 1000 scenes in Boston and Singapore. Each scene is annotated at 2 Hz and is 20s long, containing up to 23 semantic object classes as well as HD semantic maps with 11 annotated layers. We train and evaluate our model using the official benchmark split for the nuScenes prediction challenge, with 32,186 prediction instances in the train set, 8,560 instances in the validation set, and 9,041 instances in the test set.

TABLE IV

SPEED STATISTICS ON THE NUSCENES DATASET (M/S)

Mean	5.81
Standard deviation	3.86
Minimum	0.0
Maximum	23.45

B. Evaluation metrics

Our method for trajectory forecasting is evaluated with the following three error metrics:

- **Minimum Average Displacement Error over k (**$minADE_k$**)** : The average of pointwise L2 distances between the predicted trajectory and ground truth over the k most likely predictions.
- **Minimum Final Displacement Error over k (**$minFDE_k$**)** : The final displacement error (FDE) is the L2 distance between the final points of the prediction and ground truth. We take the minimum FDE over the k most likely predictions and average over all agents.
- **Miss Rate At 2 meters over k (**$MissRate_{2,k}$**)** : If the maximum pointwise L2 distance between the prediction and ground truth is greater than 2 meters, we define the prediction as a miss. For each agent, we take the k most likely predictions and evaluate if any are misses. The $MissRate_{2,k}$ is the proportion of misses over all agents.

C. Comparison of the two representations

We compare the two representations described in section III-B. The results are reported in Table I. We can see that using the radial grid representation (Rep 1) gives better results than the kinematics-based representation (Rep 2). It can be noticed that the radial grid representation better covers the space in front of the vehicle, which can make it more appropriate. Moreover, the kinematics-based representation contains potential goals behind the vehicle, which are rarely the ground-truth goals in the nuScenes dataset.In fact, we can see in Fig. 5 that in the training dataset, most of the observations have a target situated in front of the vehicle (alternative 9 to 14).

Fig. 4. Choice set representation (Rep 1), with numbering of alternatives.

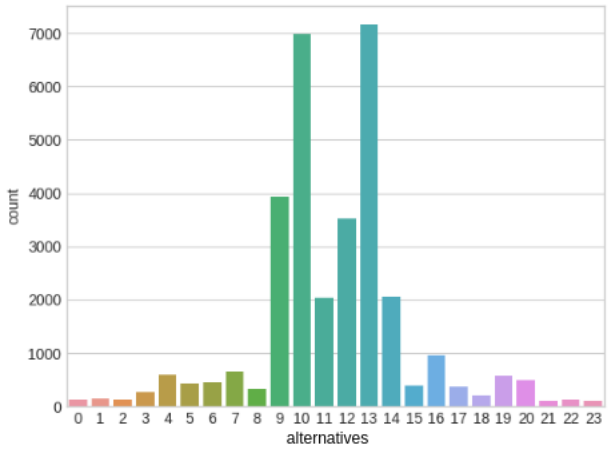

Fig. 5. Revealed choices histograms for Rep 1

D. Comparison with State-of-the-art

Table II reports results on the nuScenes prediction benchmark. We compare our models with the winning entries of the nuScenes prediction challenge, MHA-JAM [1] and PGP [9], P2T [21], THOMAS [22] and GOHOME [23]. All of these methods use rasterization of HD maps (MHA-JAM, P2T), or the encoding of lane information using graphs (PGP, GOHOME).

Additionally, we compare our approach to other models that don't use rasterized HD maps or no graphs, AgentFormer [24], SG-Net [25] and MHA-LSTM [17]. The results of SGNet are from [25] of the model where the map information are not used. We implemented and evaluated the MHA-LSTM on the nuScenes dataset.

We can see that our approach significantly outperforms methods that don't use map information. Moreover, against the state-of-the art, our model achieves very competitive performance and attains the best results for $MinADE_1$, $MinFDE_1$, $MinFDE_5$ and $MinFDE_{10}$ and the third best results for $MissRate_{10,2}$, $MissRate_{5,2}$ and $MinADE_{10}$.

We believe the strong performance of our method can be attributed to the unique representation of our model.

E. Fixed vs dynamic representation

We study the importance of introducing a dynamic representation (i.e a representation that depends on the target agent's current velocity). To do so, we compare the results of the proposed model with a dynamic radial representation described in section III-B, with a fixed representation. For the fixed representation, the radial grid does not depend on the velocity $v_T^{t_{obs}}$, but it is built using the value $v = 5.81m.s^{-1}$, which corresponds to the mean of the velocities in the nuScenes training set. Evaluation results are reported in Table III. We can see that for most of the metrics, the dynamic representation is better than the fixed one. However, for $K = 10$, the fixed representation achieve slightly better results than the dynamic one. We can conclude that the dynamic representation is better as it performs better when considering few trajectories.

F. Complexity comparison

Our proposed model essentially removes a CNN module of the original MHA-JAM model while having better results. Table V shows the complexity comparison between our model and MHA-JAM. We notice that our model has way less trainable parameters than MHA-JAM, which makes it very light and fast to train.

TABLE V

COMPLEXITY COMPARISON

Model	Number of trainable parameters
MHA-JAM	227M
Ours	0.6M

V. Conclusion and future work

We presented a lightweight goal-based model that achieves competitive results and outperforms several state-of-the-art models in the nuScenes dataset. We showed that without any information about the static environment of the vehicle, we managed to obtain great results thanks to our proposed goal set representations. As future work we plan to combine lane information with the the proposed representations in order to make our model more scene-compliant. Moreover, we plan to explore other types of grid to extract potential goals. As most of the trajectory on the nuScenes dataset are straight in front of the vehicle (see Fig. 5), we can try other types of

representations to extract the potential goals and compare it to the radial and kinematics-based grids.

References

[1] K. Messaoud, N. Deo, M. M. Trivedi, and F. Nashashibi, "Trajectory prediction for autonomous driving based on multi-head attention with joint agent-map representation," in *IEEE Intelligent Vehicles Symposium, IV 2021, Nagoya, Japan, July 11-17, 2021*. IEEE, 2021, pp. 165–170.

[2] J. Gao, C. Sun, H. Zhao, Y. Shen, D. Anguelov, C. Li, and C. Schmid, "VectorNet: Encoding hd maps and agent dynamics from vectorized representation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[3] H. Zhao, J. Gao, T. Lan, C. Sun, B. Sapp, B. Varadarajan, Y. Shen, Y. Shen, Y. Chai, C. Schmid, C. Li, and D. Anguelov, "TNT: target-driven trajectory prediction," in *4th Conference on Robot Learning, CoRL 2020*, ser. Proceedings of Machine Learning Research, vol. 155, 2020, pp. 895–904.

[4] K. Mangalam, H. Girase, S. Agarwal, K.-H. Lee, E. Adeli, J. Malik, and A. Gaidon, "It is not the journey but the destination: Endpoint conditioned trajectory prediction," in *European Conference on Computer Vision (ECCV)*. Springer, 2020, pp. 759–776.

[5] N. Rhinehart, R. McAllister, K. Kitani, and S. Levine, "PRECOG: prediction conditioned on goals in visual multi-agent settings," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR)*, 2019, pp. 2821–2830.

[6] W. Zeng, M. Liang, R. Liao, and R. Urtasun, "LaneRCNN: Distributed representations for graph-centric motion forecasting," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 532–539.

[7] Y. Chai, B. Sapp, M. Bansal, and D. Anguelov, "MultiPath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction," in *CoRL*, 2019.

[8] T. Phan-Minh, E. C. Grigore, F. A. Boulton, O. Beijbom, and E. M. Wolff, "CoverNet: Multimodal behavior prediction using trajectory sets," *CoRR*, vol. abs/1911.10298, 2019. [Online]. Available: http://arxiv.org/abs/1911.10298

[9] N. Deo, E. Wolff, and O. Beijbom, "Multimodal trajectory prediction conditioned on lane-graph traversals," in *Conference on Robot Learning (CoRL)*, 2022, pp. 203–212.

[10] J. Liu, X. Mao, Y. Fang, D. Zhu, and M. Q. Meng, "A survey on deep-learning approaches for vehicle trajectory prediction in autonomous driving," in *IEEE International Conference on Robotics and Biomimetics, ROBIO 2021, Sanya, China, December 27-31, 2021*. IEEE, 2021, pp. 978–985.

[11] F. Leon and M. Gavrilescu, "A review of tracking and trajectory prediction methods for autonomous driving," *Mathematics*, vol. 9, no. 6, p. 660, 2021.

[12] S. Mozaffari, O. Y. Al-Jarrah, M. Dianati, P. Jennings, and A. Mouzakitis, "Deep learning-based vehicle behavior prediction for autonomous driving applications: A review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 1, pp. 33–47, 2020.

[13] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social LSTM: Human trajectory prediction in crowded spaces," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 961–971.

[14] T. Robin, G. Antonini, M. Bierlaire, and J. Cruz, "Specification, estimation and validation of a pedestrian walking behavior model," *Transportation Research Part B: Methodological*, vol. 43, no. 1, pp. 36–56, 2009.

[15] P. Kothari, B. Sifringer, and A. Alahi, "Interpretable social anchors for human trajectory forecasting in crowds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 15 556–15 566.

[16] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, "Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data," in *European Conference on Computer Vision (ECCV)*. Springer, 2020, pp. 683–700.

[17] K. Messaoud, I. Yahiaoui, A. Verroust-Blondet, and F. Nashashibi, "Attention based vehicle trajectory prediction," *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 1, pp. 175–185, 2020.

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[19] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.

[20] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuScenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, 2020, pp. 11 621–11 631.

[21] N. Deo and M. M. Trivedi, "Trajectory forecasts in unknown environments conditioned on grid-based plans," *arXiv preprint arXiv:2001.00735*, 2020.

[22] T. Gilles, S. Sabatini, D. Tsishkou, B. Stanciulescu, and F. Moutarde, "THOMAS: trajectory heatmap output with learned multi-agent sampling," in *ICLR*, 2022.

[23] ——, "GOHOME: graph-oriented heatmap output for future motion estimation," *arXiv preprint arXiv:2109.01827*, 2021.

[24] Y. Yuan, X. Weng, Y. Ou, and K. M. Kitani, "AgentFormer: Agent-aware transformers for socio-temporal multi-agent forecasting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9813–9823.

[25] C. Wang, Y. Wang, M. Xu, and D. Crandall, "Stepwise goal-driven networks for trajectory prediction," *IEEE Robotics and Automation Letters*, 2022.