Generative Pedestrian Trajectory Prediction with Graph Representation

Haoyu Zhang
North China University of Technology, Beijing, China
tts7274tech@hotmail.com

Abstract. Multi-agent trajectory prediction is one of the core modules of unmanned driving and intelligent robots. The traditional method is difficult to measure the relationship between multiple agents, and the modeling ability is rigid. Nowadays, most of the methods make less use of geographic information, and social relationship modeling is not sufficient. Our model uses Graph Neural Network (GNN) to measure social relationships to improve its usability. The model is built by the basic Conditional Auto-encoder (CVAE) framework, using Gaussian Mixture Model to mix multiple Gaussian distributions and the possibility of obtaining more potential space through dynamic integration to make the model achieve better results. Our model has achieved excellent results on the Stanford Drone Dataset. The evaluation by average displacement error (ADE) and final displacement error (FDE) metrics has exceeded the majority of existing models.

1. Introduction

Predicting the future course of action of pedestrians in life is a key module of driver-less and intelligent robots. In real life, humans will follow common sense to avoid contact or collision, and in different scenarios, people often have different common-sense rules, such as in crowded cities and parks with loose trails. The factors that affect such common-sense rules are often the geographical environment and the actions of others.

Geographical factors will affect people’s future trajectories. For example, people often have limited movements in narrow spaces, but in a wide space, people have a large range of action. People will also turn at the corner of the sidewalk and most people will stop moving in front of the fence, and people will try to avoid walking on the lawn. Many existing methods make use of such factors, such as: MATF[3], Sophie[2], and CGNS[4]. Such geographical factors often limit the overall direction of people’s actions, and we use Convolutional Neural Networks (CNN) to map information encoding so that the model takes environmental factors into account when making predictions about agent actions.

Other people’s actions will affect people’s course of action. For example, people will predict the trajectory of others to prevent collision with others; when people pass between two people in parallel and close to each other, it will be very impolite, so people will also try to avoid this behavior; for scenes that gather many people, people may be attracted to it, and so on. Therefore, the future action strategies of others will affect the details of people’s actions. This is the element used by many models that solve Multi-agent, such as: Social-LSTM[6], Social-Force[7], Social-GAN[1], Social-WaGDAT[31], EvolveGraph[28]. We use GNN to model the social relationship between people and use Long Short-Term Memory (LSTM) encodes the relationship between the target and others and the actions of others, and uses the Bi-directional LSTM (BLSTM) to encode the target’s future actions, so
that the model’s influencing factors on other people’s action routes are sensitive and detailed adjustments are made.

Speed and other factors will also affect people’s trajectory. For example, people prove that people who encounter running will avoid in advance. When the distance is too close, people may not be able to avoid it. There are many influencing factors like this. We regard pedestrians as nodes and use position and velocity to model them. Of course, you can add elements that may affect it to complete the model, so that the model can adapt to the situation of different agents.

The model as a whole uses InfoVAE’s\cite{11},\cite{12} objective function through the framework of CVAE\cite{10}. In the decoder, we use Gate Recurrent Unit (GRU) to generate multiple normal distributions of future trajectories and mix multi-peaks through Gaussian Mixture Model (GMM)\cite{20}. We use the method of dynamic integration to enable the model to discover more latent variable space so that the model can pursue more possibilities. In the end, in Stanford Drone Datasets (SDD)\cite{22}, the ADE and FDE indicators were tested, and better results were obtained than other models.

2. Related Work

2.1. Trajectory Prediction

In recent years, intelligent robotics and autonomous vehicles have emerged and played a significant role in people’s daily lives. As an essential functional module of the intelligent systems, trajectory prediction has been widely studied by researchers. Trajectory can be viewed as a sequence and predicted by recurrent neural networks (RNN) or LSTM such as \cite{6},\cite{17},\cite{19}, and it can also be predicted by deep generative models such as generative adversarial networks (GAN), such as \cite{1},\cite{2},\cite{15},\cite{21},\cite{25} or use the frame of CVAE to explicitly encode multi-modality, such as \cite{5},\cite{8},\cite{18},\cite{26},\cite{29},\cite{32}. For multi-agent trajectory prediction, the relationship between various agents can be expressed with GNN. Loads of researches on it, such as \cite{9},\cite{23},\cite{30}.

2.2. Graph Neural Network

The graph data structure plays a vital role in mathematical modeling since it is widely applicable. Traditional graph modeling usually represents low-dimensional feature vectors and the attribute of a single node by graph embedding. However, graph neural network (GNN) can represent the attributes of multiple nodes. Recently, many kinds of GNN have been developed by researchers, such as Graph Convolution Networks (GCN), Graph Attention Networks (GAT)\cite{13}, Graph Auto-Encoders (GAE)\cite{14}, Graph Generative Networks (GGN), Graph Spatial-temporal Networks (GSTN)\cite{16}. We take advantage of the useful properties of GNN to express the attributes of multiple nodes to express the relationship of the involved agents, such as vehicle-pedestrian interactions, vehicle-vehicle interactions, etc.

2.3. Generative Modeling

The generative models have an advantage over discriminative models since they learn the underlying data distribution in nature. In recent years, as the development of deep learning techniques, deep generative models such as Variational Auto-Encoder (VAE), Generative Adversarial Network (GAN), and their variants have been widely studied. Besides unconditional deep generative models, there have been conditional counterparts that handle the generation tasks with specific goals or preferences. In the trajectory prediction tasks, since we need to forecast the future behaviors of various agents based on their historical information, it is natural to use conditional generative models. In this work, we employ the Conditional Variational Auto-Encoder (CVAE) as the basis of the prediction framework.

2.4. Problem Statement

Pedestrian and vehicle trajectory forecasting has been playing an important role on the autonomous agents in society. In this research, we aim to get the possible trajectory distribution of agents that changes with time $t$. And we have $N_t$ agents in time $t$. We define interacting agents as $R_1,R_2,...,R_{N_t}$.
Each $R_i$ has 2 classes: pedestrian and vehicle. At time $t$, we define agent $R_i$ state as $S^t_i \in \mathbb{R}^D$. So we can get all state information from $t_0$ to $t$ of each agent: $x = \{S^{t_{0:t}}_{1,2,\ldots,N_t}\} \in \mathbb{R}^{(t-t_0+1) \times N_t \times D}$. Our goal is to train a model to generate the trajectory which from time to: $y = \{S^{t+1:t+H}_{1,2,\ldots,N_t}\} \in \mathbb{R}^{H \times N_t \times D}$.

3. Method
In this section, we first provide a high-level overview of the proposed prediction framework, which illustrates the core components and their functions. Then, we demonstrate the detailed model architecture and prediction algorithms. Finally, we introduce the loss function and the training procedures. Since all the components in the framework are implemented by deep neural networks, the whole model can be trained in an end-to-end fashion.

3.1. Model Overview
Our approach uses a Spatio-temporal graph to quantify the scene in question is created from a pedestrian position. Then we use a deep learning model which including Long Short-term Memory Network (LSTM) and dense to encode the edge and node information, GRU, and GMM to decode.

3.2. Encoding

![Diagram](Figure 1. We use the node represent agents’ information and the edge represents a social relationship, then feeds this information into LSTM and gets the output vector. The map is fed into 4 layers CNN and get the map vector. We need the node future to train the model, so we use BLSTM to encode them. Finally, we get a concatenation of them and feed it into Dense.)
We use a graph to represent the real scene. In the following, the graph is represented by \( G = (V, E) \). In graph \( G \), nodes represent pedestrians and edges represent their interaction such as relative position. The edge \((P_i, P_j)\) represents the influence between \( P_i \) and \( P_j \).

In this model, Euclidean Distance is used to measure the degree of mutual influence.

\[
(x_i - x_j)^2 + (y_i - y_j)^2 \leq d
\]

where \( x_i \) and \( y_i \) are the 2D world position of \( P_i \), similarly \( P_j \), \( d \) is the threshold value to determine whether the two influence each other.

![Figure 2. Classic scene of pedestrians avoiding each other. Many factors in life affect people’s future trajectories, the most intuitive factor is the distance between people.](image)

You can also use other conditions to express the influence between them. Simultaneously, you need more computing power to construct graphs and train models. The GPU we used in this work was NVIDIA RTX TITAN and it took about 10 minutes to represent the scene.

\[
x = S_{t, 2, \ldots, N_t}^{(t, t)} \in \mathbb{R}^{(t - t_0 + 1) \times N_t \times D}
\]

We fed it into the model and \( D \) is the D-dimensional state of the agents.

In this work, we would like to model the influence between pedestrian by edges from graph \( G \). The edge information is aggregated from the neighboring pedestrian. And we use the sum of the element-wise to express the aggregated operation. It can save count information for prediction. After completing the operation, we input the result into eight hidden dimension LSTMs. When the weights are updated, all the weights from edges will update too. And we will get the vector which represents all neighboring nodes’ influence. At last, we concatenate the node history and the edge influence vector to get a single node representation vector, \( I_x \).
Figure 3. First, calculate the edge of the graph element-wise sum and then feed it into 32-dimension-LSTM to complete the encoding of the edge.

We also use the HD map information. HD maps are often used by trajectory prediction. It has some geological information that can help the model to predict. We encode the local map, and we adjust it to the same orientation as the agent with the four layers CNNs. It has filters 5,5,5,3 and respective strides of 2,2,1,1. And then fed into a fully connected layer with thirty-two dimensions, and the output is concatenated with the influence representation vector and the history information.

Figure 4. Map encoding module
When we train the model, we need the real future ego-agent trajectory vector. In this work, we use a Bi-directional Long Short-term memory network (BLSTM) which has thirty-two dimensions to encode it. Each training sequence is two Long Short-term Memory networks (LSTM) forward and backward, and these two are connected to an output layer. In most cases, BLSTM has an excellent performance in sequence models. And the output can are concatenated into the vector $I_x$.

Our model can process multimodal data by CVAE latent variable framework. It can output the trajectory prediction distribution. And we use a bi-directional LSTM to encode the node real future trajectory for the training model.

3.3. Decoding

![Diagram](image)

Figure 5. Use encoder output and $[e_x, e_R, z, y_t \text{ or } \hat{y}_t]$ as GRU input and GRU output multiple distributions. Use GMM to encode multimodality explicitly for them, and then optimize integration through dynamic integration.

The decoder is made of Gated Recurrent Unit (GRU) and Gaussian Mixed Model (GMM), and we use the sampling of distribution $Z$ as the first input, and we use graph information represent and robot represent $e_x, e_R$, distributed sampling $z$, t-time Route information $y_t$ as the second input feed into GRU. We get some gaussian distributions $P$ as the output from GRU. And we use GMM to obtain trajectory distribution $P_t$. Then $P_t$ is fed into the Dynamics Integration and gets the most confidence trajectory.

We define $p = [x, y]^T$ as position and $u = [\omega, a]^T$ as input action. And we get the position mean at $u_p^{(t+1)} = u_p^{(t)} + u_u^{(t)} \Delta t$, where $u_u^{(t)}$ is the output with the model.
3.4. Loss Function and Training

The function modified based on InfoVAE object function. Since the model uses CVAE framework, it uses discrete latent states in the conditional formulation. In this model, we need to solve the optimization problem, which is given by

$$\phi, \theta, \psi \sum_{i=1}^{N} \mathbb{E}_{x \sim q_{\theta}(x|i,y)} \left[ \log p_{\psi}(y|x_i, z) \right] - \beta D_{KL}(q_{\phi}(z|x_i, y_i)||p_{\psi}(z|x_i)) + \alpha I_{\theta}(x; z)$$

(3)

where $\beta$ parameter is the KL penalty term weight which to generate multi-modalities and simplify the steps to disentangle the latent space. The $\alpha$ parameter we usually make it 1.0.

$I_{\theta}$ is the mutual information of $x$ and $z$ under the distribution $q_{\theta}(x, z)$. To find $I_{\theta}$, we use $q_{\phi}(z|x_i)$ to approximate $q_{\theta}(z|x_i, y_i)$, and get the unconditional distribution by summing $x$ in the batch. Simultaneously, Gumbel-Softmax don’t need to use $z$ backpropagation, because there is not sample during training. And in this formula, the first term can be computed directly because $z$ only have twenty-five elements.

4. Experiments

In this section, we first introduce the datasets used in our experiments. Then we present the widely used evaluation metrics and state-of-the-art baseline methods. Finally, we provide detailed quantitative and qualitative analysis of the two benchmark datasets to demonstrate the superiority of our proposed approach.

4.1. Datasets: Stanford Drone Dataset

Stanford Drone Dataset was collected in a set of crowded spaces such as the bookstore and the sidewalks of the streets. It includes the track ID, frame ID, mark size, label of agents, and other related information. There are six types of agents: bicyclist, pedestrian, skateboarder, cart, car, and bus. These agents move actively and interactively in outdoor environments on a university campus. In the videos, different agents are indicated by bounding boxes in different colors. For example, pedestrians are labeled in pink, bikers in red, skateboarders in orange, cars in green, etc. This data is pretty suitable for our task. In this work, we employ the trajectory information of all the involved agents.

4.2. Evaluation Metrics

We illustrate the widely used evaluation metrics in the following:

- **Average Displacement Error (ADE):** Average $l_2$ distance between the real trajectories and the predicted ones.

$$A(i) = \frac{1}{T} \sum_{j=1,2,...,T'} \sqrt{(x_{ij} - x_{ij'})^2 + (y_{ij} - y_{ij'})^2}$$

(4)

$$A = \frac{1}{n} \sum_{i=1,2,...,n} A(i)$$

(5)

- **Final Displacement Error (FDE):** At the prediction horizon $T$, $l_2$ distance between the real final position and the predicted final position.

$$F(i) = \sqrt{(x_{i_{T'}} - x_{i_{T''}})^2 + (y_{i_{T'}} - y_{i_{T''}})^2}$$

(6)

$$F = \frac{1}{n} \sum_{i=1,2,...,n} F(i)$$

(7)

4.3. Baselines

We compare the proposed approach with several state-of-the-art methods, which are introduced in the following:

- avoid Social-LSTM: Each person’s trajectory data is processed using LSTM, and the social layer shares the state to represent social relationships.

- Social-GAN: Use LSTM and pooling module to encode the track and then use LSTM to decode. Compared with traditional GAN, the introduction of variety loss helps GAN to develop the distribution of trajectories in space.
- Social-Force: Formulate social force between each pedestrian. It's traditional manual functions.
- MATF: MATF uses the output feature map of the multi-agent coding channel and scene coding full convolutional network to maintain the spatial structure between the agent and the scene, and uses Conditional GAN to capture the predicted trajectory.
- Desire: Desire generates multiple reasonable prediction samples $\hat{Y}$ through the CVAE-based RNN encoder-decoder, then the subsequent module serves as an IOC framework, assigns rewards to the prediction samples in turn at each time step, and learns the displacement vector $\Delta \hat{Y}$ to regression prediction Assumptions.
- Sophie: SoPhie is a GAN-based framework that obtains two pieces of information from the scene, namely the historical path and scene context information of all agents in the scene. Sophie combines social attention mechanisms and physical attention to help the model learn what to see and extract the most obvious image parts related to the path.

### 4.4. Quantitative Evaluation
We cite some test results on the SDD dataset from MATF paper, including generative baselines such as Social-GAN, Sophie, etc and some deterministic baselines such as LSTM, Social LSTM, etc.

| Method                  | ADE  | FDE  |
|-------------------------|------|------|
| LSTM Baseline           | 37.35| 77.13|
| Social Force [8]        | 36.38| 58.14|
| Social LSTM [7]         | 31.19| 56.97|
| Social GAN [2]          | 27.25| 41.44|
| MATF Multi Agent [4]    | 30.75| 65.90|
| MATF Multi Agent Scene [4]| 27.82| 59.31|
| MATF GAN [4]            | 22.59| 33.53|
| Desire [9]              | 19.25| 34.05|
| SoPhie [3]              | 16.27| 29.38|

| Method                  | ADE  | FDE  |
|-------------------------|------|------|
| Ours                    | 27.13| 47.66|
| Ours + Dynamic          | 15.99| 22.62|
| Ours + Dynamic + Map    | 16.20| 22.12|

From the results, it can be found that when only the normal model is used for processing, the testing effect is at the mid-stream level in the comparison model. When dynamic interaction is used, ADE and FDE are greatly reduced. After adding map encoding, FDE is reduced. ADE Increased. Dynamic interaction can have more variability in prediction, which will make the model find a closer trajectory. map encoding provides more geographic information, such information may mislead the model when other agents are far away, increasing ADE, but geographic information will limit the possible distribution of the model during prediction to a certain extent, so it leads to a decrease in FDE.
4.5. Qualitative Evaluation

Figure 6. In this picture, Red is the real trajectory, green is the set of points sampled on the output distribution. This picture is the quad data part of the SDD data, and subsequent analysis will also be performed by this picture.

Figure 7. It can be seen from this picture that the model does take into account the distance relationship between people during prediction.
Figure 8. It can be seen that without dynamic integration.

Fig.6, 7, 8 shows the actual effect of our model in SDD. The red is the real trajectory, and the green is the result of sampling the distribution predicted by the model. It can be seen from Fig. 7 that the model adjusts the predicted path according to the relationship between pedestrians. From the comparison of the two figures in Fig. 8, it can be seen that without dynamic integration, the distribution predicted by the model cannot cover the real path, and the model cannot find more latent distributions. So dynamic integration is an important performance improvement module.

4.6. Colour illustrations

Table 2. ADE/FDE OF BASELINE METHODS AND PROPOSED METHOD

| Method                | ADE  | FDE  |
|-----------------------|------|------|
| Ours                  | 27.13| 47.66|
| Ours + Map            | 25.90| 46.01|
| Ours + Dynamic        | 15.99| 22.62|
| Ours + Dynamic + Map  | 16.20| 22.12|

From the data, we can see that adding Dynamic Integration and Map Encoding will improve the effect.

Compared with Our, both ADE and FDE only reduce the single digits of Our+Map. The reason may be that for the terrain on the SDD campus, the impact on pedestrians is more reflected in the turns, and the impact on the straight line route is not large.

Compared with Our, Our+Dynamic has improved a lot in ADE and FDE. The reason may be that Dynamic Integration’s recalculation of control action and position provides more feasibility for model prediction, thus obtaining a more accurate prediction trajectory.

Although Our+Dynamic+Map has higher ADE compared to Our+dynamic, FDE has decreased. The reason may be that the geographic information of Map encoding has an impact on the prediction. For example, the model believes that it is impossible to walk in the grass, and in real life, someone will pass through the grass, which increases the ADE. However, the limitation of geographic
information on the prediction distribution improves the stability of the prediction path, thereby reducing FDE.

5. Conclusion
We improved the basic model based on the basic framework of CVAE and the function of InfoVAE. We quantify the social interaction of agents by GNN. And we used them and path information to train the model. Our method has achieved good results on the SDD dataset. Later, the map module and dynamic integration module were added after synthesizing these two modules, outperforming state-of-the-art methods on SDD datasets. We show that by combining information about geographic information and the interaction between all agents, our model can learn something compared to when this information is used alone, and finding more potential variables through dynamic integration can greatly improve the effect.

References
[1] G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529-551, April 1955.
[2] Gupta A, Johnson J, Fei-Fei L, et al. Social GAN: Socially Acceptable Trajectories with Generative Adversarial Networks”. 2018.
[3] Sadeghian A, Kosaraju V, Sadeghian A, et al. SoPhie: An Attentive GAN for Predicting Paths Compliant to Social and Physical Constraints”. arXiv, 2018.
[4] Zhao T, Xu Y, Monfort M, et al. Multi-Agent Tensor Fusion for Contextual Trajectory Prediction”. 2019.
[5] Li, J., Ma, H., Tomizuka, M. Conditional Generative Neural System for Probabilistic Trajectory Prediction. In 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 6150-6156). IEEE.
[6] Ivanovic B, Pavone M. The Trajectron: Probabilistic Multi-Agent Trajectory Modeling With Dynamic Spatiotemporal Graphs[C] ” 2019 IEEE/CVF International Conference on Computer Vision (ICCV). IEEE, 2019.
[7] 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 International Conference on Computer Vision and Pattern Recognition, 2016. 2, 4, 7
[8] D. Helbing and P. Molnar. Social force model for pedestrian dynamics. Physical review E, 51(5):4282, 1995. 2, 4, 7
[9] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. S. Torr, and M. Chandraker. DESIRE: distant future prediction in dynamic scenes with interacting agents. In 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017, pages 2165-2174, 2017. 2, 4, 7, 8
[10] Kipf T, Fetaya E, Wang K C, et al. Neural Relational Inference for Interacting Systems[J]. 2018.
[11] Bao, Jianmin Chen, Dong Wen, Fang Li, Houqiang Hua, Gang. (2017). CVAE-GAN: Fine-Grained Image Generation through Asymmetric Training.
[12] Zhao S, Song J, Ermon S. InfoVAE: Information Maximizing Variational Autoencoders[J]. 2017.
[13] H. Ma, J. Li, W. Zhan and M. Tomizuka, "Wasserstein Generative Learning with Kinematic Constraints for Probabilistic Interactive Driving Behavior Prediction," 2019 IEEE Intelligent Vehicles Symposium (IV), Paris, France, 2019, pp. 2477-2483, doi: 10.1109/IVS.2019.8813783.
[14] Velickovic, Petar, Cucurull G, Casanova A, et al. Graph Attention Networks[J]. 2017.
[15] Kipf T N, Welling M. Variational Graph Auto-Encoders[J]. 2016.
[16] J. Li, H. Ma and M. Tomizuka, "Interaction-aware Multi-agent Tracking and Probabilistic Behavior Prediction via Adversarial Learning," 2019 International Conference on Robotics
and Automation (ICRA), Montreal, QC, Canada, 2019, pp. 6658-6664, doi: 10.1109/ICRA.2019.8793661.

[17] Yan S, Xiong Y, Lin D. Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition"]. 2018.

[18] Morton, J., Wheeler, T.A., Kochenderfer, M.J.: Analysis of recurrent neural networks for probabilistic modeling of driver behavior. IEEE Transactions on Pattern Analysis Machine Intelligence 18(5), 1289-1298 (2017)

[19] Deo, M.F., Trivedi, J: Multi-modal trajectory prediction of surrounding vehicles with maneuver based lstms. In: IEEE Intelligent Vehicles Symposium (2018)

[20] Vemula, A., Muelling, K., Oh, J: Social attention: Modeling attention in human crowds. In: Proc. IEEE Conf. on Robotics and Automation (2018)

[21] Kosaraju, V., Sadeghian, A., Mart'ın-Mart'ın, R., Reid, I., Rezatofghi, S.H., Savarese, S.: Social-BiGAT: Multimodal trajectory forecasting using bicycle-GAN and graph attention networks. In: Conf. on Neural Information Processing Systems (2019)

[22] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese. Learning social etiquette: Human trajectory prediction in crowded scenes. European Conference on Computer Vision (ECCV), 2016.

[23] Ma Y, Zhu X, Zhang S, et al. TrafficPredict: Trajectory Prediction for Heterogeneous Traffic-Agents[J]. 2019.

[24] Battaglia, Peter & Hamrick, Jessica & Bapst, Victor & Sanchez- Gonzalez, Alvaro & Zambaldi, Vinicius & Malinowski, Mateusz & Tacchetti, Andrea & Raposo, David & Santoro, Adam & Faulkner, Ryan & Gulcehre, Caglar & Song, Francis & Ballard, Andrew & Gilmer, Justin & Dahl, George & Vaswani, Ashish & Allen, Kelsey & Nash, Charles & Langston, Victoria & Pascanu, Razvan. Relational inductive biases, deep learning, and graph networks. (2018).

[25] Ivanovic, B.; Schmerling, E.; Leung, K.; Pavone, M.: Generative modeling of multimodal multi-human behavior. In: IEEE/RSJ Int. Conf. on Intelligent Robots & Systems (2018)

[26] Battaglia P W, Pascanu R, Lai M, et al. Interaction Networks for Learning about Objects, Relations and Physics[J]. 2016.

[27] Li, J., Yang, F., Tomizuka, M., Choi, C. (2020). EvolveGraph: Heterogeneous Multi-Agent Multi-Modal Trajectory Prediction with Evolving Interaction Graphs. arXiv preprint arXiv:2003.13924.

[28] Sadeghian A, Legros F, Voisin M, et al. CAR-Net: Clairvoyant Attentive Recurrent Network[J]. 2017.

[29] Yeh, Raymond & Schwing, Alexander & Huang, Jonathan & Murphy, Kevin. (2019). Diverse Generation for Multi-Agent Sports Games. 46054614. 10.1109/CVPR.2019.00474.

[30] Li, J., Ma, H., Zhang, Z., Tomizuka, M. (2020). Social-wagdat: Interaction-aware trajectory prediction via wasserstein graph doubleattention network. arXiv preprint arXiv:2002.06241.

[31] Rhinehart N, Mcallister R, Kitani K, et al. PRECOG: PREdiction Conditioned On Goals in Visual Multi-Agent Settings[J]. 2019.