2021 IEEE/CVF International Conference on Computer Vision (ICCV) | 978-1-6654-2812-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICCV48922.2021.01292

Unlimited Neighborhood Interaction for Heterogeneous Trajectory Prediction

Fang Zheng[1] Le Wang[2*] Sanping Zhou[2] Wei Tang[3] Zhenxing Niu[4] Nanning Zheng[2] Gang Hua[5]
[1]School of Software Engineering, Xi'an Jiaotong University
[2]Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University
[3]University of Illinois at Chicago
[4]School of Computer Science and Technology, Xidian University
[5]Wormpex AI Research

Abstract

Understanding complex social interactions among agents is a key challenge for trajectory prediction. Most existing methods consider the interactions between pairwise traffic agents or in a local area, while the nature of interactions is unlimited, involving an uncertain number of agents and non-local areas simultaneously. Besides, they treat heterogeneous *traffic agents the same, namely those among agents of different categories, while neglecting people's diverse reaction patterns toward traffic agents in different categories. To address these problems, we propose a simple yet effective Unlimited Neighborhood Interaction Network (UNIN), which predicts trajectories of heterogeneous agents in multiple categories. Specifically, the proposed unlimited neighborhood interaction module generates the fused-features of all agents involved in an interaction simultaneously, which is adaptive to any number of agents and any range of interaction area. Meanwhile, a hierarchical graph attention module is proposed to obtain category-to-category interaction and agent-to-agent interaction. Finally, parameters of a Gaussian Mixture Model are estimated for generating the future trajectories. Extensive experimental results on benchmark datasets demonstrate a significant performance improvement of our method over the state-of-the-art methods.*

1. Introduction

The challenges hampering prediction accuracy largely stem from the complex interactions among agents [1, 12, 57]. Recent advances in this regard [1, 3, 23, 26, 27] mainly fall into two types: Graph-based methods [13, 32, 56] build a spatial graph at each time step and aggregate the features from adjacent nodes; RNN-based methods [4, 7, 57] model each agent's trajectory with Recurrent Neural Networks (RNNs) and pool hidden states within a surrounding area.

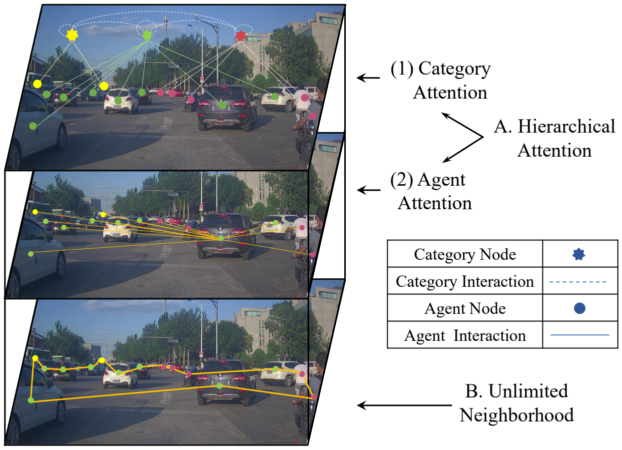

Figure 1. Hierarchical Graph Attention & Unlimited Neighborhood Interaction. Different marked shapes are used to distinguish the agent and category. **(A) Hierarchical Attention.** The agents marked by the same color belong to the same category. *(1) Category Attention.* Every category interacts with each other and itself. Attention of one category to all categories (including the attention to itself) is transferred to all agents of the category. *(2) Agent Attention.* We compute one's attention to the rest of the agents in the whole scenario, and the attention is directed. **(B) Unlimited Neighborhood.** We consider the interaction as among a collective, rather than between two agents or in a small area. The behavior of any agent may influence a group of agents around the whole scenario.

However, these methods suffer from limitations. Graph-based methods [32] only exploit pairwise relation between the nodes, while other nodes are mixed and relayed. GCN with many layers suffers from over-smoothing problem [9, 19]. In contrast, the interaction in real-world traffic is much more complex than previously assumed, such as multilateral relations (relation among three or more agents). Namely, these methods are limited by inflexible numbers of interaction agents.

[*] Corresponding author.

978-1-6654-2812-5/21/$31.00 ©2021 IEEE
DOI 10.1109/ICCV48922.2021.01292

Moreover, RNN-based methods [4, 7, 57] merely consider the local relations among an agent's manually defined surrounding area. As a result, potential interaction participants outside of such "surrounding area" will be simply discarded. Namely, these methods are limited by such hand-crafted way for interaction agent selection.

To solve these problems, we propose the Unlimited Neighborhood on heterogeneous graph to predict the future trajectories of multi-categories (*e.g.*, pedestrians, bikes, cars, .etc), as shown in Figure 1. Unlimited Neighborhood means the interactions are not limited by the number of agents or the range of area. Namely, any agent in a scenario could be involved in an interaction, as illustrated in Figure 2. In addition, many related works [12, 56] treat different agents as the homogeneous ones (*i.e.*, pedestrians), while a real traffic scenario usually involves heterogeneous agents (*i.e.*, agents in diverse categories). Due to the difference in movement patterns (*e.g.*, velocity, front and rear distance and the response to the interaction) for agents in different categories, trajectory prediction on heterogeneous agents is exactly more challenging compared to that on homogeneous ones.

Specifically, we present a simple yet effective Unlimited Neighborhood Interaction Network for heterogeneous trajectory prediction, which models the hierarchical attention and fuses all agents involved in one interaction to predict the future trajectories for all agents with different categories simultaneously. Then, regarding the agents as nodes and the agents with the same category as a category node, we can construct a spatio-temporal-category graph combining spatial, temporal and category information together. The hierarchical graph attention module acquires the category-category attention and then the agent-agent attention on the constructed graph. Note that the edges in the constructed graph are directed. Namely the edges are represented as a weighted asymmetric adjacency matrix to measure the interactions.

Once obtained the hierarchical interactions, an unlimited neighborhood interaction module is employed to capture the global information of all agents involved in the same interaction by an asymmetric convolutional network. Based on the global information and the hierarchical attention, the final interaction is obtained and fed into a Graph Convolutional Network (GCN) [32] which is followed by a Temporal Convolutional Network (TCN) [2], to estimate the parameters of Gaussian Mixed Model (GMM) [37].

Experimental results on multiple benchmark datasets demonstrate significant performance improvement of our method over the state-of-the-art methods. The visualization shows our method can learn the interaction among heterogeneous agents well. The code will be published upon acceptance.

In summary, the key contributions of this paper include:

- We propose to model the interaction among heterogeneous agents to improve the trajectory prediction;
- We present an Unlimited Neighborhood Interaction for modeling the interaction among the agents involved in the same interaction simultaneously;
- We present a Hierarchical Graph Attention module for enhancing the agent-to-agent interaction based on category-to-category interaction.

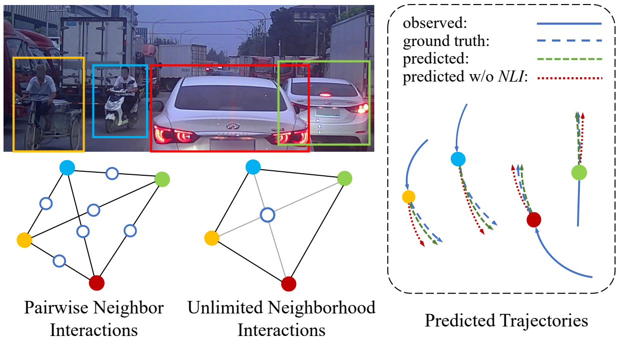

Figure 2. Comparison between Unlimited Neighborhood Interaction and Pairwise Neighborhood Interaction (*e.g.*, GCN). Different agents are enclosed in differently colored boxes, corresponding to solid circles in the same color. The hollow circle denotes an interaction. As can be seen, an interaction involves a group of agents in our method. On the right side, we show the predicted trajectories with or without our Unlimited Neighborhood.

2. Related Works

Trajectory prediction mainly involves homogeneous and heterogeneous trajectory prediction in real scenarios. *Homogeneous* trajectory prediction predicts future trajectories under the same category (*e.g.*, only pedestrians). On the contrary, *heterogeneous* predicts future trajectories under different categories (*e.g.*, pedestrians, cars and bikes).

2.1. Homogeneous Trajectory Prediction

Prior to the prevalence of deep learning, there are classical methods [47, 48, 52], including Social Force models [16], Gaussian Process regression models [47], dynamic Bayesian models [54] and hidden Markov models [44], which are limited by hard-to-design hand-crafted features.

Thanks to the representational power of deep neural networks, trajectory prediction is recently dominated by deep learning based methods, such as Recurrent Neural Networks (RNNs) [1], Generative Adversarial Networks (GANs) [12], Graph Convolutional Networks (GCNs) [32, 43] and Transformers [56]. S-LSTM [1] aggregates the interaction information through a pooling mechanism. S-GAN [12] predicts multiple socially acceptable trajectories using GANs. Later works measure the influence of interaction by attention mechanism. S-BiGAT [20] uses Graph Attention Networks [50] to model the interactions between pedestrians. STAR [56]

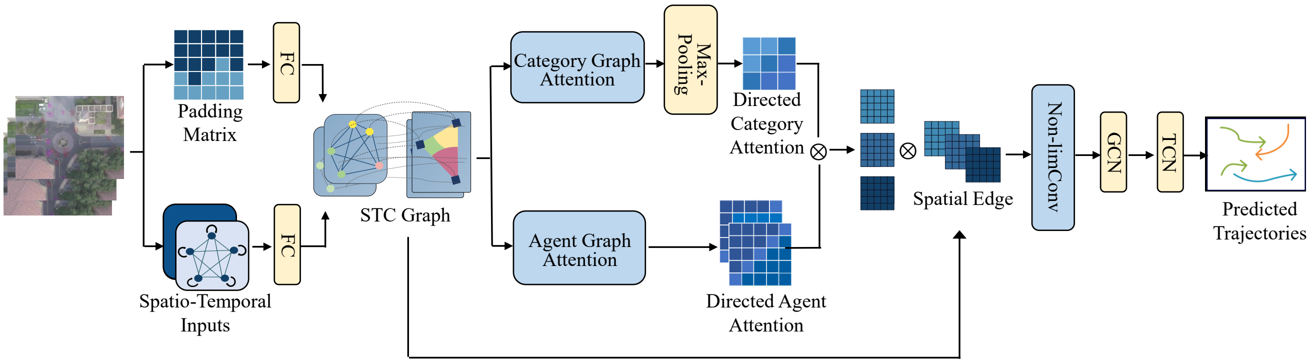

Figure 3. Framework of our UNIN. The trajectories are reformed as spatiao-temporal and category inputs, and a spatio-temporal-category graph (STC graph) is composed. Hierarchical Attention learns directed category attention representing the category interactions, and directed agent attention representing the agent interactions from the STC graph. Collective interactions are captured by subsequent Unlimited Neighborhood Interaction with the asymmetric attention matrix, and then fed into a spatio-temporal graph convolutional network and temporal convolutional network to estimate the parameters of the Gaussian Mixture Model, from which the future trajectories are predicted.

separately models spatial interaction and temporal continuity through Transformer [49] architecture on graph.

Since physical constraints in the scenario and human states are the predominant factors in trajectory prediction [40] under certain circumstances, extensive studies focus on the role of physical information recently [6, 27, 46]. Sophie [40] leverages both physical and social information to predict pedestrian trajectory. Notably, CVM [15] takes pedestrians' velocity and direction into account rather than semantic environment. ECTP [30] infers trajectory endpoints first as additional information to assist pedestrians' planning path. Different from pedestrian trajectory prediction, vehicle trajectory prediction methods can take advantage of more sensors and semantic environments, such as 3D point cloud and lane line [28, 55].

2.2. Heterogeneous Trajectory Prediction

Homogeneous traffic agents like pedestrians, vehicles follow different social conventions, and thus the homogeneous trajectory prediction methods cannot simultaneously model the interaction of all agents of different categories in the same scene and predict accurately.

Heterogeneous trajectory prediction in the real traffic scenes gradually attracted more research interest. JPKT [3] treats vehicles as rigid particles, where non-particle objects are subject to kinematics, and model vehicles and pedestrians with separate LSTMs. DATF [34] models agent-to-agent and agent-to-scene interactions and proposes a new approach to estimate trajectory distribution. In brief, these methods focus on different behavior patterns of heterogeneous traffic agents, and the influence of the semantic environment.

While previous works ignore the interaction at the categorical granularity and unlimited interactions among agents, we model the interaction among all agents. In our method, physical constraints are implicitly learned through observed trajectories without environmental semantics as prior.

2.3. Graph Neural Network

Graph neural network (GNN) [42] extends the neural network to process data without nature order. GNN learns a state vector embedding containing information about every node and its corresponding neighbors. In order to gather information from neighbor nodes and their edges and enrich the representation of GNN, extensive works [14, 18, 25, 50] study more complex graph structures. GCN [18] and Graph Sage [14] use spectral and spatial convolutional aggregation respectively, in which spectral convolution utilizes Fourier frequency domain to calculate graph Laplacian eigenvalue decomposition and spatial convolution operates on adjacent neighbor nodes in the spatial domain. GGNN [25] proposes a gated graph neural network to improve long-term information dissemination. GAT [50] introduces the attention mechanism to acquire the hidden state of the node by adding attention to its neighbor nodes. Highway GCN [36] leverages skip connection to avoid introducing more noise from superimposing [24] on the network layer.

The previous trajectory prediction methods, *e.g.*, GCN and GAT, lack of a clear and proper distinction between heterogeneous nodes and homogeneous nodes, while our method takes large scale heterogeneous graph into account. In addition, most existing graph neural networks group heterogeneous nodes into a subgraph, which suffer from data imbalance and ineffective global information aggregation. In contrast, we utilize hierarchical graph attention to aggregate the information of large-scale heterogeneous nodes.

3. Our Method

In this section, we introduce our proposed UNIN, which aims to model interactions of heterogeneous traffic agents under the guidance of unlimited neighborhood interaction.

Given a succession of video frames of traffic scenarios over time $t \in \{1, 2, \dots, T_{\text{obs}}\}$, there are C categories with N agents. The goal of trajectory prediction is to predict the location of each traffic agent $i \in \{1, \dots, N\}$ within a future time horizon $t \in \{T_{obs+1}, T_{obs+2}, \dots, T_{\text{pred}}\}$. For a traffic agent $i \in \{1, \dots, N\}$ of category $c \in \{1, 2, \dots, C\}$, it is denoted as $V_t^i = \left(x_t^i, y_t^i, c\right)$, where $\left(x_t^i, y_t^i\right)$ is the location coordinate of traffic agent i at time step $t \in \{1, 2, \dots, T_{\text{pred}}\}$.

As discussed, the interactions in previous works are only considered between two traffic agents or in a local area, while an unlimited number of other agents may be simultaneously involved in an interaction regardless of their category. Additionally, most of the existing works neglect people's diverse reactions to heterogeneous agents, which is spontaneous in real traffic scenarios, is under-explored. To mitigate these limitations, we propose the Unlimited Neighborhood Interaction to capture the impact that all agents experience at the same time, and a hierarchical attention module to model heterogeneous interactions among traffic agents of different categories.

The overall framework of UNIN is illustrated in Figure 3. To aggregate information of agents involved in the same interaction, an interaction graph is built first to gather global interaction information. Subsequently, the Hierarchical Attention Module is used to obtain the category-category interaction and agent-agent interaction based on the global interaction information. Next, we introduce the Unlimited Neighborhood Module directly modeling interactions by pooling features among unlimited neighborhood agents. Finally, a heterogeneous graph convolution network and a temporal convolutional network are used to predict the parameters of a Gaussian Mixture Model for trajectory prediction.

3.1. Heterogeneous Graph Construction

There are agents in multiple categories in heterogeneous trajectory prediction, and thus we build a spatio-temporal-category graph $\mathcal{G}_{stc}$ to model them altogether, as shown in Figure 3, where every agent is regarded as a node and the interactions among agents are regarded as edges. To enhance the representations of category-category interaction, we also regard all agents with the same category as a category node:

$$\mathcal{G}_{stc} = (V_t^i, E_t^i, E_t^{ij}, S_t^c, D_t^{c_1,c_2}, D_t^c), \tag{1}$$

where $i \in \{1, ..., N\}, t \in \{1, \dots, T_{pred}\}, c = \{1, \dots, C\}$ represent the index of node, time step and category, respectively. $V_t^i = \{(x_t^i, y_t^i, c)\}$ represents the node i with category c at the time step t. $E_t^i = \{(V_t^i, V_{t+1}^i)\}$ is a temporal edge connecting node V_t^i and V_{t+1}^i. $E_t^{i,j} = \{(V_t^i, V_t^j)\}$ is a spatial edge connecting node V_t^i and V_t^j. $S_t^c = \{V_t^i \mid \forall i \in \{1, \dots, N\}\}$ is the category node with category c at time step t generated by the concatenation of all agents with category c at time step t, which represents the embedding, *i.e.*, the concatenation of the agents of category c. We consider the difference of various categories of agents, and project them with a common transformation matrix. Then we concatenate the agent features of the same category. $D_t^{c_1,c_2} = \{(S_t^{c_1}, S_t^{c_2}) \mid c_1, c_2 \in \{1, \dots, C\}\}$ is the spatial category edges connecting category node $S_t^{c_1}$ and $S_t^{c_2}$ at time step t. $D_t^c = \{\left(S_t^c, V_t^i\right)\}$ is the spatial category-agent edges connecting category node S_t^c and each spatial node V_t^i belonged to category c.

The built spatio-temporal-category graph $\mathcal{G}_{stc}$ includes not only the information of each agent, but also the information of each category. Therefore, we can leverage $\mathcal{G}_{stc}$ to build category-to-category and agent-to-agent interaction.

3.2. Hierarchical Graph Attention

The interaction among agents is an essential factor for trajectory prediction. Especially, the heterogeneous interaction is more complex due to diverse object categories compared with homogeneous interaction [33]. In traffic scenarios, traffic agents (pedestrians, drivers, bikers, etc.) tend to react differently according to the categories of agents they encounter because of the difference in social habits and experiences. Hence, the interaction between categories (*i.e.*, category-category interaction) is also an important factor affecting agent's trajectories.

In order to model the interaction among agents with multiple categories, we propose a Hierarchical Graph Attention module. It models the category-category interaction first, based on which the agent-agent interaction is modeled.

Category-Category Interaction. To build the interaction among categories, we obtain the category features of each category first on our built spatio-temporal-category graph, based on which the category-wise interaction weights are obtained through pooling operation.

In light of the imbalanced amount of agents in different scenarios, we employ a padding operation to align them to the same amount. Then, the embeddings h_t^c of each category are obtained by a linear projection, *i.e.*,

$$h_t^c = \phi\left(W_e, \Theta\left(S_t^c\right)\right), \tag{2}$$

where $\phi\left(\cdot,\cdot\right)$ denotes linear projection, S_t^c is the category node with category c at time step t, h_t^c is the embedding of category c at time step t, Θ is the padding operation, and W_e is the learnable weight of linear projection. The padding size equals to the largest number of nodes in the scenario for efficient computation. Padded convolution is also flexible for arbitrary number of agents, as convolution on 0 will not change the result.

After acquiring the embeddings of each category, the embeddings of any two categories are concatenated to obtain fused embeddings. Subsequently, the category-category attention scores A_t are generated by graph attention mechanism [51], as follows:

$$A_t^{c_1,c_2} = \delta\left(\mu_c \cdot \left(h_t^{c_1} \parallel h_t^{c_2}\right)\right), \tag{3}$$

where $A_t^{c_1,c_2}$ is the attention score vector of category c_1 to c_2 at time step t, μ_c denotes a learnable attention weight vector of category c used to adjust the weights among categories, $\delta(\cdot)$ denotes a non-linear activation function.

The attention score vector $A_t^{c_1,c_2}$ measures the interaction of one category to other categories. The category-category interaction aims to assist agent-agent interaction, and thus we only acquire an importance factor by pooling operation for each attention score vector $A_t^{c_1,c_2}$. We employ the max pooling(Υ) to choose the biggest value in $A_t^{c_1,c_2}$ as the importance factor $a_t^{c_1,c_2}$, *i.e.*,

$$a_t^{c_1,c_2} = \Upsilon\left(A_t^{c_1,c_2}\right). \tag{4}$$

After acquiring the importance factor between any two categories, the final category-category interaction $\mathrm{CI}_t^{c_1,c_2}$ is obtained by normalizing all the importance factors (which the number of categories is n):

$$\mathrm{CI}_t^{c_1,c_2} = \frac{\exp\left(a_t^{c_1,c_2}\right)}{\sum_{i,j\in n}\exp\left(a_t{}^{i,j}\right)}. \tag{5}$$

The weights of spatial category edges $D_t^{c1,c2}$ represent the category-category interaction, and thus we assign value to $D_t^{c1,c2}$ by the obtained interaction values.

Agent-Agent Interaction. Some related works [32] indicate the relative distance between agents is essential in some special scenarios. Therefore, we obtain the agent-agent interaction by a combination of learning-based method and distance-based method.

It is intuitive to define a weight that grows for approaching agents based on an assumption that agents are more susceptible to closer ones. Meanwhile, the attention mechanism ensures that far interacting agents can also be recognized by the model.

The distance-based method initializes the spatial edge E_t with the relative distance between the corresponding agents. Then, the normalized interaction matrices R_t is obtained by Laplace Transform [31] as follows:

$$\begin{aligned} &E_t^{i,j} = \begin{cases} 1/\|p_t^i - p_t^j\|_2, & \|p_t^i - p_t^j\|_2 \neq 0 \\ 0, & \text{Otherwise} \end{cases}, \\ &R_t = \Lambda_t^{-\frac{1}{2}} \hat{E}_t \Lambda_t^{-\frac{1}{2}}, \end{aligned} \tag{6}$$

where p_t^i, p_t^j is the location coordinates for agent i, j at time step t, $\hat{E}_t = E_t + I$, and Λ_t is the diagonal node degree matrix of E_t.

For the learning-based method, we need to fuse the features of all agents. Fortunately, the learned attention score vector A_t shown in Equation 3 already includes the required information, and thus we directly employ the learned A_t to obtain the agent-agent interaction ATT_t, *i.e.*,

$$\mathrm{ATT}_t = R_t \otimes A_t, \tag{7}$$

where operator $\otimes$ denotes dot-product operation.

3.3. Unlimited Neighborhood Interaction

In a real traffic scenario, interactions differ among the uncertain numbers of agents, *i.e.*, an agent could respond differently as the number of interacted agents varies. However, the existing graph attention mechanism [53] only computes the interaction between pair-wise agents because the inner-product is operated only between two vectors once. And graph convolutional network with many layers suffers from over-smoothing [9, 19]. According to our observation, GCN with one or two layers is optimal for our task. Hence, the learned agent-agent interaction ATT_t can not adaptively capture the interaction among the uncertain number of agents.

To mitigate this, we propose the Unlimited Neighborhood Interaction module to capture the information of all agents involved in a same interaction simultaneously. Note that all agents involved in an interaction are called "unlimited neighborhood", regardless of the numbers of agents. In particular, we employ an asymmetric convolution to obtain and aggregate the global interaction information on ATT_t, *i.e.*,

$$h_t = \delta(\mathrm{Conv1D}(\mathrm{ATT}_t)), \tag{8}$$

where δ is the non-linear activation function, and we use padding operation to ensure the output size the same as the input size.

The asymmetric convolution is computed repeatedly and thus the global spatial interaction information can be aggregated, meaning that all agents involved in an interaction are considered, regardless of the number of the agents. Because small asymmetrical kernel with padding captures implicit interactions, which are not limited by the number of agents or the range of area. It ensures any number of agents in a specific interaction can be considered, while a big symmetric kernel mixes different numbers/ranges of agents in different interactions.

The final interaction F_t is obtained through fusing unlimited neighborhood and category-category interaction:

$$F_t = \mathrm{CI}_t^{c_1,c_2} \otimes h_t. \tag{9}$$

3.4. Trajectory Prediction

After obtaining the final interaction F_t, we regard it as the adjacency matrix of the spatio-temporal-category graph and feed it in GCN, which is followed by a TCN to estimate the parameters of Gaussian Mixture Model. A residual connection is used in GCN, *i.e.*,

$$\begin{aligned} &H_t^{(l)} = \delta(H_t^{(l-1)} + F_t^{(l)} \cdot \mathrm{Conv}(H_t^{(l-1)})), \\ &HT = \mathrm{TCN}(H_t), \end{aligned} \tag{10}$$

where δ is a non-linear activation function, l is the index of layers of GCN, $H_t^0 = V_t$ represents the node of the graph, and HT is the output features of TCN. Thus we acquire the

Models	Argoverse		nuScenes		Avg		Apolloscape	
	ADE	FDE	ADE	FDE	ADE	FDE	WADE	WFDE
S-LSTM [1]	1.385	2.567	1.390	2.676	1.388	2.622	1.89	3.40
DESIRE* [21]	0.896	1.453	1.079	1.844	0.988	1.649	-	-
R2P2-MA [38]	1.108	1.771	1.179	2.194	1.144	1.983	-	-
CAM [34]	1.131	2.504	1.124	2.318	1.128	2.411	-	-
MFP [45]	1.399	2.684	1.301	2.740	1.350	2.712	-	-
MATFD [58]	1.344	2.484	1.261	2.538	1.303	2.511	-	-
MATFG* [58]	1.261	2.313	1.053	2.126	1.157	2.220	-	-
STGCNN [32]	1.305	2.344	1.274	2.198	1.289	2.371	-	-
StarNet [59]	-	-	-	-	-	-	1.343	2.498
TPNet [11]	-	-	-	-	-	-	1.281	1.910
NLNI (Ours)	**0.792**	**1.256**	**1.049**	**1.521**	**0.921**	**1.388**	**1.094**	**1.545**

Table 1. Comparison with other methods on dataset Argoverse, nuScenes and Apollscape in ADE and FDE metrics (the lower the better). All methods observe 2 seconds and predict the next 3 seconds of trajectories. Note that the Apolloscape dataset uses weighted ADE and FDE metric, *i.e*., the weights of vehicles, pedestrians and cyclists are assigned as 0.20, 0.58 and 0.22, respectively. The methods marked by "*" use additional scene context. Our UNIN significantly outperforms the state-of-the-art works.

Datasets	Models										
	S-LSTM [1]	MATF [58]	DESIRE [21]	NRI [17]	S-GAN [12]	SOPHIE [40]	Traject++ [41]	STGCN [32]	SIMAUG* [32]	STGAT [20]	**Ours**
SDD	31.2 / 57	22.6 / 33.5	19.3 / 34.1	25.6 / 40.3	27.3 / 41.4	16.3 / 29.4	19.3 / 32.7	20.6 / 33.1	**15.7** / 30.2	18.8 / 31.3	**15.9 / 26.3**

Table 2. Comparison with the previous approaches on the SDD benchmark dataset, which mainly contains the trajectories of pedestrian. The performance is evaluated in ADE/FDE metrics (the lower the better). The approach marked by "*" uses additional simulation data.

collective interaction information from both the space and the time information.

Loss Function. Since the traffic agents of different categories have their own unique movement pattern, *e.g*., a certain velocity range, front and rear distance to another object, we assume that the trajectory coordinates (x_t^i, y_t^i) of traffic agents i follow a Gaussian Mixture Model [10]. Hence, our model is trained by minimizing the negative log-likelihood loss as follows:

$$\mathrm{L^i} = -\sum_{t=T_{obs}+1}^{T_{pred}} \log \sum_{k=1}^{K} \pi_k N\left((x_t^i, y_t^i) \mid \hat{\mu}_n^t, \hat{\sigma}_n^t, \hat{\rho}_n^t\right), \tag{11}$$

where $\hat{\mu}_n^t$ is the mean, $\hat{\sigma}_n^t$ is the standard deviation, $\hat{\rho}_n^t$ is the correlation co-efficient, and π_k is the weight factor of the k-th Gaussian distribution.

4. Experiments

Datasets. Some datasets focus on homogeneous trajectories, and contain fewer traffic scenes, *e.g*., ETH [35] and UCY [22], which only label pedestrian trajectory within three scenes. However, there are often diverse categories in the real scenario, and thus we train and evaluate our model on more complex datasets, including Stanford Drone Dataset(SDD) [39], nuScenes [5], Argoverse [8] and Apolloscape [29], which are widely used in heterogeneous trajectory prediction with diverse categories and rich traffic scenes. The SDD consists of eight unique scenes on the university campus, more than 100 static scenes, $19K$ traffic agents of 6 categories, and approximately $40K$ interactions. The nuScenes, Argoverse and Apolloscape are large-scale trajectory datasets for urban streets with dense traffic in highly complicated situations. Besides, trajectories in them are collected through an in-vehicle camera so that they have more different scenarios.

We follow the existing works, observing 3.2 seconds of trajectories while predicting the next 4.8 seconds in Stanford Drone Dataset, and observing 2 seconds while predicting the next 3 seconds in nuScenes, Argoverse, and Apolloscae datasets.

Evaluation Metrics. We follow existing works [56] and employ two common metrics to evaluate the performance: Average Displacement Error (ADE) and the Final Displacement Error (FDE), which are defined as follows:

$$\begin{aligned} \mathrm{ADE} &= \frac{\sum_{n \in N} \sum_{t \in T_p} \|\hat{p}_\mathrm{t}^n - p_\mathrm{t}^n\|_2}{N \times T_p}, \\ \mathrm{FDE} &= \frac{\sum_{n \in N} \|\hat{p}_T^n - p_T^n\|_2}{N \times T_p}, \end{aligned} \tag{12}$$

where ADE measures the average L2 distance between ground truth and our predicted future positions over all time steps, while FDE measures the L2 distance between our predicted final destination and the true final destination.

4.1. Implementation Details

In the Hierarchical Attention Module, the embedding dimension of one category is set to 8 and the output size after padding is equal to the largest number of nodes in the scenario. In the Unlimited Neighborhood Module, the kernel-size k of the convolution(UNIConv) is fixed at 3. We train our model with SGD, and the learning rate is set to

Dataset	MLP		CNN		NLIN (Ours)
	w/o HGA	w/o UNI	w/o HGA	w/o UNI	
Apolloscape	1.460/1.794	1.576/1.843	1.837/2.014	1.792/1.955	**1.094/1.545**
nuScenes	1.613/1.969	1.547/1.728	1.763/1.982	1.701/1.934	**1.049/1.521**

Table 3. The ablation study of each component (Using MLP/CNN to replace each component). UNIN (Ours) combines with each component.

UNIConv Size	1	2	3	5	10
ADE	1.179	**0.921**	**0.998**	1.247	2.691
FDE	1.632	**1.388**	**1.323**	1.766	3.515

Table 4. Ablation study of kernel size for Unlimited Neighborhood convolution.

0.005, which decays by a factor 0.2 after every 10 epochs. The weighted factor of GMM loss is acquired from the Hierarchical Attention Module and the approximate ratio of the categories in scenes. We train our model on an RTX2080Ti GPU for up to 50 epochs. And we use a dataset split of 60%, 20%, 20% for training, validation and testing, respectively. The complete code will be published once upon acceptance.

4.2. Quantitative Evaluation

Table 1 and Table 2 show the comparison of our method against state-of-the-art approaches, including Social LSTM [1], Social GAN [12], STGAT [20], Social STGCNN [32], Trajectron++ [41], NRI [17], SoPhie [40], MATF [58], DESIRE [21], SimAug [26], P2P2-MA [38], CAM [34], MFP [45], StarNet [59] and TPNet [11]. Overall, our method significantly outperforms all compared methods on all datasets according to the tables. Particularly, our UNIN surpasses the DESIRE (the second best) by 2.7% on average in ADE and 13.85% on average in FDE for nuScenes, Argoverse and Apolloscape. Meanwhile, our method achieves a performance improvement by 10.5% on average in FDE for SDD dataset. The underlying reason is that our method can model the collective interaction among the agents involved in the same interaction simultaneously. Meanwhile, the Hierarchical Attention enhances the agent-agent interaction based on category-category interaction.

nuScenes, Argoverse and Apolloscape. Our UNIN outperforms all the competing methods on the three datasets. The nuScense, Argoverse and Apolloscape are multi-category mixed datasets with a majority of vehicles. Compared with the RNN-based method, such as S-LSTM [1], our method surpasses it by $42.8\%/51.1\%$ in FDE/ADE metrics. We speculate that S-LSTM employs a pooling mechanism to aggregate local agents' states, while it does not take the long-range interaction into account. In addition, our method also outperforms the Graph-based methods, *e.g.*, S-STGCNN [32], by $28.5\%/41.3\%$ in FDE/ADE metrics. We speculate it takes the long-range interaction into account but the interactions are only modeled between pairwise agents. Interestingly, our method outperforms the methods employed scene context, such as DESIRE [21] and MATFG [58]. Both of them employ a LSTM to model each agent and fuse the interaction with in a local area, while our model considers the unlimited neighborhood, which is not limited by the number of agents and the range of interaction. Thus, our method can capture more global and local detail information to improve the accuracy of future trajectories.

Stanford Drone Dataset. Stanford Drone Dataset(SDD) is a multi-category mixed dataset including pedestrians, bicyclists, skateboarders, carts, cars and buses with a majority of pedestrians. Our method outperforms the methods modeling the interaction in a local area, such as S-LSTM [1] (ours achieves $49\%/52\%$ better on average in ADE/FDE) and S-GAN [12] (ours achieves $41.7\%/36.5\%$ better on average in FDE/ADE). We speculate the reason is they employ a pooling mechanism to aggregate the local agent's interaction states, while our method employs an unlimited interaction capable of capturing the information of flexible interactions. In addition, our method is better than the graph-based methods, such as STGCNN [32], by $22.8\%/20.5\%$ average. Moreover, our method is slightly outperformed by SIMAUG [26] in ADE metric, possibly due to SIMAUG uses extra 3D simulation data for training, leading to more robust representations. We also evaluate the data efficiency and generalization ability of our model, please refer to supplemental material for detail.

4.3. Qualitative Evaluation

We further study the ability of our method to model interactions of large-scale traffic agents with multiple categories. As discussed previously, there are often interactions with large numbers of agents and uncertain distances between them in real traffic scenarios. And agents often adopt different strategies when interacting with different categories of traffic participants. We illustrate some qualitative evaluation results in Figure 4. Overall, our predicted trajectory distributions are in line with the ground truth trajectories. Result (a) is the long time trajectories from the beginning time instant to the last time instant, which demonstrates the great prediction accuracy achieved by our method. Result (b) shows a single traffic agent that is turning. As expected, our model captures the agent's tendency of turning. Result (c) shows our method successfully predicts the trajectory when two agents are going in parallel orienting towards the same direction, which means our method does not appear to be over-fitting. In (d), two non-adjacent agents interact with each other rather than with another closest agent to them. Our method leverages the UNI to capture the long-range

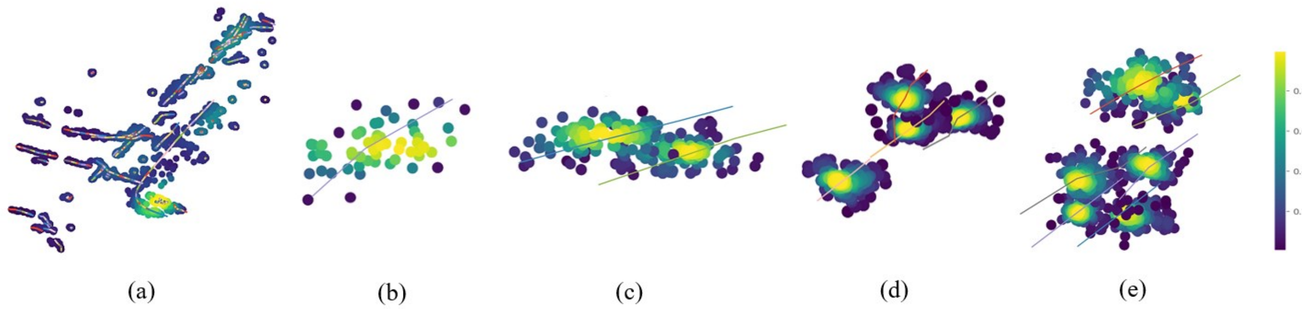

Figure 4. **Visualization of predicted trajectory distribution.** Each line represents the ground-truth trajectory of an agent. The colored dots represent our predicted trajectory distribution, and different colors represent different densities of our predicted distribution, where yellow represents the most likely trajectory distribution. (a) shows the overall trajectories in the whole scene at all of the time instants. (b) shows that we successfully predict a turning agent. (c) shows that we successfully predict two agents going in parallel to the same direction. (d) shows that we successfully predict two agents separated by another interacting and avoiding each other. (e) shows that we successfully predict the possible trajectory of a group of agents after collective interaction. All results are randomly sampled from the nuScenes dataset.

interaction, successfully predicting that relatively distant agents interacting and the subsequent trajectories. Result (e) shows the collective interaction involving a group of agents belonging to different categories. Our method successfully predicts the possible trajectory of them with a complex interaction. And our predicted trajectory distributions show that the agents of different categories react differently when interacting with a specific agent, which demonstrates the efficiency of our HGA. We also visualize the relation between category attention and agent attention in supplemental material.

4.4. Ablation Study

We study the contribution of each component in our model as shown in Table 3. In addition, we set different values of kernel size of Unlimited Neighborhood Interaction to find the empirical optimal value, as shown in Table 4.

Contribution of Each Component. As illustrated in Table 3, we evaluate two variants of our method: **(1)** UNIN w/o HGA, which means the category-to-category attention is replaced with CNN/MLP and only the agents-to-agents interaction is kept; **(2)** UNIN w/o UNI, which means the unlimited neighborhood interaction is replaced with CNN/MLP. According to the results, removing any component will lead to a large performance drop. Particularly, the results of UNIN w/o HGA show a performance reduction by $28.3\%/16.4\%$ in ADE/FDE metrics, reflecting the effectiveness of hierarchical attention. The results of UNIN w/o UNI shows a performance degradation by $30.1\%/14.2\%$ in ADE/FDE metrics, which validates the contribution of unlimited neighborhood interaction.

Optimal Kernel Size. As shown in Table 4, the optimal value of the kernel size of Unlimited Neighborhood Interaction convolution is 2 in ADE metric, and 3 in FDE metric. From the table, a larger kernel size is unhelpful. The convolution with kernel size 2 and 3 are the best performing settings to capture the relation among group agents.

5. Conclusion

To capture the interaction information with varying numbers of agents from an uncertain distance, we present an Unlimited Neighborhood Interaction Network to predict trajectories in multiple categories. An Unlimited Neighborhood Interaction Module generates the interaction with all of the agents involved in the interaction simultaneously. A Hierarchical Graph Attention module is designed to acquire the category-to-category interaction and agent-to-agent interaction, where the former one is used to enhance the representation of agent-to-agent interaction. Extensive quantitative evaluations show our method achieves state-of-the-art performance, even outperforming methods leveraging additional scene context. Qualitative evaluations illustrate the advantage of our method when predicting heterogeneous trajectories in dense and complex traffic scenarios.

Acknowledgment

This work was supported partly by National Key R&D Program of China Grant 2018AAA0101400, NSFC Grants 62088102, 61976171, 61773312 and 62106192, the General Program of China Postdoctoral Science Foundation under Grant No. 2020M683490, Young Elite Scientists Sponsorship Program by CAST Grant 2018QNRC001, and the Youth program of Shanxi Natural Science Foundation under Grant No. 2021JQ-054.

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 *CVPR*, pages 961–971, 2016. 1, 2, 6, 7

[2] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. An empirical

evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018. 2

[3] Huikun Bi, Zhong Fang, Tianlu Mao, Zhaoqi Wang, and Zhigang Deng. Joint prediction for kinematic trajectories in vehicle-pedestrian-mixed scenes. In *ICCV*, pages 10383–10392, 2019. 1, 3

[4] Niccoló Bisagno, Bo Zhang, and Nicola Conci. Group lstm: Group trajectory prediction in crowded scenarios. In *ECCVW*, 2018. 1, 2

[5] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, pages 11621–11631, 2020. 6

[6] Sergio Casas, Cole Gulino, Simon Suo, Katie Luo, Renjie Liao, and Raquel Urtasun. Implicit latent variable model for scene-consistent motion forecasting. In *ECCV*, pages 624–641, 2020. 3

[7] Rohan Chandra, Tianrui Guan, Srujan Panuganti, Trisha Mittal, Uttaran Bhattacharya, Aniket Bera, and Dinesh Manocha. Forecasting trajectory and behavior of road-agents using spectral clustering in graph-lstms. *RAL*, 5(3):4882–4890, 2020. 1, 2

[8] 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 *CVPR*, pages 8748–8757, 2019. 6

[9] Nima Dehmamy et al. Understanding the representation power of graph neural networks in learning graph topology. In *NeurIPS*, 2019. 1, 5

[10] W Dong and M Zhou. Gaussian classifier-based evolutionary strategy for multimodal optimization. *NNLS*, 25(6):1200–1216, 2017. 6

[11] Liangji Fang, Qinhong Jiang, Jianping Shi, and Bolei Zhou. Tpnet: Trajectory proposal network for motion prediction. In *CVPR*, pages 6797–6806, 2020. 6, 7

[12] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *CVPR*, pages 2255–2264, 2018. 1, 2, 6, 7

[13] S. Haddad and S. Lam. Self-growing spatial graph networks for pedestrian trajectory prediction. In *WACV*, pages 1140–1148, 2020. 1

[14] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *NeurIPS*, pages 1024–1034, 2017. 3

[15] Irtiza Hasan, Francesco Setti, Theodore Tsesmelis, Alessio Del Bue, Marco Cristani, and Fabio Galasso. " seeing is believing": Pedestrian trajectory forecasting using visual frustum of attention. In *WACV*, pages 1178–1185, 2018. 3

[16] Dirk Helbing and Peter Molnar. Social force model for pedestrian dynamics. *Physical review E*, 51(5):4282, 1995. 2

[17] Thomas Kipf, Ethan Fetaya, Kuan-Chieh Wang, Max Welling, and Richard Zemel. Neural relational inference for interacting systems. In *ICML*, pages 2688–2697, 2018. 6, 7

[18] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2016. 3

[19] Johannes Klicpera et al. Predict then Propagate: Graph Neural Networks meet Personalized PageRank. In *ICLR*, 2018. 1, 5

[20] Vineet Kosaraju, Amir Sadeghian, Roberto Martín-Martín, Ian Reid, Hamid Rezatofighi, and Silvio Savarese. Social-bigat: Multimodal trajectory forecasting using bicycle-gan and graph attention networks. In *NeurIPS*, pages 137–146, 2019. 2, 6, 7

[21] 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 *CVPR*, pages 336–345, 2017. 6, 7

[22] Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. Crowds by example. In *CGF*, pages 655–664, 2007. 6

[23] Jiachen Li, Hengbo Ma, and Masayoshi Tomizuka. Interaction-aware multi-agent tracking and probabilistic behavior prediction via adversarial learning. In *ICRA*, pages 6658–6664, 2019. 1

[24] Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *AAAI*, number 1, 2018. 3

[25] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated graph sequence neural networks. In *ICLR*, 2016. 3

[26] Junwei Liang, Lu Jiang, and Alexander Hauptmann. Simaug: Learning robust representations from simulation for trajectory prediction. In *ECCV*, pages 275–292, 2020. 1, 7

[27] Junwei Liang, Lu Jiang, Kevin Murphy, Ting Yu, and Alexander Hauptmann. The garden of forking paths: Towards multi-future trajectory prediction. In *CVPR*, pages 10508–10518, 2020. 1, 3

[28] Ming Liang, Bin Yang, Rui Hu, Yun Chen, Renjie Liao, Song Feng, and Raquel Urtasun. Learning lane graph representations for motion forecasting. In *ECCV*, pages 541–556, 2020. 3

[29] Yuexin Ma, Xinge Zhu, Sibo Zhang, Ruigang Yang, Wenping Wang, and Dinesh Manocha. Trafficpredict: Trajectory prediction for heterogeneous traffic-agents. In *AAAI*, number 01, pages 6120–6127, 2019. 6

[30] Karttikeya Mangalam, Harshayu Girase, Shreyas Agarwal, Kuan-Hui Lee, Ehsan Adeli, Jitendra Malik, and Adrien Gaidon. It is not the journey but the destination: Endpoint conditioned trajectory prediction. In *ECCV*, pages 759–776, 2020. 3

[31] Naoki Masuda and Luis E. C. Rocha. A gillespie algorithm for non-markovian stochastic processes: Laplace transform approach. *Siam Review*, 60(1):95–115, 2017. 5

[32] Abduallah Mohamed, Kun Qian, Mohamed Elhoseiny, and Christian Claudel. Social-stgcnn: A social spatio-temporal graph convolutional neural network for human trajectory prediction. In *CVPR*, pages 14424–14432, 2020. 1, 2, 5, 6, 7

[33] Z. Niu, M. Zhou, L. Wang, X. Gao, and G. Hua. Hierarchical multimodal lstm for dense visual-semantic embedding. In *ICCV*, pages 1899–1907, 2017. 4

[34] Seong Hyeon Park, Gyubok Lee, Jimin Seo, Manoj Bhat, Minseok Kang, Jonathan Francis, Ashwin Jadhav, Paul Pu Liang, and Louis-Philippe Morency. Diverse and admissible trajectory forecasting through multimodal context understand-

ing. In *ECCV*, pages 282–298, 2020. 3, 6, 7

[35] Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *ICCV*, pages 261–268, 2009. 6

[36] Trevor Rahimi, Afshinand Cohn and Timothy Baldwin. Semi-supervised user geolocation via graph convolutional networks. In *ACL*, 2018. 3

[37] Douglas A Reynolds. Gaussian mixture models. *Encyclopedia of biometrics*, 741:659–663, 2009. 2

[38] Rhinehart, N. Kitani, and P K.M. Vernaza. R2p2: A reparameterized pushforward policy for diverse, precise generative path forecasting. In *ECCV*, pages 772–788, 2018. 6, 7

[39] Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Learning social etiquette: Human trajectory understanding in crowded scenes. In *ECCV*, pages 549–565, 2016. 6

[40] 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 *CVPR*, pages 1349–1358, 2019. 3, 6, 7

[41] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. *arXiv preprint arXiv:2001.03093*, 2020. 6, 7

[42] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *Neural Networks*, 20(1):61–80, 2008. 3

[43] Liushuai Shi, Le Wang, Chengjiang Long, Sanping Zhou, Mo Zhou, Zhenxing Niu, and Gang Hua. Sgcn: Sparse graph convolution network for pedestrian trajectory prediction. In *CVPR*, 2021. 2

[44] Amit Surana and Kunal Srivastava. Bayesian nonparametric inverse reinforcement learning for switched markov decision processes. In *ICMLA*, pages 47–54, 2014. 2

[45] Yichuan Charlie Tang and Ruslan Salakhutdinov. Multiple futures prediction. In *NeurIPS*, 2020. 6, 7

[46] Chaofan Tao, Qinhong Jiang, Lixin Duan, and Pingss Luo. Dynamic and static context-aware lstm for multi-agent motion prediction. In *ECCV*, pages 547–563, 2020. 3

[47] Meng Keat Christopher Tay and Christian Laugier. Modelling smooth paths using gaussian processes. In *FSR*, pages 381–390, 2008. 2

[48] Adrien Treuille, Seth Cooper, and Zoran Popović. Continuum crowds. *TOG*, 25(3):1160–1168, 2006. 2

[49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017. 3

[50] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2018. 2, 3

[51] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *stat*, 1050:4, 2018. 4

[52] Jack M Wang, David J Fleet, and Aaron Hertzmann. Gaussian process dynamical models for human motion. *PAMI*, 30(2):283–298, 2007. 2

[53] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. Heterogeneous graph attention network. In *WWW*, pages 2022–2032, 2019. 5

[54] Xiaogang Wang, Xiaoxu Ma, and W Eric L Grimson. Unsupervised activity perception in crowded and complicated scenes using hierarchical bayesian models. *PAMI*, 31(3):539–555, 2008. 2

[55] Guotao Xie, Hongbo Gao, Lijun Qian, Bin Huang, Keqiang Li, and Jianqiang Wang. Vehicle trajectory prediction by integrating physics-and maneuver-based approaches using interactive multiple models. *Industrial Electronics*, 65(7):5999–6008, 2017. 3

[56] Cunjun Yu, Xiao Ma, Jiawei Ren, Haiyu Zhao, and Shuai Yi. Spatio-temporal graph transformer networks for pedestrian trajectory prediction. In *ECCV*, pages 507–523, 2020. 1, 2, 6

[57] Pu Zhang, Wanli Ouyang, Pengfei Zhang, Jianru Xue, and Nanning Zheng. Sr-lstm: State refinement for lstm towards pedestrian trajectory prediction. In *CVPR*, pages 12085–12094, 2019. 1, 2

[58] Tianyang Zhao, Yifei Xu, Mathew Monfort, Wongun Choi, Chris Baker, Yibiao Zhao, Yizhou Wang, and Ying Nian Wu. Multi-agent tensor fusion for contextual trajectory prediction. In *CVPR*, pages 12126–12134, 2019. 6, 7

[59] Yanliang Zhu, Deheng Qian, Dongchun Ren, and Huaxia Xia. Starnet: Pedestrian trajectory prediction using deep neural network in star topology. In *IROS*, pages 8075–8080, 2019. 6, 7