Higher-order Relational Reasoning for Pedestrian Trajectory Prediction

Sungjune Kim[1], Hyung-gun Chi[2], Hyerin Lim[3], Karthik Ramani[4], Jinkyu Kim[1*], and Sangpil Kim[1*]

[1]Korea University, [2]Hanwha Vision America, [3]Hyundai Motor Company, [4]Purdue University

Abstract

Social relations have substantial impacts on the potential trajectories of each individual. Modeling these dynamics has been a central solution for more precise and accurate trajectory forecasting. However, previous works ignore the importance of 'social depth', meaning the influences flowing from different degrees of social relations. In this work, we propose HighGraph, a graph-based pedestrian relational reasoning method that captures the higher-order dynamics of social interactions. First, we construct a collision-aware relation graph based on the agents' observed trajectories. Upon this graph structure, we build our core module that aggregates the agent features from diverse social distances. As a result, the network is able to model complex social relations, thereby yielding more accurate and socially acceptable trajectories. Our HighGraph is a plug-and-play module that can be easily applied to any current trajectory predictors. Extensive experiments with ETH/UCY and SDD datasets demonstrate that our HighGraph noticeably improves the previous state-of-the-art baselines both quantitatively and qualitatively.

1. Introduction

Human trajectory forecasting is vital for designing safer engineering applications such as urban planning [24] and autonomous vehicles [6, 8, 27]. However, human motion is easily susceptible to heterogeneous external forces, and handling these factors still remains a challenge. Undoubtedly, one of the most dominant forces that affect human behavior in a crowded scene is a 'social force', and many previous works attempt to model this by incorporating inter-agent relations for trajectory prediction.

Early works demonstrate the importance of modeling social relations by extracting pair-wise interaction features. For example, Social-LSTM [2] and Social-GAN[9] develop pooling mechanisms that encode pair-wise attractions. Social-STGCNN [22] formulate inter-agent relations as a graph and perform graph convolution to aggregate the features of neighboring pairs. However, considering the 'flocking behavior' of humans, these methods are unavailable to capture the scaled interactions. Therefore, recent works suggest group-wise relational reasoning for a more comprehensive understanding of human behaviors [17, 36]. These methods construct hypergraphs that encode social behaviors in diverse group sizes. Increasing the topological magnitude of the relations offers a wider perception of the scenes. Nevertheless, directly pairing each neighbor or speculating their behaviors as a group may overlook the importance of indirect higher-order influences of each agent.

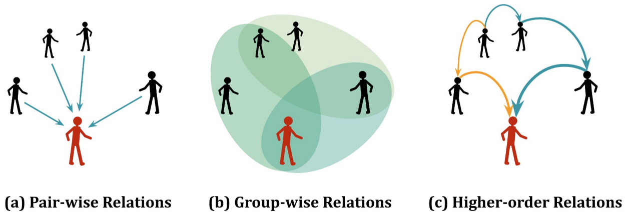


Figure 1. Example of different types of social relations. Higher-order relations contrast from pair-wise and group-wise relations in that they capture the indirect and long-ranging influences of socially distant interactions. To the best of our knowledge, our work is the first to consider higher-order social relations for pedestrian trajectory prediction. Best viewed in color.

Social relations are in-depth. Human decisions are affected not only by direct and nearing factors but also by socially distant ones. A certain pair-wise interaction between the local neighbors influences their future behaviors, and the decision they make may further affect the future behaviors of the third person in the distance, and so on. This chain-rule effect can be easily found in an urban crowded scene. For example, when two acquaintances walk toward each other, the relationship affects the path of those whose original destination is near the meeting point of the two. This trivial modification of the trajectories may further escalate to influence the following pedestrians' motions.

Despite its massive impact on future behaviors, incorporating higher-order social relations for trajectory forecasting has not well been explored. In this work, for the first time in the literature, we propose a novel method that models the higher-order dynamics of social interactions. Our work emphasizes the substantial influences of indirect interactions

[*] Corresponding authors ({jinkyukim, spk7}@korea.ac.kr)

of each individual, which differs from existing group-wise relations where the social scale was the main consideration. The comparison is visualized in Figure 1.

We develop HighGraph, a graph-based pedestrian relational reasoning method that infers higher-order relations in a crowded scene. First, we devise a new kernel function to construct a graph adjacency based on the observed social dynamics. Specifically, we update the potential collision point of the pedestrians for each time frame and provide weights based on the distance from the current position to the potential collision point. Then, based on this adjacency, we build our core module. We leverage the relational reasoning power of graph convolutional networks (GCNs)[14], but further modify in a way that aggregates the combinations of the features from diverse social degrees. The consecutive powers of the adjacency matrix enable linear fusion of the features with adjustable observation depth size. This results in more complex social understanding by yielding more accurate and socially acceptable outputs.

Our HighGraph is a plug-and-play module that can be attached to any existing predictors to improve its performance. We show how to plug our HighGraph into diverse models ranging from conventional recurrent models to recent conditional variational autoencoder (CVAE) [31] based models. With ETH/UCY and SDD datasets, we apply our HighGraph to previous state-of-the-art methods and show noticeable improvements both quantitatively and qualitatively. This demonstrates the importance of higher-order relational reasoning for pedestrian trajectory prediction.

The main contributions of this work are as follows:

- To the best of our knowledge, our work is the first to reason about indirect higher-order social relations of each individual for trajectory prediction.
- We develop HighGraph, a graph-based higher-order relational reasoning module based on a novel collision-aware kernel function.
- Our HighGraph module is designed to be a plug-and-play that can seamlessly integrate with any existing state-of-the-art methods, offering a simple and effective way to enhance their performance.

2. Related Work

2.1. Social Relations for Trajectory Prediction

Social relational reasoning is crucial for understanding the complex consequences of human behaviors [19, 29]. Conventional methods approach this in a statistical manner by inferring probabilistic dependencies between variables [7]. With the success of deep neural networks in many domains, many works exploit high-dimensional latent features, which enable them to model more complex social relations. Social-LSTM [2] and Social-GAN [9] design pooling mechanisms that encode pair-wise attractions. Social-Attention [32] assigns relative importance to each pair with an attention module. Several works, such as [12, 15, 22] utilize graph structures to perform neighborhood aggregation between the paired entities. However, these methods fail to view the social scene from a broader perspective (e.g. group-wise interaction). Traditional group-aware methods use clustering algorithms to group the agent behaviors [4]. Recent deep learning-based works such as [17, 36] propose a group-wise relational reasoning method with multiscale hypergraphs. Our work contrasts with the previous social models in that the main focus of this work is to capture the higher-order influences from socially distant interactions. which has never been explored in the literature.

2.2. Graph Convolutional Networks

Our HighGraph is built upon the feature aggregation algorithm of GCNs [14], which is widely used to learn the representations of relational data. Its idea is to aggregate the messages of the neighboring nodes in a non-linear manner to update the status of the target nodes. Due to its efficiency, GCN is utilized in diverse domains such as recommendation [10, 13, 33, 34], human action recognition [5, 20, 39], visual question answering [18, 23], and many more. However, the vanilla GCN is designed to interact only with the local neighboring nodes, limiting itself to observing deeper and long-ranging connections. MixHop [1] introduces the higher-order GCN by using the powers of the adjacency matrix. This motivates many studies to reason about deeper graphical relations of the data [35, 40]. Our work further enhances this concept to be applicable to the trajectory prediction task. First, we construct a collision-aware relational graph. The powers of its adjacency represent the complex social influences from both local and distant neighbors. Then, we add combination layers at the end of each higher-order graph convolution, which compiles the social features from diverse degrees as one representative feature. Further details will be explained in the following sections.

3. Preliminaries

3.1. Trajectory Prediction

The objective of the task is to predict the future trajectories of the agents given their historical observations. Formally, let $X \in \mathbb{R}^{h\times N\times 2}$ be the past h frames of observed trajectory coordinates of N agents, and $Y \in \mathbb{R}^{f\times N\times 2}$ be the f frames of ground truth trajectories continuing from the observed sequences. With given observations, the deterministic models predict the 2D coordinate values, whereas the probabilistic models estimate the conditional distribution $p(Y|X;\Theta)$, where Θ denotes the model parameters.

3.2. Graph Convolutional Networks

For a given graph $G = (V, E)$ with V and E as a set of nodes and edges, respectively, the message passing in layer l of vanilla GCN can be formulated as follows:

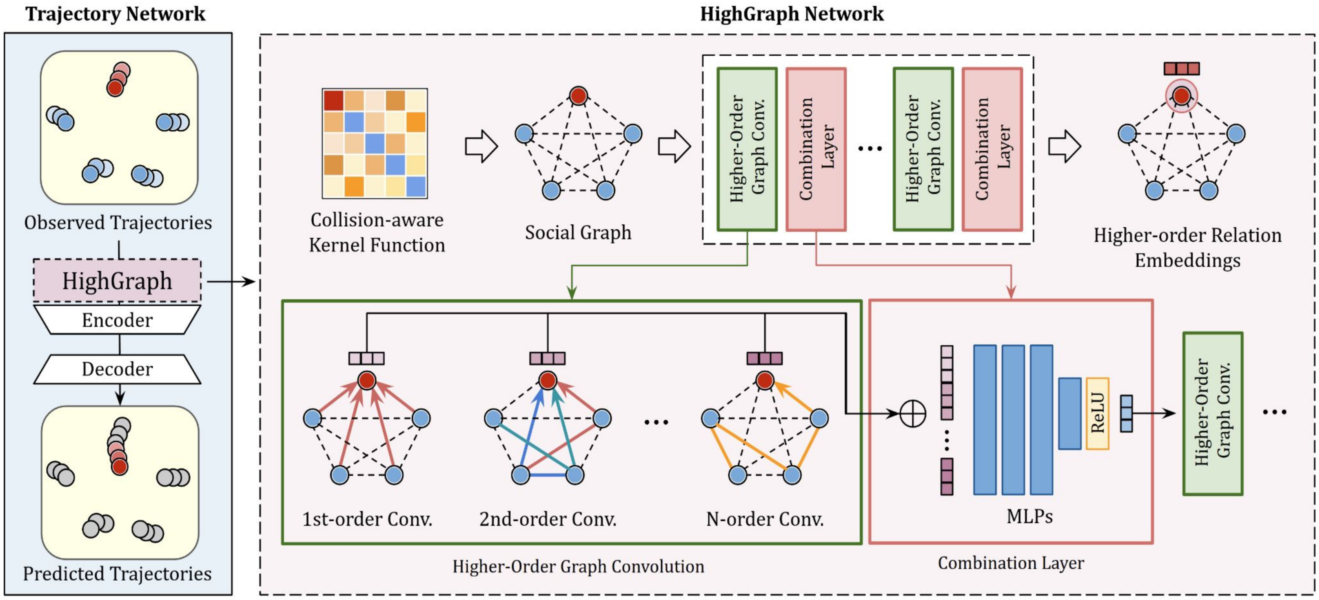


Figure 2. The higher-order relational reasoning architecture of HighGraph. First, we define the social interactions with the proposed collision-aware kernel function. Then HighGraph recursively aggregates the social features from diverse distances and combines them to obtain complex social influences. These fruitful feature embeddings can be used in any existing trajectory network. Best viewed in color.

$$\mathbf{H}^{(l+1)} = \sigma(\hat{\mathbf{A}}\mathbf{H}^{(l)}\mathbf{W}^{(l)}), \tag{1}$$

where $\mathbf{H} \in \mathbb{R}^{|V|\times d}$ is a d-dimensional feature matrix of the nodes, $\hat{\mathbf{A}} = \mathbf{D}^{-\frac{1}{2}}(\mathbf{A}+\mathbf{I})\mathbf{D}^{-\frac{1}{2}}$ is a symmetrically normalized adjacency matrix of $\mathbf{A}$, $\mathbf{W} \in \mathbb{R}^{d\times d'}$ is a trainable weight matrix and $\sigma(\cdot)$ is an activation function. Our HighGraph module is built upon this neighborhood aggregation architecture. With our proposed collision-aware kernel function and the higher-order feature combination technique, HighGraph is able to infer deeper and long-ranging social relations.

4. HighGraph

In this section, we explain our proposed HighGraph in detail. First, we introduce a novel collision-aware graph kernel function that quantifies the strength of each pair-wise interaction based on the agent dynamics. Then, our HighGraph performs higher-order graph convolution upon the constructed graph. The learned embeddings of HighGraph can be plugged into the encoders of any existing prediction methods and enhance the performance. The overall architecture is illustrated in Figure 2.

4.1. Collision-aware Kernel Function

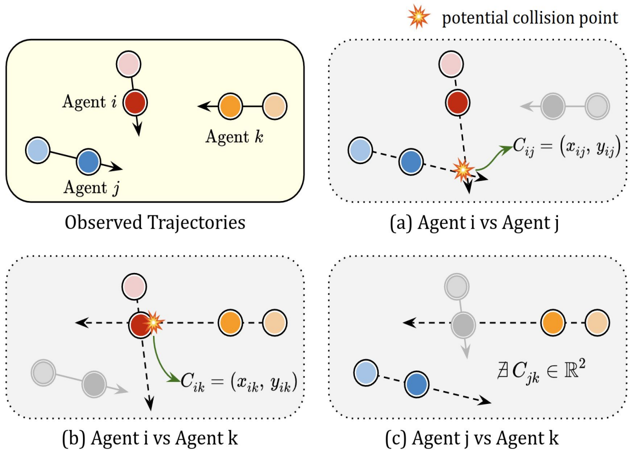
Pedestrians in a crowd walk in a way that avoids collision with each other. Considering this social prior, we come up with a kernel function that assigns interaction weight for each agent pair. First we construct a social graph G for each timestamp, with its adjacency matrix $\mathbf{A} = \{a_{ij}|\forall i,j \in \{1,...,N\}\}$, where the initial value of a_{ij} is set as 0 if $i=j$, and 1 if $i \neq j$. Then, our collision-aware kernel function takes the elements of the adjacency matrix as input and assigns weights for each pair. Specifically, we find a potential collision point C for each pair. This is computed by the intersection of two half-lines starting from each agent's previous location coordinate P and passing the point of the current location Q. Formally, for an agent i, its trajectory half-line can be denoted as $\overrightarrow{P_iQ_i}$. Then the potential collision point between two agent pair $\{i,j\}$ can be described as

$$C_{ij} = (x_{ij}, y_{ij}) = \overrightarrow{P_iQ_i} \cap \overrightarrow{P_jQ_j}. \tag{2}$$

Next, we calculate the distance from each agent's current locations Q_i and Q_j to their potential collision point, where our kernel function assigns higher weights for the interactions with imminent potential collision. The overall equation of our collision-aware kernel function is as follows:

$$g(a_{ij}) = \begin{cases} 1/\Big(d_{ij}\sum_{e\in\{i,j\}} \|C_{ij}-Q_e\|_2\Big), & \text{if } \exists C_{ij} \in \mathbb{R}^2 \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

where $d_{ij} = \|Q_i - Q_j\|_2$ is the distance between the agents' current location. We use this to scale the proximity of the interaction. Finally, we normalize the weighted adjacency matrix as described in section 3.2 and obtain $\hat{\mathbf{A}}$. We provide a visualization of the method in Figure 3 for better understanding. Further details of the kernel method will be explained in the supplementary material.


potential collision point
Agent i
Agent k
Agent j
$C_{ij} = (x_{ij}, y_{ij})$
$C_{ik} = (x_{ik}, y_{ik})$
$\nexists C_{jk} \in \mathbb{R}^2$

Observed Trajectories

(a) Agent i vs Agent j

(b) Agent i vs Agent k

(c) Agent j vs Agent k

Figure 3. Visualization of social interaction modeling of our collision-aware kernel function. Case (a) and (b) describe the situation where two agents are expected to collide at a certain point. We assign weight values based on the distance between the potential collision point to each agent and the distance between each other. Case (c) is an example where no collision is expected. We assign 0 for the kernel weight in such a case. Best viewed in color.

4.2. HighGraph Layers

Higher-order Graph Convolution. Powering the adjacency matrix opens up the path to the nodes in distant hops. HighGraph leverages this concept to model the higher-order social influences between individuals. We first make M adjacency matrices for each layer, each of which represents the m^{th}-powers of our collision-aware adjacency matrix $\hat{\mathbf{A}}$ from section 4.1, where $m \in \{1, ..., M\}$. Multiplying each of them with the agent social feature $\mathbf{X}$ computes the m^{th}-order social influences between the agents:

$$\mathbf{H}_m^{(l)} = \hat{\mathbf{A}}^m \mathbf{X}^{(l)} \mathbf{W}_m^{(l)}, \tag{4}$$

where l is a layer number and $\mathbf{W}_m^{(l)}$ is a trainable weight matrix that learns the m^{th}-order social relations in layer l.

Combintation Layer. Now we devise a combination layer that distills influential social interactions from various distances and integrates them as one representative vector. First, our combination layer concatenates the M higher-order features. Then it goes through the mixture function $\mathcal{F}$, where we utilize multi-layer perceptrons (MLPs) with a non-linear activation function. The described combination layer can be formulated as follows:

$$\mathbf{X}^{(l+1)} = \mathcal{F}(\lambda_1 \mathbf{H}_1^{(l)} \parallel \lambda_2 \mathbf{H}_2^{(l)} \parallel ... \parallel \lambda_M \mathbf{H}_M^{(l)}), \tag{5}$$

where λ_m denotes the trainable weight for the m^{th}-order feature, and $\parallel$ denotes column concatenation. By letting $\sum_{m \in \{1,...,M\}} \lambda_m = 1$, the model discriminates the social influences from different degrees. We repeatedly stack the explained HighGraph layers in order and obtain the higher-order relation embeddings.

5. Experiments

5.1. Datasets

ETH/UCY. The dataset [16, 26] is a classical benchmark for the pedestrian trajectory prediction task. It consists of five different top-down viewed scenes, with coordinate values given in meters. We follow the conventional 'leave-one-out' evaluation strategy [9, 21, 30] and predict the trajectories of 12 frames (4.8 seconds) with 8 frames (3.2 seconds) of observed history.

Stanford Drone Dataset. We also evaluate our method with another widely-used benchmark Stanford Drone Dataset (SDD) [28]. Unlike ETH/UCY, the coordinates of SDD are represented in pixel values. We follow the standard setup from the previous works [30, 37] and report the prediction result of 12 (4.8 seconds) frames with 8 frames (3.2 seconds) of trajectories as inputs.

5.2. Evaluation Metrics

We use two Euclidean-distance-based metrics minADE$_K$ and minFDE$_K$ for evaluation. minADE$_K$ measures the minimum of L2 distances between the ground truth trajectories and the predicted trajectories across K predictions. minFDE$_K$ calculates the minimum of L2 distances between the ground-truth endpoint and the predicted endpoint across K predictions. Following the previous works [9, 22, 38], K is set as 20 in our experiments.

5.3. Implementation Details

We implement HighGraph with PyTorch [25]. Since HighGraph is a plug-and-play module, the optimal way of plugging the module differs between models, and the implementation for each baseline is explained in Section 5.4. The number of graph layers and the higher-order observation degree M is searched within {1,2,3,4,5}, since a larger number rather deteriorates the performance.

5.4. Baselines

HighGraph generates highly beneficial social representations. Accordingly, it can be merged with any existing methods and enhance its prediction. Therefore, we validate the efficacy of HighGraph by plugging it into previous state-of-the-art models: Social-GAN[9], SoPhie[30], Social-STGCNN[22], BiTraP[38], SocialVAE[37] and EigenTrajectory[3]. For Social-GAN and SoPhie, we feed the output of their recurrent encoder as our HighGraph's input. Among the variants of SoPhie, we reproduced SoPhie+$\mathbf{T_A}$ + $\mathbf{I_A}$ by generating the image features by using map features. Since our higher-order

	ETH	HOTEL	UNIV	ZARA1	ZARA2	AVG	SDD
Social-GAN [9]	0.92/1.77	0.71/1.43	0.58/1.22	0.37/0.74	0.40/0.91	0.60/1.21	27.23/41.44
+ HighGraph	0.88/1.52	0.49/0.99	0.56/1.18	0.35/0.71	0.39/0.79	0.53/1.04	18.60/37.00
	4%/14%	**31%/31%**	**3%/3%**	**5%/4%**	**3%/13%**	**9%/13%**	**31%/10%**
SoPhie [30]	0.75/1.51	0.80/1.69	0.59/1.31	0.34/0.64	0.39/0.80	0.57/1.19	16.27/29.38
+ HighGraph	0.62/1.14	0.62/1.28	0.51/1.25	0.32/0.61	0.29/0.75	0.47/1.01	14.60/27.41
	17%/25%	**23%/24%**	**14%/5%**	**6%/5%**	**26%/6%**	**18%/15%**	**10%/7%**
Social-STGCNN [22]	0.64/1.11	0.49/0.85	0.44/0.79	0.34/0.53	0.30/0.48	0.44/0.75	20.60/33.10
+ HighGraph	0.60/0.93	0.31/0.40	0.40/0.70	0.33/0.49	0.29/0.45	0.39/0.59	15.38/26.41
	6%/16%	**36%/52%**	**9%/11%**	**3%/7%**	**3%/6%**	**11%/21%**	**25%/20%**
BiTraP [38]	0.56/0/98	0.17/0.28	0.25/0.47	0.23/0.45	0.16/0.33	0.27/0.50	9.09/16.31
+ HighGraph	0.47/0.73	0.17/0.27	0.25/0.47	0.18/0.33	0.11/0.23	0.24/0.41	8.83/14.90
	16%/25%	**0%/3%**	**0%/0%**	**21%/26%**	**31%/30%**	**11%/18%**	**3%/8%**
SocialVAE [37]	0.41/0.58	0.13/0.19	0.21/0.36	0.17/0.29	0.13/0.22	0.21/0.33	8.10/11.72
+ HighGraph	0.40/0.55	0.13/0.17	0.20/0.33	0.17/0.27	0.11/0.21	0.20/0.30	7.98/11.42
	2%/5%	**0%/11%**	**5%/11%**	**0%/7%**	**15%/5%**	**4%/7%**	**1%/3%**
EigenTrajectory [3]	0.36/0.57	0.13/0.21	0.24/0.49	0.20/0.35	0.15/0.26	0.22/0.38	8.05/13.25
+ HighGraph	0.33/0.56	0.13/0.21	0.23/0.47	0.19/0.33	0.15/0.25	0.21/0.36	7.81/11.09
	8%/2%	**0%/0%**	**4%/4%**	**5%/8%**	**0%/4%**	**5%/4%**	**3%/16%**

Table 1. minADE$_{20}$/minFDE$_{20}$ result of HighGraph. Our method noticeably improves the previous state-of-the-art methods in both datasets, demonstrating the importance of higher-order relational reasoning. The third rows in each baseline are the relative improvements in percentage. The performance improvement achieved by our proposed method is indicated in **bold**.

graph convolution includes pair-wise reasoning, we replace the existing graph convolution module in Social-STGCNN with the HighGraph network. For conditional variational autoencoder (CVAE) [31] based models, we concatenate the HighGraph representations to their original encoded features. We report the quantitative result of BiTraP-NP from SocialVAE for the recently reported issue [37]. Additionally, we report the result of SocialVAE with final position clustering (FPC) for a fair comparison. Lastly, we added a recent state-of-the-art EigenTrajectory+SGCN as a new baseline to emphasize the capacity of our method. The rest of the configurations are adopted from the official repository of each baseline. More details of implementation are provided in the supplementary material.

5.5. Quantitative Analysis

In this section, we analyze the impact of reasoning higher-order relations for trajectory prediction by plugging our proposed HighGraph into previous state-of-the-art baselines. The results are reported in Table 1.

HighGraph consistently improves the minADE$_{20}$ and minFDE$_{20}$ results for all baselines. Especially, applying HighGraph in Social-GAN and Social-STGCNN achieves more than 52% performance gain of HOTEL subset in minFDE$_{20}$. The relative increase in the CVAE-based methods is minor compared to that of non-CVAE-based baselines. We assume this is because the models are more drawn to learning the CVAE latent space for better sampling. However, as will be shown in the following qualitative results, HighGraph stands out in producing much more socially reasonable trajectories, even with the small quantitative increase in the metrics.

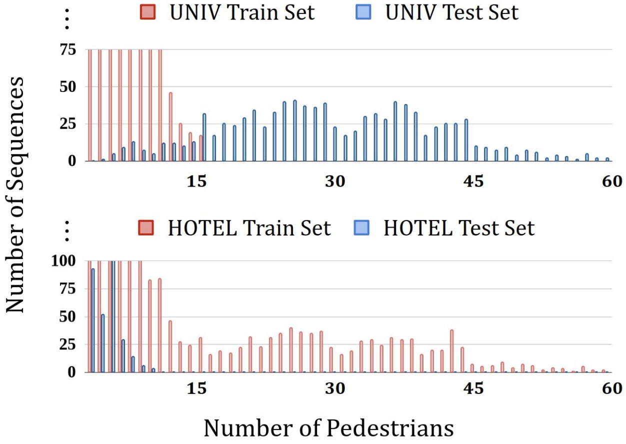


Figure 4. The distribution of the number of pedestrians for the UNIV subset data. UNIV data is trained with sequences with a small number of pedestrians, while the test set contains sequences with a larger number of pedestrians. Best viewed in color.

For a more thorough analysis, we provide the distributions of the pedestrians in the UNIV and HOTEL subset of the ETH/UCY dataset in Figure 4. The distribution shows that the UNIV subset is trained with data with a relatively small amount of pedestrians in the sequences(average 4.4 peds/seq). However, the amount of pedestrians in the test set is comparably larger than the other subsets(average 25.6

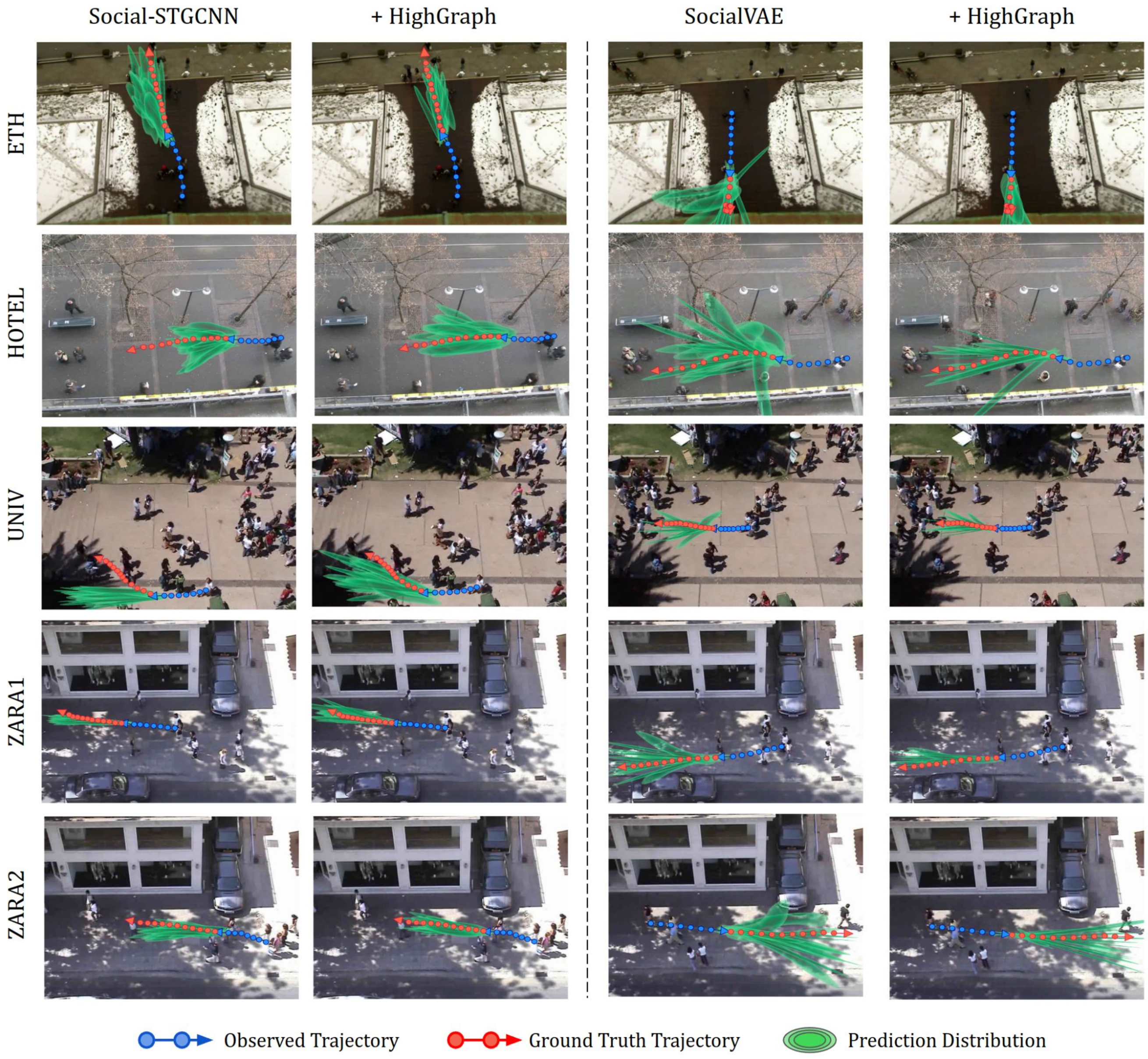


Figure 5. Visualization of the qualitative influence of HighGraph. We illustrate the result of each subset of the ETH/UCY dataset, comparing the original model and the HighGraph-plugged model. Best viewed in color.

peds/seq), indicating the shortage of training data for the model to learn complex social interactions. Accordingly, we observe that the benefits of HighGraph are lowest in the UNIV data for all baselines. However, the other datasets, such as HOTEL, are trained with sufficient social interactions, thereby showing substantial improvements.

5.6. Qualitative Analysis

The previous section demonstrated the quantitative efficiency of HighGraph. In this section, we present the qualitative results of HighGraph that the numerical metrics were unable to highlight. Specifically, we first analyze the general prediction results of a single agent. Then we move on to the multi-agent predictions with more complex social interactions to validate the higher-order relational reasoning power of our HighGraph.

5.6.1 Single-agent General Prediction.

First, in Figure 5, we visualize the enhancement of the predictions of the general cases when HighGraph is combined with existing models. We use Social-STGCNN and SocialVAE as baselines for our qualitative analysis since Social-STGCNN is a representative graph-based method and SocialVAE is a recent CVAE-based state-of-the-art method. In

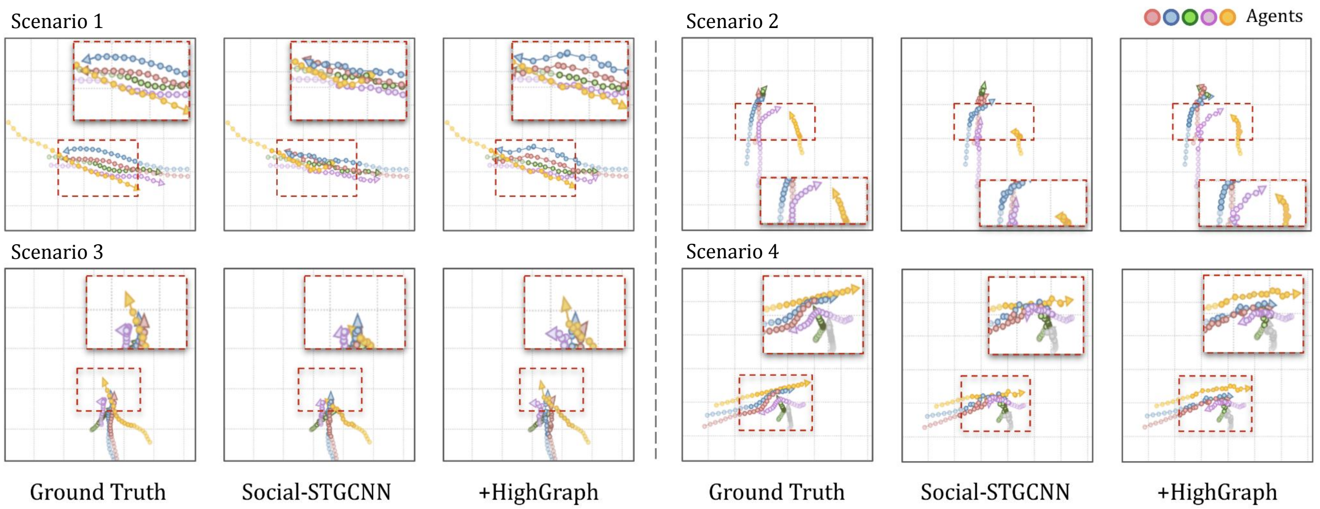


Figure 6. Visualization of how HighGraph can improve the predictions in higher-order social scenarios. The scenes are selected from ETH/UCY subsets. Best viewed in color.

every illustrated case, the predicted distributions of HighGraph are closer to the ground truth. While the benefits of the probabilistic models are in multiple trajectory predictions, some of the examples in Social-STGCNN and Social-VAE generate socially unacceptable ones (e.g. colliding, walking into the driveway). However, HighGraph noticeably adjusts the trajectories to be socially acceptable while being closer to the ground truth.

5.6.2 Multi-agent Higher-order Prediction.

Next, we illustrate the social scenarios with explicit higher-order influences and emphasize the strength of HighGraph's relational reasoning. Especially, we refine the multi-agent trajectory prediction of Social-STGCNN. We manually choose the social scenarios where higher-order influences between agents seem significant. Then, we illustrate the ground truth trajectories, baseline prediction, and our refined prediction in order for each scenario in Figure 6. For the clarity of the illustration, we remove the background image and other less-influential agents from the scene.

Scenario 1. The first scenario depicts a case where two groups of parallel-walking agents (first-order) are walking toward each other. This triggers a potential collision, and naturally, the groups avoid collision by walking to their right (second-order). This interaction between them further influences the yellow agent behind them to navigate to the right (third-order). **Scenario 2.** This scenario demonstrates a case where three agents (green, red, blue) are gathering at one place (first-order). Therefore, a following agent (purple) stirs its path to the right (second-order). These successive flow of interactions eventually affects the yellow agent to slow down to avoid collision with the purple agent. **Scenario 3.** The yellow agent in this scene runs into the agents coming from its left (red, blue). Therefore, it turns right and passes the agents with a faster velocity (first-order). Accordingly, the red and blue agents slow down (second-order) and this further affects the purple and green agents to slow down as well (third-order). **Scenario 4.** The last scenario also well-describes the chaining influences of higher-order relations. The trajectories of the grey agent stir the path of the green agent (first-order). This causes the purple agent to detour (second-order), and as a consequence, the red and blue agents turn left (third, fourth-order).

Comparing the predicted trajectories in such scenarios, the results commonly demonstrate the necessity of HighGraph for two main reasons. First, HighGraph clearly avoids collisions. We speculate this is the benefit of our collision-aware kernel function, which explicitly models potential collisions. Second, with higher-order graph convolutions, HighGraph generates more 'socially influenced' trajectories that can not be captured only with pair-wise relational reasoning. More examples of the comparison will be presented in the supplementary material.

5.7. Ablation Study

5.7.1 Effect of the Collision-aware Kernel Function.

In this section, we demonstrate the effectiveness of our collision-aware kernel function. Table 2 illustrates the AVG results of Social-STGCNN differing with the choice of the kernel functions. The velocity-based function proposed in [22] does not take the distance between agents into consideration and naively assigns equal weights to agents with the same observed velocity. However, our collision-aware kernel function models the complex pair-wise relations based on the social prior that pedestrians walk in a way that avoids collision. Therefore, it naturally generates fewer prediction

	No Weights	Velocity -based	Ours	
			w/o d	w/ d
AVG	0.49/0.79	0.44/0.75	0.43/0.71	**0.39/0.61**

Table 2. We compare the average $\text{minADE}_{20}/\text{minFDE}_{20}$ result of 5 ETH/UCY subsets with different kernel functions applied to Social-STGCNN.

	LSTM	+GCN	+MixHop	**+HighGraph**
ETH	2.13/3.30	2.11/3.31	2.03/3.13	1.95/3.07
HOTEL	1.01/1.38	1.02/1.35	1.02/1.34	0.97/1.29
UNIV	0.94/1.08	0.95/1.06	1.02/1.21	0.88/1.01
ZARA1	1.43/2.36	1.41/2.30	1.37/2.19	1.21/2.02
ZARA2	1.10/1.50	1.08/1.46	1.07/1.42	0.95/1.38
AVG	1.32/1.92	1.31/1.90	1.30/1.86	1.19/1.75

Table 3. $\text{minADE}_{20}/\text{minFDE}_{20}$ result of a simple LSTM model with different GNN variants for social relation modeling. ETH/UCY dataset is used for this experiment.

errors than the previous method. Furthermore, we scale the interactions with the distance between the agents d, and refine the weight values, which produces better results.

5.7.2 Comparison with GNN Variants.

Next, we distinguish our HighGraph against other graph neural network (GNN) variants. This is done by attaching GNNs to a simple long short-term memory (LSTM) [11] encoder-decoder based trajectory network. We compare the social relation modeling ability of HighGraph with GCN [14] and MixHop [1]. For a fair comparison, the number of graph layers is equally set as 3, and the higher-order observation degree is set as 2 for MixHop and HighGraph. The results are summarized in Table 3. We observe that enabling a simple RNN model to incorporate relational reasoning with graphs improves performance. MixHop and our HighGraph generate better results than GCN for their ability to reason about higher-order features. However, with complex interaction modeling with a collision-aware kernel function, and a combination layer that combines and distills influential information from higher-order features, HighGraph performs best among the compared variants.

5.7.3 Component Analysis.

HighGraph consists of two main components: the collision-aware kernel function (C.A) and the higher-order graph convolution (H.G.C). This section analyzes how each of these components contributes to the overall performance of HighGraph. We begin from Social-STGCNN and apply C.A and H.G.C each in order, where $M = 2$ for H.G.C. Then we combine all the components to form the complete version of HighGraph. The results are reported in Figure 7. We observe that both of the components of HighGraph show noticeable improvements in reducing displacement errors. Specifically, the average of ETH/UCY benefits more from the higher-order graph convolution, and SDD benefits more from the collision-aware kernel function. However, assembling all the components together leads to a further decrease in both minADE_{20} and minFDE_{20}, emphasizing the importance of higher-order relational reasoning based on complex social interaction modeling.

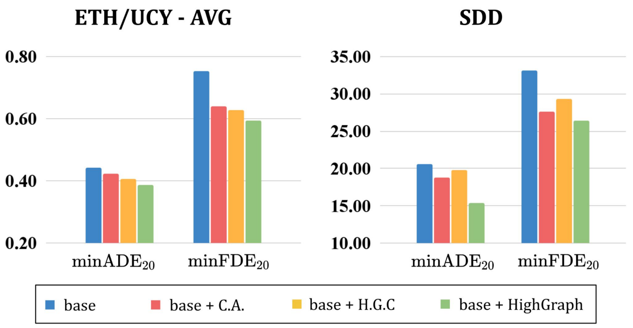


Figure 7. Results of how each component of HighGraph improves the performance when applied to the baseline model Social-STGCNN. Combining them all together as HighGraph yields the best result. Best viewed in color.

6. Conclusion

Behind each and every human decision lies an immeasurable amount of chaining social relations that originate the behavior. This work focuses on learning these indirect higher-order influences from socially distant neighbors, which haven't yet been explored in the literature. Therefore, we develop HighGraph, a graph-based higher-order relational reasoning method for trajectory prediction. Our collision-aware kernel function first constructs a social graph adjacency based on the prior that pedestrians walk in a way that avoids collision. Powering the designed adjacency matrix expands the social observation, enabling HighGraph to aggregate and combine influences from various distances. We develop HighGraph as a plug-and-play module and demonstrate how HighGraph can be applied to existing state-of-the-art methods and improve their performance both quantitatively and qualitatively.

Acknowledgements. This research was supported by Autonomous Driving Center, Hyundai Motor Company R&D Division.(50%), Electronics and Telecommunications Research Institute(ETRI) grant funded by the Korean government. (24ZC1200, Research on hyper-realistic interaction technology for five senses and emotional experience,11%), the Korea Research Institute for Defense Technology Planning and Advancement (KRIT) Grant funded by the Korean Government(Grant KRIT-CT-23-021,10%), the Basic Science Research Program (NRF-2021R1A6A1A13044830,10%) and by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (2022-0-00043,10%, IITP-2024-2020-0-01819, 5%, 2019-0-00079 (Artificial Intelligence Graduate School Program(Korea University)), 1%), and the National Research Foundation of Korea grant(NRF-2022R1F1A1074334,3%).

References

[1] Sami Abu-El-Haija, Bryan Perozzi, Amol Kapoor, Nazanin Alipourfard, Kristina Lerman, Hrayr Harutyunyan, Greg Ver Steeg, and Aram Galstyan. Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing. In *international conference on machine learning*, pages 21–29. PMLR, 2019. 2, 8

[2] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 961–971, 2016. 1, 2

[3] Inhwan Bae, Jean Oh, and Hae-Gon Jeon. Eigentrajectory: Low-rank descriptors for multi-modal trajectory forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10017–10029, 2023. 4, 5

[4] Niccoló Bisagno, Bo Zhang, and Nicola Conci. Group lstm: Group trajectory prediction in crowded scenarios. In *Proceedings of the European conference on computer vision (ECCV) workshops*, pages 0–0, 2018. 2

[5] Hyung-gun Chi, Myoung Hoon Ha, Seunggeun Chi, Sang Wan Lee, Qixing Huang, and Karthik Ramani. Infogcn: Representation learning for human skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20186–20196, 2022. 2

[6] Pranav Singh Chib and Pravendra Singh. Recent advancements in end-to-end autonomous driving using deep learning: A survey. *IEEE Transactions on Intelligent Vehicles*, 2023. 1

[7] Lise Getoor and Ben Taskar. *Introduction to statistical relational learning*. MIT press, 2007. 2

[8] Mahsa Golchoubian, Moojan Ghafurian, Kerstin Dautenhahn, and Nasser Lashgarian Azad. Pedestrian trajectory prediction in pedestrian-vehicle mixed environments: A systematic review. *IEEE Transactions on Intelligent Transportation Systems*, 2023. 1

[9] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2255–2264, 2018. 1, 2, 4, 5

[10] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 639–648, 2020. 2

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

[12] Yingfan Huang, Huikun Bi, Zhaoxin Li, Tianlu Mao, and Zhaoqi Wang. Stgat: Modeling spatial-temporal interactions for human trajectory prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6272–6281, 2019. 2

[13] Sungjune Kim, Seongjun Yun, Jongwuk Lee, Gyusam Chang, Wonseok Roh, Dae-Neung Sohn, Jung-Tae Lee, Hogun Park, and Sangpil Kim. Self-supervised multimodal graph convolutional network for collaborative filtering. *Information Sciences*, 653:119760, 2024. 2

[14] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 2, 8

[15] 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. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[16] Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. Crowds by example. In *Computer graphics forum*, pages 655–664. Wiley Online Library, 2007. 4

[17] Jiachen Li, Chuanbo Hua, Jinkyoo Park, Hengbo Ma, Victoria Dax, and Mykel J Kochenderfer. Evolvehypergraph: Group-aware dynamic relational reasoning for trajectory prediction. *arXiv preprint arXiv:2208.05470*, 2022. 1, 2

[18] Linjie Li, Zhe Gan, Yu Cheng, and Jingjing Liu. Relation-aware graph attention network for visual question answering. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10313–10322, 2019. 2

[19] Wanhua Li, Yueqi Duan, Jiwen Lu, Jianjiang Feng, and Jie Zhou. Graph-based social relation reasoning. In *European conference on computer vision*, pages 18–34. Springer, 2020. 2

[20] Yanan Liu, Hao Zhang, Yanqiu Li, Kangjian He, and Dan Xu. Skeleton-based human action recognition via large-kernel attention graph convolutional network. *IEEE Transactions on Visualization and Computer Graphics*, 29(5):2575–2585, 2023. 2

[21] Karttikeya Mangalam, Yang An, Harshayu Girase, and Jitendra Malik. From goals, waypoints & paths to long term human trajectory forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15233–15242, 2021. 4

[22] Abduallah Mohamed, Kun Qian, Mohamed Elhoseiny, and Christian Claudel. Social-stgcnn: A social spatio-temporal graph convolutional neural network for human trajectory prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14424–14432, 2020. 1, 2, 4, 5, 7

[23] Medhini Narasimhan, Svetlana Lazebnik, and Alexander Schwing. Out of the box: Reasoning with graph convolution nets for factual visual question answering. *Advances in neural information processing systems*, 31, 2018. 2

[24] Yunmi Park and Max Garcia. Pedestrian safety perception and urban street settings. *International journal of sustainable transportation*, 14(11):860–871, 2020. 1

[25] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 4

[26] Stefano Pellegrini, Andreas Ess, and Luc Van Gool. Improving data association by joint modeling of pedestrian trajectories and groupings. In *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part I 11*, pages 452–465. Springer, 2010. 4

[27] Amir Rasouli and John K Tsotsos. Autonomous vehicles that interact with pedestrians: A survey of theory and practice. *IEEE transactions on intelligent transportation systems*, 21 (3):900–918, 2019. 1

[28] Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Learning social etiquette: Human trajectory understanding in crowded scenes. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14*, pages 549–565. Springer, 2016. 4

[29] Andrey Rudenko, Luigi Palmieri, Michael Herman, Kris M Kitani, Dariu M Gavrila, and Kai O Arras. Human motion trajectory prediction: A survey. *The International Journal of Robotics Research*, 39(8):895–935, 2020. 2

[30] Amir Sadeghian, Vineet Kosaraju, Ali Sadeghian, Noriaki Hirose, Hamid Rezatofighi, and Silvio Savarese. Sophie: An attentive gan for predicting paths compliant to social and physical constraints. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1349–1358, 2019. 4, 5

[31] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28, 2015. 2, 5

[32] Anirudh Vemula, Katharina Muelling, and Jean Oh. Social attention: Modeling attention in human crowds. In *2018 IEEE international Conference on Robotics and Automation (ICRA)*, pages 4601–4607. IEEE, 2018. 2

[33] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*, pages 165–174, 2019. 2

[34] Tianjun Wei, Tommy WS Chow, Jianghong Ma, and Mingbo Zhao. Expgcn: Review-aware graph convolution network for explainable recommendation. *Neural Networks*, 157:202–215, 2023. 2

[35] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 753–763, 2020. 2

[36] Chenxin Xu, Maosen Li, Zhenyang Ni, Ya Zhang, and Siheng Chen. Groupnet: Multiscale hypergraph neural networks for trajectory prediction with relational reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6498–6507, 2022. 1, 2

[37] Pei Xu, Jean-Bernard Hayet, and Ioannis Karamouzas. Socialvae: Human trajectory prediction using timewise latents. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IV*, pages 511–528. Springer, 2022. 4, 5

[38] Yu Yao, Ella Atkins, Matthew Johnson-Roberson, Ram Vasudevan, and Xiaoxiao Du. Bitrap: Bi-directional pedestrian trajectory prediction with multi-modal goal estimation. *IEEE Robotics and Automation Letters*, 6(2):1463–1470, 2021. 4, 5

[39] Xikun Zhang, Chang Xu, and Dacheng Tao. Context aware graph convolution for skeleton-based action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14333–14342, 2020. 2

[40] Zhiming Zou, Kenkun Liu, Le Wang 0003, and Wei Tang. High-order graph convolutional networks for 3d human pose estimation. In *BMVC*, 2020. 2