Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 4192367, 10 pages
https://doi.org/10.1155/2022/4192367

*Research Article*

# MDST-DGCN: A Multilevel Dynamic Spatiotemporal Directed Graph Convolutional Network for Pedestrian Trajectory Prediction

**Shaohua Liu**,[1] **Haibo Liu**,[1] **Yisu Wang**,[1] **Jingkai Sun**,[1] **and Tianlu Mao**[2]

[1]*School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[2]*Beijing Key Laboratory of Mobile Computing and Pervasive Device, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China*

Correspondence should be addressed to Tianlu Mao; ltm@ict.ac.cn

Received 28 December 2021; Revised 1 March 2022; Accepted 10 March 2022; Published 12 April 2022

Academic Editor: Andrea Loddo



Pedestrian trajectory prediction is an essential but challenging task. Social interactions between pedestrians have an immense impact on trajectories. A better way to model social interactions generally achieves a more accurate trajectory prediction. To comprehensively model the interactions between pedestrians, we propose a multilevel dynamic spatiotemporal digraph convolutional network (MDST-DGCN). It consists of three parts: a motion encoder to capture the pedestrians' specific motion features, a multilevel dynamic spatiotemporal directed graph encoder (MDST-DGEN) to capture the social interaction features of multiple levels and adaptively fuse them, and a motion decoder to produce the future trajectories. Experimental results on public datasets demonstrate that our model achieves state-of-the-art results in both long-term and short-term predictions for both high-density and low-density crowds.

## 1. Introduction

The task of pedestrian trajectory prediction is to predict pedestrians' future trajectories given their historical trajectories in the scenario. Pedestrian trajectory prediction plays a notable role in many aspects, such as automatic driving [1] and robot navigation [2–5]. To predict an accurate trajectory, only considering the historical trajectory of the target pedestrian is not enough. Other pedestrians' influences on the target pedestrian, which are called "social interaction features," can often help make a better prediction. With the longer prediction horizon and denser crowds, the temporal correlations in the trajectories between current and previous time steps grow weaker and the impact of interactions on pedestrians' motion grows stronger.

To model social interactions, traditional methods use rule-based functions [6–10]. While rule-based methods can only capture simple interactions, data-driven methods use neural networks to automatically extract the social interaction features from the data, which can make use of the interaction features more effectively. Many data-driven methods obtained social interaction features based on pooling [11–14] or attention mechanisms [1, 15–20]. The graph convolutional neural networks have developed rapidly in recent years, and the graph structure is naturally suitable for directly describing the interactions between pedestrians. As a result, graph convolutional neural networks [21–25] have achieved excellent results in pedestrian trajectory prediction.

Although there are many graph convolutional neural network-based methods, they do not make full use of them. For example, Social-BiGAT [21] only uses the graph representation as the pooling mechanism on the states of the recurrent neural networks. The new methods STGAT [22] and Social-STGCNN [23] constructed spatiotemporal graphs to model social interactions and achieved excellent results in predictions.

However, they ignore a crucial point that even if the social interactions with nearby pedestrians or distant pedestrians are of the same type, they will result in different actions of the target pedestrian. As shown in Figure 1, at time steps $t_1$ and $t_2$, when the target pedestrian marked with the red circle avoids the nearby pedestrians and the distant pedestrians, respectively, his avoidance movements will be different. The former is a sudden avoidance producing a trajectory with high curvature, while the latter is an early avoidance producing a trajectory with low curvature. Moreover, with the increase in prediction horizon, pedestrians far from the target pedestrian may become more important. From time step $t_1$ to $t_2$, pedestrian B has little impact on the target pedestrian, but in the total period from $t_1$ to $t_3$, the merging of them is the main factor affecting the target pedestrian's trajectory. In other words, the influence of nearby pedestrians is mainly sudden and short-term, while faraway pedestrians have long-term effects on the target pedestrian's movement tendency.

Most previous methods [21–25] use a single graph to model these two types of influences and tend to capture "average social interaction features." However, these two types of influences are more suitable to be modeled separately at different levels of a multilevel graph. Besides, many methods [23, 25] build an undirected graph to model social interactions. However, social interactions between pedestrians are nonsymmetrical. Therefore, building a digraph is more suitable for social interactions. Other methods [24] build a directed graph by predefined rules, such as inserting edges from all people inside the view area. But predefined rules are incomplete. For example, a pedestrian may slow down to wait for his companion without looking at him. Thus, a data-driven way to build a directed edge is much better.

To address the limitations of these works, we propose a multilevel dynamic spatiotemporal directed graph representation to model the interactions between pedestrians comprehensively. In our graph, different levels model interactions of pedestrians at different distance ranges. As shown in Figure 1, whether there is a spatial edge from a pedestrian to the target pedestrian at a level depends on whether their distance is within the corresponding distance range. With the change of time, the spatial edge between two pedestrians may break at one level and link at another. Even if the edge keeps linking at the same level, the influence of the neighbour also changes dynamically over time. To process the multilevel graph, we propose a multilevel dynamic spatiotemporal digraph convolutional network (MDST-DGCN). At each level of the graph, we use a node aggregator architecture to generate social interaction embedding by sampling and aggregating features from a node's spatial neighborhood like GraphSAGE [26]. Because social interactions are location independent, we do an aligning operation before aggregating features, which can advance performance significantly. Through the orderly use of sampling, aligning, and aggregating, the aggregator architecture becomes a naturally data-driven way to describe a directed edge. For each level of the graph, after the spatial interactions are captured, an LSTM [27] is used to capture the temporal correlations of interactions. And then, MDST-DGCN fuses interaction features of all levels adaptively. Through modelling social interactions at different levels, our multilevel dynamic spatiotemporal digraph convolutional network (MDST-DGCN) can fully extract pedestrians' social interaction features.

In summary, our contribution is twofold. First, we propose using the spatiotemporal dynamic map with a multilevel concept to separate pedestrian nodes, resulting in varying effects on the trajectory depending on the distance between pedestrians, which may aid in the extraction of social interaction features by partitioning pedestrian distances at various levels. Second, we create an aggregator based on the GraphSAGE that converts the original static adjacency graph structure into a dynamic directed graph structure by sampling, aligning, and aggregating, reducing the effect of individual coordinates on the model and the overfitting phenomenon. We verified the performance of the model on the general pedestrian trajectory datasets. The experimental results show that our model has achieved state-of-the-art results in both long-term and short-term predictions for both high-density and low-density crowds.

## 2. Related Work

*2.1. Pedestrian Trajectory Prediction.* Pedestrian trajectory prediction has become a focal task in recent years, and corresponding solutions have been springing up. Comprehensively modelling the interactions between pedestrians is a crucial point to obtain better prediction results.

Traditionally, researchers created hand-crafted functions [3, 6–10] to predict trajectories, but hand-crafted functions are limited, so they are unable to model all types of social interactions. Recently, deep learning-based methods have become popular because they can learn to model various interactions from data.

Some researchers designed their methods based on pooling mechanisms [11–14] to capture dependencies between pedestrians. The S-LSTM [11] introduces a "social" pooling layer which allows the LSTMs of spatially proximal sequences to share their hidden states with each other. Group-LSTM [12] adjusts the pooling layer by dropping the information of pedestrians who are moving coherently with the target pedestrian. MX-LSTM [13] has a pooling layer, which exploits the Vislet information. The above three pooling methods only consider the pedestrians in the local area and fuse their features averagely, while SGAN introduces a pooling module considering all pedestrians in a computationally efficient way and adaptively select their features with a max-pooling operation.

While most pooling-based methods treat pedestrians equally, attention-based methods [1, 15–20] assign different weights to interactive pedestrians. Most of these methods [1, 11–14, 16–20] assign an LSTM for each pedestrian, and the pooling mechanisms or the attention mechanisms usually work on the hidden states of pedestrians' LSTMs to adaptively fuse other pedestrians' motion features with the target pedestrian. More recently, STAR [28] captures

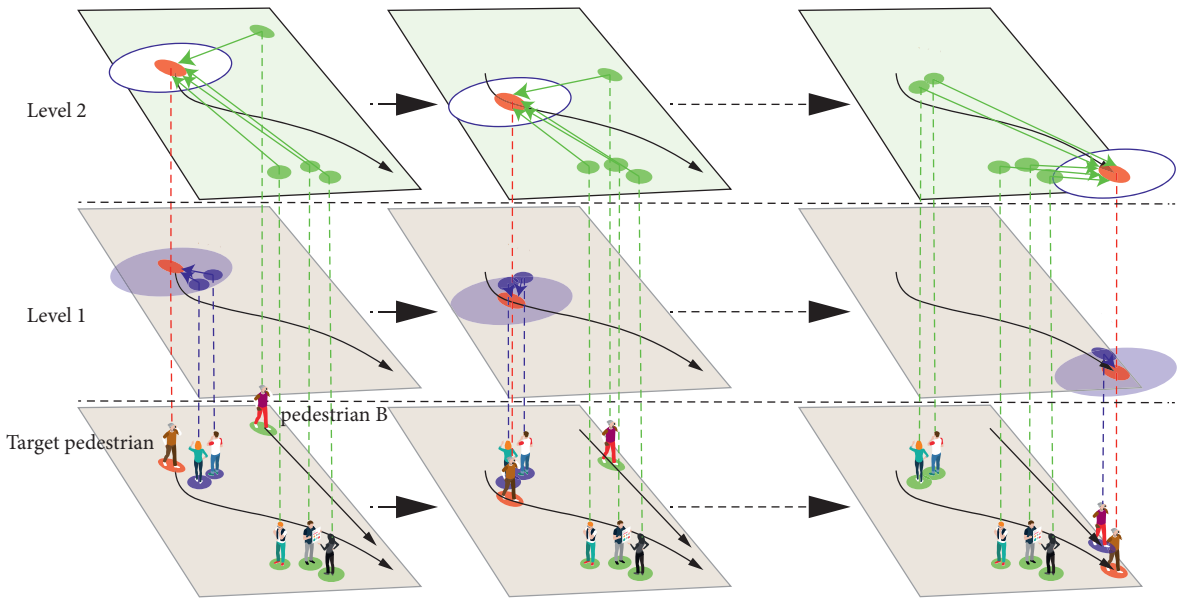


FIGURE 1: The influences of pedestrians in the nearby area and the faraway area on the target pedestrian are more suitable to be modeled separately at different levels.

complex spatiotemporal interactions by interleaving between spatial and temporal transformers [29].

As the graph structure is naturally suitable for directly describing the interactions between pedestrians, graph convolutional neural networks are introduced to this task. Social-BiGAT [21] replaced the pooling mechanisms with the graph attention network, which also works on the hidden states of LSTMs. In other words, Social-BiGAT did not model the whole duration of the crowds' interactions as a spatiotemporal graph but only used the graph attention network to capture the spatial social interactions. Social-STGCNN [23] and STGAT [22] both constructed spatiotemporal graphs to model social interactions. However, the graph of Social-STGCNN is a complete undirected graph. It does not conform to the asymmetry of pedestrian interactions. Zhang et al. [24] built a directed graph by inserting edges from all people inside the view area. However, all of these graphs model all the social interactions at only one level. Instead, we build a multilevel dynamic spatiotemporal directed graph to overcome their limitations.

*2.2. Graph Convolutional Neural Network.* Graph convolutional neural network is an emerging topic in deep learning research, and it provides a practical approach to process graph data with nongrid structures. We can divide graph convolutional neural networks into spectral approaches [30–32] and spatial approaches [26, 33, 34]. Spectral approaches work with a spectral representation of the graphs, while spatial approaches define convolutions directly on the graph, operating on groups of spatially close neighbours. Spectral approaches' learned filters depend on the Laplacian eigenbasis, which depends on the graph structure. Thus, a model trained on a specific structure cannot be directly applied to a graph with a different structure. However, the graph used to model pedestrians' social interactions changes with time. Thus, spectral approaches are not suitable for pedestrian trajectory prediction. And, our approach belongs to the spatial approaches.

In fact, our approach follows the methodology of GraphSAGE [26]. However, our graph is a multilevel dynamic spatiotemporal directed graph, while GraphSAGE can only process a fixed spatial graph without multiple levels. ST-GCN [34] built a dynamic spatiotemporal graph to automatically learn both the spatial and temporal patterns of human actions to recognize skeleton-based actions. Social-STGCNN [23], which is a variant of ST-GCN that builds a single-level undirected graph to model all the social interactions, has achieved excellent results in pedestrian trajectory prediction.

## 3. Methods

*3.1. Problem Definition.* Given the historical trajectories of all pedestrians in the scenario, the task of trajectory prediction is to predict their future trajectories simultaneously. The notations $p1$, $p_2$, …, $p_N$ represent $N$ pedestrians in the scenario. The position of a specific pedestrian $p_i\,(i \in [1, N])$ at any historical time step $t\,(t \in [1, T_{\text{obs}}])$ is defined as $X_i^t = (x_i^t, y_i^t)$. Our goal is to predict the positions of pedestrians at any future time step $t\{t\varepsilon[T_{\text{obs}} + 1 + T_{\text{obs}} + T_{\text{pred}}]\}$, and for a specific pedestrian $p_i$, the predicted position is denoted as $\widehat{Y}_i^t = (\widehat{x}_i^t, \widehat{y}_i^t)$, while the ground truth is defined as $Y_i^t = (x_i^t, y_i^t)$. The first-order difference trajectory of a pedestrian $p_i$ is defined as $\{\Delta X_i^t | t \in [1, T_{\text{obs}} + T_{\text{pred}}]\}$, where $\Delta X_i^t = X_i^t - X_i^{t-1}$.

*3.2. Overall Model.* As shown in Figure 2(b), MDST-DGCN consists of three parts: a motion encoder, a multilevel dynamic spatiotemporal directed graph encoder (MDST-DGEN), and a motion decoder. The motion encoder is used

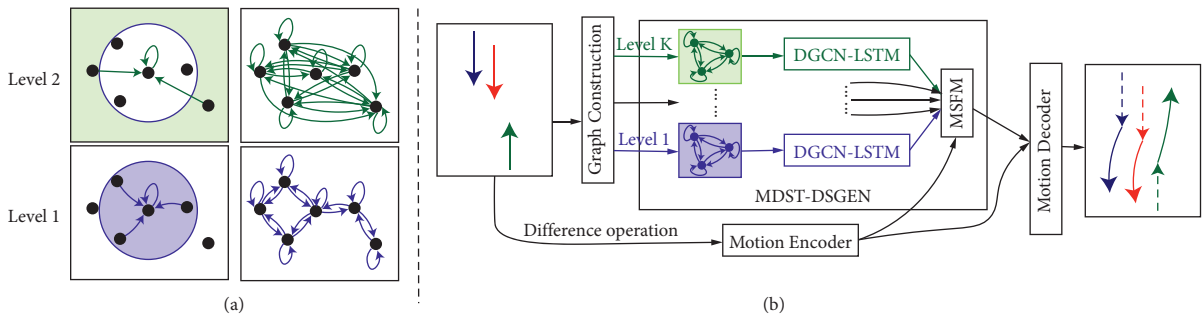


FIGURE 2: (a) Illustration of how to build a two-level spatial graph with the level distance list $\{d_1, +\infty\}$ at a certain time step. Level 1 shows the relation between pedestrians within the distance of $d_1$, and Level 2 shows it beyond the distance of $d_1$. The left graphs show the edges from others to a specific pedestrian at different levels, and the right ones show all edges in the graphs of different levels. (b) The architecture of MDST-DGCN.

to capture the pedestrian-specific motion features, and the MDST-DGEN is used to capture the social interaction features. We construct a multilevel dynamic spatiotemporal digraph processed by the MDST-DGEN to model the social interactions between pedestrians. After the motion features and social interaction features are extracted, they are fed into the motion decoder to predict future trajectories.

*3.3. Graph Construction.* We construct a multilevel dynamic spatiotemporal directed graph to model the multilevel social interactions between pedestrians. The nodes of the graph are the pedestrians in the scenario. Given the hyperparameter level distance list $\{d_1, d_2, \ldots, d_K\}$, we construct a graph with $K$ levels. At each time step, if the distance from node $v_j$ to node $v_i$ is more than $d_{k-1}$ and less than $d_k$, a spatial edge from $v_j$ to $v_i$ will exist in the $k_{th}$ $(k \in [1, K])$ level. Specifically, in the $1_{st}$ level, a spatial edge exists when the distance is less than $d_1$. For each node at all levels, we add a loop spatial edge. Figure 2(a) shows how to build a two-level spatial graph with the level distance list $\{d_1, +\infty\}$ at a certain time step. In addition to spatial edges, there are temporal edges, which connect the same pedestrians in consecutive frames. If there is only one level and $d_1 = +\infty$, the graph will degrade into a complete graph, which is of the same structure as STGAT. At the time step $t$, the attribute of node $v_i^t$ is the position $X_i^t$ of pedestrian $p_i$.

*3.4. Motion Encoder.* The motion encoder is used to extract pedestrian-specific motion features. The input is the first-order difference trajectory $\{\Delta X_i^t | t \in [1, T_{\text{obs}}]\}$. The motion encoder is composed of a linear layer and an LSTM. The linear layer transforms the $\Delta X_i^t$ into a higher dimension vector. Then, it is fed into the LSTM to get a motion feature vector. For each pedestrian $p_i$, the process can be formulated as

$$h_{\text{mo}}^t(i) = \text{LSTM}_{\text{mo}}\left(h_{\text{mo}}^{t-1}(i), \text{Linear}_{\text{en}}\left(\Delta X_i^t; W_{\text{en}}\right); W_{\text{mo}}\right). \quad (1)$$

Here, $W_{\text{en}}$ denotes the trainable weights of the linear layer, $W_{\text{mo}}$ is the trainable weights of the LSTM ($\text{LSTM}_{\text{mo}}$), and the hidden states of $\text{LSTM}_{\text{mo}}$ at the previous time step and the current time step of pedestrian $p_i$ are denoted as $h_{\text{mo}}^{t-1}(i)$ and $h_{\text{mo}}^t(i)$, respectively. At last, the motion encoder obtains each pedestrian's motion feature vector $h_{\text{mo}}^{T_{\text{obs}}}(i)$, which is marked as $h_{\text{mo}}(i)$ in the following sections.

*3.5. MDST-DGEN.* MDST-DGEN is a crucial component of our model. It processes the multilevel dynamic spatiotemporal directed social graph to obtain the social interaction features. If the graph is of $K$ levels, MDST-DGEN will have $K$ DGCN-LSTMs to process each level of it and an MSFM to fuse the features extracted from each level. In our implementation, $K$ DGCN-LSTMs share the weights, so increasing the number of levels does not increase the parameters of the model.

*3.6. DGCN-LSTM.* After building the multilevel graph, each level of the graph is fed into a DGCN-LSTM. A DGCN-LSTM consists of a node aggregator architecture to process the spatial edges and an LSTM to process the temporal edges. We follow the design of GraphSAGE [26], which processes graphs by sampling and aggregating. Our node aggregator architecture generates embedding by sampling, aligning, and aggregating features from a node's spatial neighbourhood at each level.

*3.6.1. Sampling.* Due to the different numbers of pedestrians in the scene, to process all nodes of different graphs in parallel, we expand the number of neighbours to a fixed number $m$ by uniformly sampling a certain number of neighbours. Here, if there is an edge from node $v_j$ to node $v_i$, $v_j$ will be the neighbour of $v_i$. We denote the $m$ neighbours of any node $v$ as the neighbourhood set $\mathcal{N}(v)$.

*3.6.2. Aligning.* For the node $v_i$, its attribute is the pedestrian's position $X_i^t$ and the attributes of its neighbourhood set can be denoted as $\left\{X_j^t | \forall v_j \in \mathcal{N}(v_i)\right\}$. Social interaction is location independent, so we design an aligning operation to make the node aggregator architecture more generalizable. After aligning is done, the aligned attributes of any node $v_i$'s neighbourhood set can be denoted as $\left\{X_j^t - X_i^t | \forall v_j \in \mathcal{N}(v_i)\right\}$. The intuitive understanding of the

alignment operation is that we change the origin of coordinates to the position of node $v_i$.

*3.6.3. Aggregating.* After the aligning, we aggregate the aligned attributes of $v_i$'s neighborhood set to obtain the new feature embedding of $v_i$. It can be formulated as follows:

$$V_i^t = MAX\left(\left\{f\left(X_j^t - X_i^t\right) | \forall v_j \in \mathcal{N}(v_i)\right\}\right), \tag{2}$$

where MAX is the max operator that take the elementwise max of the transformed attribute vectors $\left\{f(X_j^t - X_i^t) | \forall v_j \in \mathcal{N}(v_i)\right\}$ and $f$ is the trainable linear mapping to convert a low-dimension vector to high dimension. We implement the max operator by using a max-pooling layer. Through the orderly use of sampling, aligning, and aggregating, our model can meet the requirement of a directed graph that the relation between two nodes in the directed graph is asymmetric.

After the spatial edges are processed, an LSTM is used to process the temporal edges as follows:

$$h_g^t(i) = \mathrm{LSTM}_g\left(h_g^{t-1}(i), V_i^t; W_g\right), \tag{3}$$

where $W_g$ is the trainable weights of the LSTM ($\mathrm{LSTM}_g$) and the hidden states of $\mathrm{LSTM}_g$ at previous time step and current time step are correspondingly denoted as $h_g^{t-1}(i)$ and $h_g^t(i)$. At last, the DGCN-LSTM obtains each pedestrian's social interaction feature vector $h_g^{T_{\mathrm{obs}}}(i)$ at a certain level, and in the following sections, we denote $h_g^{T_{\mathrm{obs}}}(i)$ of the $k_{th}$ level as $H_g^k(i)$.

*3.7. MSFM.* There are $K$ levels in our graph, so there are $K$ DGCN-LSTMs and the node $v_i$'s social interaction feature vectors obtained by them can be denoted as $\left\{H_g^1(i), H_g^2(i), \ldots, H_g^K(i)\right\}$. We use an MSFM to fuse all levels' social interaction feature vectors of node $v_i$. The MSFM computes the weighted sum of $\left\{H_g^1(i), H_g^2(i), \ldots, H_g^K(i)\right\}$. The formulations are as follows:

$$\begin{aligned} \alpha_i^k &= \frac{\exp\left(h_{\mathrm{mo}}(i)^T H_g^k(i)\right)}{\sum_{j\in[1,K]}\exp\left(h_{\mathrm{mo}}(i)^T H_g^j(i)\right)}, \\ H_g(i) &= \sum_{k\in[1,K]} \alpha_i^k H_g^k(i). \end{aligned} \tag{4}$$

Here, $h_{\mathrm{mo}}(i)$ is the motion feature vector of pedestrian $p_i$, $\cdot^T$ represents transposition, $H_g^k(i)$ is the corresponding social interaction feature vector at level $k$, the fusion weight $\alpha_i^k$ is a scalar, and $H_g(i)$ is the final fused social interaction feature vector.

*3.8. Motion Decoder.* The motion decoder is used to predict future trajectories based on the motion features and the fused social interaction features. There are two types of motion decoders: motion decoders without noise and motion decoders with noise. The former makes the whole model a deterministic one, and the latter makes it a stochastic one. For the deterministic type, we only concatenate $H_g(i)$ and $h_{\mathrm{mo}}(i)$ as the initial hidden state of an LSTM and we train the model with L1 loss. For the stochastic type, we concatenate$H_g(i)$, $h_{\mathrm{mo}}(i)$, and a noise vector $z$ sampled from a standard Gaussian distribution to work as the initial hidden state of an LSTM. The formulation which shows how to get the initial hidden state of the stochastic motion decoder is as follows:

$$h_{\mathrm{de}}(i) = \mathrm{Linear}_h\left(\mathrm{concat}\left(H_g(i), h_{\mathrm{mo}}(i), z\right); W_h\right). \tag{5}$$

Moreover, we train the whole model with the variety loss proposed by SGAN [14] to encourage it to produce diverse samples. At the first prediction time step $T_{\mathrm{obs}} + 1$, the decoder gets $\Delta X_i^{T_{\mathrm{obs}}}$ as the initial input and predicts the next position offset $\Delta\widehat{X}_i^{T_{\mathrm{obs}}+1}$. The predicted position offset is marked as $\left\{\Delta\widehat{X}_i^t | t \in [T_{\mathrm{obs}}+1, T_{\mathrm{obs}}+T_{\mathrm{pred}}]\right\}$. The formulations which show how the stochastic motion decoder works are as follows:

$$\begin{aligned} h_{\mathrm{de}}^t(i) &= \mathrm{LSTM}_{\mathrm{de}}\left(h_{\mathrm{de}}^{t-1}(i), \mathrm{Linear}_{\mathrm{de}}\left(\Delta\widehat{X}_i^t; L_{\mathrm{de}}\right); W_{\mathrm{de}}\right), \\ \Delta\widehat{X}_i^{t+1} &= \mathrm{Linear}_{\mathrm{pred}}\left(h_{\mathrm{de}}^t(i); W_{\mathrm{pred}}\right), \\ \widehat{Y}_i^{t+1} &= \widehat{Y}_i^t + \Delta\widehat{X}_i^{t+1}, \end{aligned} \tag{6}$$

where $L_{\mathrm{de}}$ and $W_{\mathrm{pred}}$ are the trainable weights of the corresponding linear layers, concat means concatenating operation, and $W_{\mathrm{de}}$ denotes the trainable weights of the LSTM ($\mathrm{LSTM}_{\mathrm{de}}$).

## 4. Experiments

*4.1. Datasets, Baseline Methods, and Metrics*

*4.1.1. Datasets.* We evaluate our method on three commonly used datasets, ETH [35], UCY [36], and a high-density pedestrian dataset, pedestrian walk path dataset [37], which is referred to as PEDWALK in the rest of the article. ETH and UCY contain 1536 pedestrians' real-world trajectories, while PEDWALK contains the manually labeled trajectories of 12684 pedestrians, and coordinates are provided in pixels. The image size of PEDWALK is $1920 \times 1080$ pixels. ETH and UCY consist of a total of five unique scenes: ETH, HOTEL (from ETH), ZARA1, ZARA2, and UNIV (from UCY). For ETH and UCY, we follow the leave-one-out evaluation methodology in SGAN [14], training on 4 scenes and testing on the remaining one. For PEDWALK, we use 70% of its total frames for training and leave the remaining 30% for evaluation. The interval of trajectory sequences of ETH and UCY is 0.4 seconds, while the interval of trajectory sequences of PEDWALK is 0.8 seconds. We take 8 ground truth positions as observation and predict the trajectories of the following 12 time steps. It means, for ETH and UCY, we observe for 3.2 seconds and predict the future at 4.8 seconds (short-term prediction), while for PEDWALK, we observe for 6.4 seconds and predict the future at 9.6 seconds (long-term prediction).

*4.1.2. Baseline Methods.* We compare MDST-DGCN of deterministic type (MDST-DGCN-D) with deterministic models, e.g., LSTM [27], S-LSTM [11], Social Attention [15],

and CIDNN [19]. Furthermore, we compare MDST-DGCN of stochastic type (MDST-DGCN-S) with stochastic models, e.g., SGAN [14], SGAN-P [14], SoPhie [16], GAT [21], Social-BiGAT [21], STGAT [22], and Social-STGCNN [23].

*4.1.3. Metrics.* There are two commonly used metrics: average displacement error (ADE) and final displacement error (FDE). ADE is the average L2 distance between ground truth and the predicted trajectory over all the predicted time steps, and FDE is the distance between the predicted final position and the actual final position at the end of the prediction period $T_{\text{obs}} + T_{\text{pred}}$. For stochastic models, similar to prior work [14, 22], 20 samples are generated and the closest sample to the ground truth is selected to compute ADE and FDE. After checking the codes of SGAN, STGAT, and Social-STGCNN, we find there are two different ways to select the closest sample: selecting the closest trajectory of each pedestrian in a sample used by Social-STGCNN [23] and selecting the closest sample used by SGAN [14] and STGAT [22]. A sample includes all pedestrians' trajectories in the scenario for a total duration of $(T_{\text{obs}} + T_{\text{pred}})$ time steps. Following the tradition of SGAN and STGAT, we select the closest sample to compute the ADE and FDE of MDST-DGCN-S.

*4.2. Model Configuration and Training Details.* For the motion encoder, the output dimension of the linear layer is 32 and the hidden state dimension of $\text{LSTM}_{\text{mo}}$ is 64. For the MDST-DGEN, the output dimension of $f$ and $\text{LSTM}_g$ is 64. We implement $f$ with a convolution layer. To process nodes in different scenarios in parallel, the fixed neighbour number $m$ needs to be larger than the maximum number of pedestrians in a sample. The most crowded scene in PEDWALK contains 133 pedestrians, and in ETH and UCY, there are 57 pedestrians in the most crowded scene. So we set it 135 for PEDWALK and 60 for ETH and UCY. For the motion decoder, the output dimension of $\text{Linear}_h$ is 32, the hidden state dimension of $\text{LSTM}_{\text{de}}$ is 64, and the output dimension of $\text{Linear}_{\text{pred}}$ is 2. For the MDST-DGCN-S, the dimension of the noise vector $z$ is half of the hidden state dimension.

Our implementation is based on the PyTorch library. The model is trained on one NVIDIA GeForce GTX 1080Ti graphics card for 200 epochs. To calculate the variety loss with less GPU memory usage, we generate only 5 possible output predictions for each scene. In training, a batch size of 32 was used; we use the Adam optimizer with a learning rate of 0.0001. $\{1, 5, +\infty\}$ is the default-level distance list for ETH and UNIV, and $\{150, +\infty\}$ is the default-level distance list for PEDWALK.

*4.3. Quantitative Evaluation.* To validate the proposed MDST-DGCN, we present the prediction performance for both short-term trajectory prediction on ETH and UCY and long-term trajectory prediction on PEDWALK, and we present the prediction performance for various pedestrian densities. We elaborate on an ablation study to validate the effects of our multilevel graph and the aligning operation.

*4.3.1. MDST-DGCN-D.* As Table 1 shows, MDST-DGCN-D outperforms all deterministic methods and some stochastic methods on ETH and UCY. And, as Table 2 shows, MDST-DGCN-D even outperforms stochastic methods including STGAT. It shows that our model has good performance in capturing interaction features, and we think there are three reasons. First, PEDWALK has many more pedestrians in a scene than ETH and UCY, and then it has more interaction types and more frequent interaction activities in a sample. Second, high-density limits the randomness of pedestrian movement. Third, the prediction horizon on PEDWALK is 9.6 s, while it is 4.8 s on ETH and UCY. When the prediction horizon is short, lots of decisions in movement occur in the observation period and continue to the prediction stage, so lots of useful cues exist in pedestrians' motion features and it is not necessary to infer from interactive information. High density and long-term predictions enhance the impact of interactions on trajectory prediction, and high density reduces the effect of multimodality.

*4.3.2. MDST-DGCN-S.* As Tables 1 and 2 show, when the best sample of 20 predictions is selected to calculate ADE and FDE, MDST-DGCN-S outperforms all methods on PEDWALK and achieves comparable ADE and FDE with STGAT. The reasons why MDST-DGCN-S is not better than STGAT on ETH and UCY are the same as the reasons stated in (1). When the best trajectory of 20 predictions is selected, MDST-DGCN-S outperforms Social-STGCNN in ADE, but Social-STGCNN gets better FDEs in several subdatasets. It is mainly because there are accumulated errors when LSTM is used in our model.

*4.3.3. Various Pedestrian Densities.* Table 2 presents the results on the PEDWALK for various pedestrian densities. We use samples with the specified densities to make the comparison. With the increase in density, the performance of each method decreases. Both MDST-DGCN-D and MDST-DGCN-S outperform other methods for various pedestrian densities. When the density is low, such as $10 \le d \le 30$, the performance gap between SGAN and other methods is much smaller, which means when crowds are sparse, the effects of interactions are smaller and models get fewer useful cues to infer pedestrians' future movements, but the multimodality will work better. This phenomenon also confirms our previous reasoning in (1).

*4.3.4. Different Level Distance Lists.* Table 3 presents the ADEs and FDEs of MDST-DGCN-D with different level distance lists. The level distance list $\{+\infty\}$ means that MDST-DGCN-D models all social interactions at the same level, which is similar to STGAT and Social-STGCNN. Details about the level distance list are presented in Section 3. C. As shown in Table 3, modelling social interactions by a multilevel graph promotes the performance. On UNIV, the level distance list $\{1, +\infty\}$ helps MDST-DGCN-D to get the highest

Table 1: We compare deterministic baseline models with MDST-DGCN of deterministic type (MDST-DGCN-D) and stochastic baseline models with MDST-DGCN of stochastic type (MDST-DGCN-S) on ETH and UCY.

| Method | ETH | HOTEL | UNIV | ZARA1 | ZARA2 | AVG |
|---|---|---|---|---|---|---|
| LSTM [14, 27] | 1.09/2.41 | 0.86/1.91 | 0.61/1.31 | 0.41/0.88 | 0.52/1.11 | 0.70/1.52 |
| S-LSTM [11, 14] | 1.09/2.35 | 0.79/1.76 | 0.67/1.40 | 0.47/1.00 | 0.56/1.17 | 0.72/1.54 |
| Social Attention [15, 22] | 1.39/2.39 | 2.51/2.91 | 1.25/2.54 | 1.01/2.17 | 0.88/1.75 | 1.41/2.35 |
| CIDNN [19, 22] | 1.25/2.32 | 1.31/2.36 | 0.90/1.86 | 0.50/1.04 | 0.51/1.07 | 0.89/1.73 |
| MDST-DGCN-D | **0.86/1.75** | **0.44/0.90** | **0.55/1.16** | **0.40/0.86** | **0.32/0.68** | **0.51/1.07** |
| SoPhie*1 [16] | 0.70/1.43 | 0.76/1.67 | 0.54/1.24 | 0.30/0.63 | 0.38/0.78 | 0.54/1.15 |
| GAT*1 [21] | 0.68/1.29 | 0.68/1.40 | 0.57/1.29 | 0.29/0.60 | 0.37/0.75 | 0.52/1.07 |
| Social-BiGAT*1 [21] | 0.69/1.29 | 0.49/1.01 | 0.55/1.32 | 0.30/0.62 | 0.36/0.75 | 0.48/1.00 |
| SGAN*2 [14] | 0.81/1.52 | 0.72/1.61 | 0.60/1.26 | 0.34/0.69 | 0.42/0.84 | 0.58/1.18 |
| SGAN-p*2 [14] | 0.87/1.62 | 0.67/1.37 | 0.76/1.52 | 0.35/0.68 | 0.42/0.84 | 0.61/1.21 |
| STGAT*2 [22] | **0.65/1.12** | 0.35/0.66 | 0.52/**1.10** | 0.34/**0.69** | 0.29/0.60 | **0.43/0.83** |
| MDST-DGCN-S*2 | 0.69/1.45 | **0.34/0.58** | **0.51**/1.11 | **0.33**/0.70 | **0.28/0.59** | **0.43**/0.89 |
| Social-STGCNN*3 [23] | 0.64/**1.11** | 0.49/0.85 | 0.44/**0.79** | 0.34/**0.53** | 0.30/0.48 | 0.44/0.75 |
| MDST-DGCN-S3 | **0.56**/1.12 | **0.27/0.50** | **0.38**/0.81 | **0.27**/0.56 | **0.22/0.46** | **0.34/0.69** |

We predict future at 4.8 seconds (short-term prediction), given the previous 3.2 seconds. The errors reported are ADE or FDE in meters. Methods marked with $*$ draw 20 samples. The ADE and FDE of methods marked with superscript 2 are calculated by selecting the closest sample; the ADE and FDE of methods marked with superscript 3 are calculated by selecting the closest trajectory; and for the ADE and FDE of methods marked with superscript 1, we are not sure which type they belong to, because we cannot find their code. The values with the least error in the comparison model are bolded.

Table 2: ADEs and FDEs of different methods for long-term trajectory prediction on the PEDWALK with various pedestrian densities.

| Density (d) | $10 \le d \le 30$ | $30 \le d \le 50$ | $50 \le d \le 70$ | $70 \le d \le +\infty$ | Overall |
|---|---|---|---|---|---|
| SGAN $*$ | 35.57/70.39 | 44.02/87.08 | 43.30/85.84 | 47.34/93.24 | 44.02/86.96 |
| SGAN-P $*$ | 36.06/71.02 | 41.92/81.39 | 40.70/78.70 | 45.09/87.39 | 42.03/81.54 |
| STGAT $*$ | 33.20/60.21 | 38.06/68.25 | 38.33/69.18 | 41.97/75.98 | 39.02/70.47 |
| MDST-DGCN-D | **32.62**/63.05 | **36.38**/69.15 | **35.61/67.17** | **40.80**/77.77 | **37.31**/70.80 |
| MDST-DGCN-S $*$ | **30.53/57.88** | **34.62/64.81** | **34.68/64.81** | **39.75/75.21** | **35.99/67.53** |

The density (d) means the number of pedestrians in the scenario, and $D1 \le d \le D2$ means we select the samples in which the number of pedestrians is not less than D1 and not greater than D2. All methods predict 9.6 seconds, given the previous 6.4 seconds. Errors reported are ADE/FDE in pixels on the original size of $1920 \times 1080$. Methods marked with $*$ draw 20 samples and select the best sample. The values with the least error in the comparison model are bolded.

Table 3: Ablation study of MDST-DGCN-D with different level distance lists and with or without aligning operation for short-term prediction on ETH and UCY.

| Level distance list | ETH | HOTEL | UNIV | ZARA1 | ZARA2 | AVG |
|---|---|---|---|---|---|---|
| $\{+\infty\}$ | 0.870/1.792 | 0.490/1.011 | 0.626/1.273 | 0.407/0.867 | 0.333/0.703 | 0.545/1.129 |
| $\{1, +\infty\}$ | 0.862/1.772 | 0.465/0.989 | **0.532/1.139** | 0.402/0.862 | 0.324/0.687 | 0.517/1.090 |
| $\{5, +\infty\}$ | **0.853**/1.757 | 0.453/0.931 | 0.624/1.269 | 0.400/0.852 | 0.322/0.684 | 0.530/1.100 |
| $\{1, 5, +\infty\}$ $*$ | 0.859/**1.749** | **0.437/0.900** | 0.547/1.161 | 0.402/0.860 | 0.320/0.684 | **0.513/1.071** |
| Without aligning | 0.90/1.94 | 1.48/2.49 | 0.60/1.25 | **0.37/0.79** | **0.30/0.65** | 0.73/1.42 |

The level distance list $\{1, 5, +\infty\}$ $*$ is the default setting, and it is used for MDST-DGCN-D without aligning operation. The errors reported are ADE or FDE in meters. The values with the least error in the comparison model are bolded.

improvement. It is mainly because UNIV has a higher pedestrian density than the other four subdatasets, and more people will walk within one meter, the social comfort distance.

*4.3.5. Effects of Aligning Operation.* As shown in Table 3, the aligning operation advances the performance on ETH, HOTEL, and UNIV, but it reduces the performance on ZARA1 and ZARA2. Because ZARA1 and ZARA2 are collected in the same place and have the same coordinate system, when they are used separately as a test set, the model without aligning will overfit on the coordinates.

*4.4. Qualitative Evaluation.* We compare the predicted trajectories of MDST-DGCN-D and STGAT in Figure 3. Figure 3(a) shows that the target pedestrian is walking in the same direction with a nearby pedestrian A, and he will finally gather with a faraway pedestrian B, both of STGAT and MDST-DGCN-D.

We successfully predict the merging phenomenon. However, MDST-DGCN-D succeeds in predicting that the target pedestrian maintains his relative position with nearby pedestrian A, while STGAT does not. Thus, MDST-DGCN-D obtains more accurate predictions. As shown in Figure 3(b), two pedestrians in a group are changing their directions in advance to avoid collisions with the pedestrians standing in the distance. For the target pedestrian, MDST-DGCN-D assigns a weight of 0.72 to the social interaction feature of the third level, which helps avoid possible collisions with distant pedestrians. However, STGAT only successfully predicts group behaviour but fails to predict early collision avoidance behaviour. All predictions in Figure 3

| | (A) | (B) |
|---|---|---|
| scene | PEDWALK | ZARA2 |
| level distance list | {150, +∞} pixel | {1, 5, +∞} meter |
| fusion weight ($\alpha$) | 0.54, 0.46 | 0.14, 0.14, 0.72 |
| prediction |  | |



Figure 3: Qualitative comparison between MDST-DGCN-D and STGAT. For better visualization, only a few trajectories are drawn and we draw the area of each level at the last historical time step.

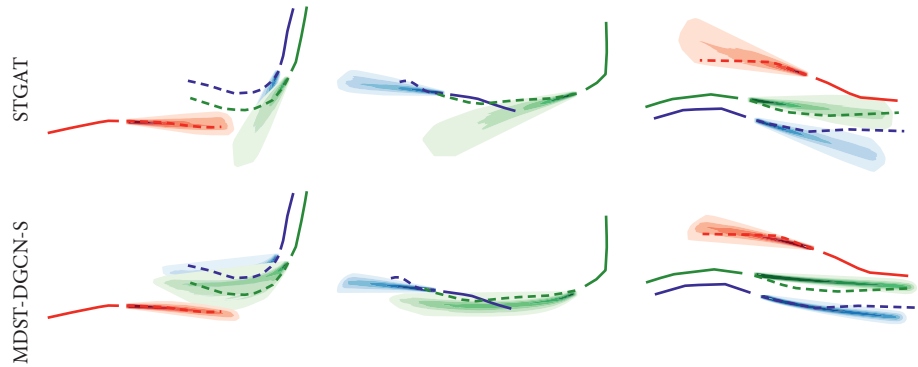


Figure 4: Trajectory distribution visualization for MDST-DGCN-S and STGAT. For better visualization, only a few trajectories are drawn.

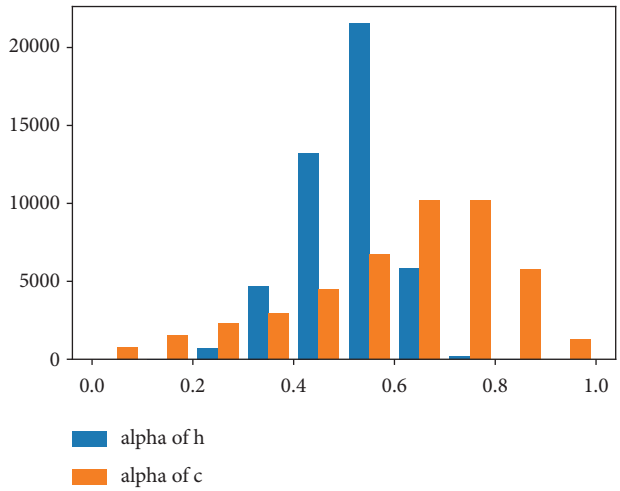


Figure 5: The distribution of fusion weight ($\alpha$). Alpha of h is the fusion weight of the hidden state, and alpha of c is the fusion weight of the cell state.

indicate that a multilevel graph structure can model social interactions more accurately and comprehensively.

We also visualize the trajectory distributions of MDST-DGCN-S and STGAT in Figure 4. As shown in Figure 4, in all three samples of pedestrian avoidance, pedestrian following, and pedestrian walking in group, our model outperforms STGAT.

We count the distribution of fusion weight ($\alpha$) on PEDWALK, which shows that the social interaction features of the first level and second level are of different importance in a sample. The distribution of fusion weight ($\alpha$) is shown in Figure 5.

## 5. Conclusions

In this article, we propose a multilevel dynamic spatio-temporal directed graph representation to model the interactions between pedestrians and introduce MDST-DGCN to process the multilevel graph. Experimental results indicate that our multilevel graph structure can model social interactions more accurately and comprehensively and show that MDST-DGCN outperforms most of the state-of-the-art methods.

## Data Availability

Previously reported ETH and UCY data were used to support this study and are available at https://doi.org/10.1109/ICCV.2009.5459260 and https://doi.org/10.1111/j.1467-8659.2007.01089.x. These prior studies are cited at relevant places within the text as references. Previously reported PEDWALK data were used to support this study and are available at https://doi.org/10.1109/CVPR.2015.7298971. The prior study is cited at relevant places within the text as references.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was in part supported by the Major Program of the National Natural Science Foundation of China (91938301) and the National Natural Science Foundation of China (62002345).

## References

[1] Y. Ma, X. Zhu, S. Zhang, R. Yang, W. Wang, and D. Manocha, "Trafficpredict: trajectory prediction for heterogeneous traffic-agents," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 6120–6127, 2019.

[2] M. Bennewitz, W. Burgard, G. Cielniak, and S. Thrun, "Learning motion patterns of people for compliant robot motion," *The International Journal of Robotics Research*, vol. 24, no. 1, pp. 31–48, 2005.

[3] N. Pradhan, T. Burg, and S. Birchfield, "Robot crowd navigation using predictive position fields in the potential function framework," in *Proceedings of the American Control Conference*, pp. 4628–4633, San Francisco, CA, USA, June 2011.

[4] Z. Chen, C. Song, Y. Yang et al., "Robot navigation based on human trajectory prediction and multiple travel modes," *Applied Sciences*, vol. 8, no. 11, 2018.

[5] C. Chinag and C. Ding, "Robot navigation in dynamic environments using fuzzy logic and trajectory prediction table," in *Proceedings of the International Conference on Fuzzy Theory and Its Applications (iFUZZY2014)*, vol. 99–104, Kaohsiung, Taiwan, November 2014.

[6] Y. Ma, D. Manocha, and W. W. Autorvo, "Local navigation with dynamic constraints in dense heterogeneous traffic," 2018, https://arxiv.org/abs/1804.02915.

[7] R. Mehran, A. Oyama, and M. Shah, "Abnormal crowd behavior detection using social force model," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 935–942, Miami, FL, USA, June 2009.

[8] D. Helbing and P. Molnár, "Social force model for pedestrian dynamics," *Physical Review A*, vol. 51, no. 5, pp. 4282–4286, 1995.

[9] L. Tian and C. Collins, "An effective robot trajectory planning method using a genetic algorithm," *Mechatronics*, vol. 14, no. 5, pp. 455–470, 2004.

[10] A. Richards and J. P. How, "Aircraft trajectory planning with collision avoidance using mixed integer linear programming," in *Proceedings of the American Control Conference*, vol. 3, pp. 1936–1941, Anchorage, AK, USA, May 2002.

[11] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social LSTM: human trajectory prediction in crowded spaces," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 961–971, Las Vegas, NV, USA, June 2016.

[12] N. Bisagno, B. Zhang, and N. Conci, "Group LSTM: group trajectory prediction in crowded scenarios," in *Proceedings of the European Conference on Computer Vision Workshops*, vol. 213–225, Munich, Germany, September 2018.

[13] I. Hasan, F. Setti, T. Tsesmelis, A. Del Bue, F. Galasso, and M. Cristani, "MX-LSTM: mixing tracklets and vislets to jointly forecast trajectories and head poses," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6067–6076, Salt Lake City, UT, USA, June 2018.

[14] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social GAN: socially acceptable trajectories with generative adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2255–2264, Salt Lake City, UT, USA, June 2018.

[15] A. Vemula, K. Muelling, and J. Oh, "Social Attention: modeling attention in human crowds," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 4601–4607, South Brisbane, Australia, May 2018.

[16] A. Sadeghian, V. Kosaraju, A. Sadeghian, N. Hirose, H. Rezatofighi, and S. Savarese, "SoPhie: an attentive gan for predicting paths compliant to social and physical constraints," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1349–1358, Long Beach, CA, USA, June 2019.

[17] J. Liang, L. Jiang, J. C. Niebles, A. G. Hauptmann, and L. FeiFei, "Peeking into the future: predicting future person activities and locations in videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5718–5727, Long Beach, CA, USA, June 2019.

[18] P. Zhang, W. Ouyang, P. Zhang, J. Xue, and N. Zheng, "SR-LSTM: state refinement for lstm towards pedestrian trajectory prediction," in *Proceedings of the IEEE Conference on*

*Computer Vision and Pattern Recognition*, Article ID 12077, Long Beach, CA, USA, June 2019.

[19] Y. Xu, Z. Piao, and S. Gao, "Encoding crowd interaction with deep neural network for pedestrian trajectory prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5275–5284, Salt Lake City, UT, USA, June 2018.

[20] J. Amirian, J. Hayet, and J. Pettr, "Social Ways: learning multi-modal distributions of pedestrian trajectories with GANs," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 2964–2972, Long Beach, CA, USA, June 2019.

[21] V. Kosaraju, A. Sadeghian, R. Mart´ın-Mart´ın, I. Reid, H. Rezatofighi, and S. Savarese, "Social-BiGAT: multimodal trajectory forecasting using Bicycle-GAN and graph attention networks," in *Advances in Neural Information Processing Systems*, pp. 137–146, Vancouver, Canada, 2019.

[22] Y. Huang, H. Bi, Z. Li, T. Mao, and Z. Wang, "STGAT: modeling spatial-temporal interactions for human trajectory prediction," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 6271–6280, Seoul, Korea (South), October 2019.

[23] A. Mohamed, K. Qian, M. Elhoseiny, and C. Claudel, "Social-STGCNN: a social spatio-temporal graph convolutional neural network for human trajectory prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Article ID 14412, Seattle, WA, USA, June 2020.

[24] L. Zhang, Q. She, and P. Guo, "Stochastic trajectory prediction with social graph network," 2019, https://arxiv.org/abs/1907.10233.

[25] B. Ivanovic and M. Pavone, "The Trajectron: probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs," in *Proceedings of the 2019 IEEE International Conference on Computer Vision*, pp. 2375–2384, Seoul, Korea (South), October 2019.

[26] W. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Advances in Neural Information Processing Systems*, pp. 1024–1034, Long beach, CA, USA, 2017.

[27] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[28] C. Yu, X. Ma, J. Ren, H. Zhao, and S. Yi, "Spatio-temporal graph transformer networks for pedestrian trajectory prediction," in *Proceedings of the European Conference on Computer Vision*, Glasgow, UK, August, 2020.

[29] A. Vaswani, N. Shazeer, N. Parmar et al., "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 6000–6010, Toronto, ON, Canada, June 2017.

[30] B. Joan, Z. Wojciech, S. Arthur, and L. Yann, "Spectral networks and locally connected networks on graphs," 2013, https://arxiv.org/abs/1312.6203.

[31] D. Michael, B. Xavier, and V. Pierre, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Advances in Neural Information Processing Systems*, pp. 3844–3852, Barcelona, Spain, 2016.

[32] T. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proceedings of the International Conference on Learning Representations*, Toulon, France, 2017.

[33] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," in *Proceedings of the International Conference on Learning Representations*, Vancouver, Canada, 2018.

[34] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, New Orleans, Louisiana, USA, 2018.

[35] S. Pellegrini, A. Ess, K. Schindler, and L. van Gool, "You'll never walk alone: modeling social behavior for multi-target tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 261–268, Kyoto, Japan, September 2009.

[36] L. Alon, C. Yiorgos, and L. Dani, "Crowds by example," *Computer Graphics Forum*, vol. 26, pp. 655–664, 2007.

[37] S. Yi, H. Li, and X. Wang, "Understanding pedestrian behaviors from stationary crowd groups," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3488–3496, Boston, MA, USA, June 2015.