TAGCN: Topology-Aware Graph Convolutional Network for Trajectory Prediction

Arindam Biswas[1] and Brendan Tran Morris[2](✉)

[1] Indian Institute of Technology Kharagpur, Kharagpur 721302, West Bengal, India
aribis369@iitkgp.ac.in
[2] University of Nevada, Las Vegas, NV 89154, USA
brendan.morris@unlv.edu

Abstract. Predicting future trajectories of agents in a dynamic environment is essential for natural and safe decision making of autonomous agents. The trajectory of an agent in such an environment not only depends on the past motion of that agent but also depends on its interaction with other agents present in that environment. To capture the effect of other agents on trajectory prediction of an agent, we propose a three stream topology-aware graph convolutional network (TAGCN) for interaction message passing between the agents. In addition, temporal encoding of local- and global-level topological features are fused to better characterize dynamic interactions between participants over time. Results are competitive compared to previous best methods for trajectory prediction on ETH and UCY datasets and highlights the need for both local and global interaction structure.

Keywords: Trajectory prediction · Graph embedding · Graph convolutional network

1 Introduction

Trajectory prediction is an essential task needed for autonomous agents moving in an environment where other agents are present. The autonomous agents should traverse safely in the presence of other agents and avoid any collision. Trajectory prediction of other agents in a crowded environment provides the autonomous agents valuable information for safe and smooth path planning. The most vulnerable agents in such environments are pedestrians, so their trajectory prediction is quite important to ensure their safety. In this paper we focus our study on the prediction of pedestrian trajectories.

Predicting the future path of agents is a challenging task as it incorporates many complications. The path of an agent is not just determined by the agent's previous track but also on the presence of obstacles in the environment as well, interactions with other agents, and on the goal of the agent [1]. Some earlier works in trajectory prediction used hand-picked features like social forces [2]

G. Bebis et al. (Eds.): ISVC 2020, LNCS 12509, pp. 542–553, 2020.
https://doi.org/10.1007/978-3-030-64556-4_42

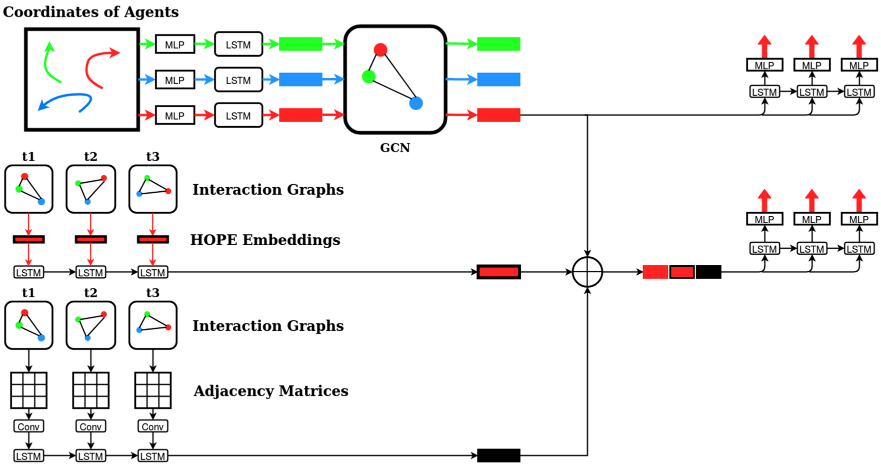


Fig. 1. Three stream TAGCN architecture. Stream1 (top) provides message passing between agents through a GCN for social interactions. Stream2 (middle) uses HOPE embedding to model local topology. Stream3 (bottom) encodes global topology. All streams are fused for trajectory prediction (Stream1 alone to reinforce training as an auxiliary task).

and energy potentials [3] to incorporate the interaction between the agents. Some prior work has also been done to incorporate the semantics of the environment in the model. Some recent methods use different types of maps like semantic segmentation maps [4] and rasterized maps [5] to get the semantics of the environment.

With the rise of deep learning based sequence encoders like Long-Short-Term-Memory (LSTM) networks [9] and Temporal Convolutional Network (TCN) [8], most works in this area use these networks for temporal encoding of the trajectories of agents. Much of the research has explored how to incorporate more context (e.g. social interactions with social pooling, graph convolutional network (GCN) message passing, scene structure with convolutional neural networks (CNNs), or semantic embedding) into the learning framework.

Some recent works also used multi-task learning for trajectory prediction, with trajectory prediction as the primary task. Jointly training a network for more than one task helps the model to learn more coherent knowledge from data and thus results in better predictions. Auxiliary tasks used in such models include agent behavior prediction [6], pose prediction [7], etc.

We propose a new topology-aware graph convolutional network (GCN) with LSTM decoder for pedestrian trajectory prediction which specifically considers not only the social influence of neighboring pedestrians but also their group evolution. The TAGCN framework, as shown in Fig. 1, utilizes three streams for 1) social interactions encoded by GCN, 2) dynamic local topology through HOPE [20] embeddings, and 3) global topology and structure evolution. Experimental results on standard pedestrian prediction datasets is competitive without the use of scene structure, attention, or stochastic prediction.

2 Related Research

To account for interactions among agents, many prior work have used social pooling [9], occupancy maps [10], attention mechanisms [11] and social tensors [12]. These methods do mot take into account the temporal nature of interaction among the agents (how interaction evolves over time). In very recent works [8,13–15], interaction networks between agents are built to model the influence of surrounding agents. These interaction networks are used to model the spatial interactions among agents. These interaction networks are also dynamic in nature so that they can model the temporal interactions between agents effectively. Some methods use these interaction networks for encoding the dynamic interactions between agents while others use them to share information among agents. To have message passing between the agents in an interaction network, recent works use graph neural networks like Graph Convolutional Network (GCN) [19] or Graph Attention Network (GAT) [18]. Both these networks have been shown to encode the influence on the other agents effectively for trajectory prediction. These networks can handle varying number of agents unlike some methods discussed previously. Since both of these networks have similar performance, we elect to use GCN for message passing in our model.

Many recent works either use the messaging between agents or use temporal encoding of the interactions among agents. But none take advantage of both these techniques, as per our knowledge. In our work, we take advantage of both the messaging passing between agents in the interaction network and also fuse in temporal encoding of the dynamic interaction network. Most works use the entire interaction network for temporal encoding of interactions which provides a global context in the predictions. But there should be more information about the local context of an agent in the interaction network. To get local context we use graph embeddings (HOPE node embeddings [20]) which tend to preserve the structural properties of the interaction network. Temporal encoding of these graph embeddings provide a local context for the evolving topology around an agent. Thus our model uses message passing between the agents as well as local and global topological context of the dynamic interaction graph of the agents.

Trajectory prediction of pedestrians is complicated since one of many plausible tracks can be selected. To counter this, some methods generate multiple predictions of future trajectories using General Adversarial Network (GAN) [16] and Variational Autoencoders (VAE) [17]. Producing multi-modal trajectories increases the chances of getting the an accurate prediction. These methods tend to perform better than deterministic models which produce only a single forecasted trajectory. In our work we only explore the deterministic results of our model. Our deterministic model outperforms some stochastic models and gives competitive results.

3 Topology-Aware GCN Trajectory Prediction Method

Trajectory prediction is a complex task that requires not only the past trajectory as an observation but needs to take into account intentions of other neighboring agents in a scene and the *dynamic interactions* among the agents. Our framework is a seq-seq model which includes message passing using GCN and it also includes temporal encoding of node and graph embeddings of the dynamic interaction graph of the agents. The use of GCN brings in awareness about the future intentions of the neighboring agents while the use of temporal encoding of node and graph embeddings brings in awareness about the changing topology of the interaction network at local and global context respectively.

3.1 Problem Definition

We are given spatial coordinates of a set of N pedestrians which are represented by $X_t^i = (x_t^i, y_t^i)$ where the subscript denotes the t-th time frame and the superscript denotes the i-th pedestrian. We observe each of these pedestrians for T_{obs} ($t = 1 : Tobs$) time-steps and predict the coordinates of these pedestrians for the next T_{pred} ($t = Tobs + 1 : Tobs + Tpred$) time-steps all at once. Our objective is to produce future trajectories very close to the ground truth future trajectories of the pedestrians.

3.2 Overall Framework

The overview of the TAGCN framework for trajectory prediction can be seen in Fig. 1. The encoder part of our seq-seq model has three streams, each encodes a different type of information for trajectory prediction.

The first stream (Stream1) of the encoder network does temporal encoding of past coordinates $X_{t=1:Tobs}^i$ of the target followed by message passing using GCN between neighbor agents. Temporal encoding of these coordinates are done using LSTM network. The coordinates $X_{t=1:Tobs}^i$ are first passed through a fully connected layer to encode the two-dimension coordinates to higher dimension representations e_t^i. This encoding is passed to LSTM network to produce the temporal encoding of coordinates h_t^i.

$$e_t^i = MLP^H(X_t^i, W_{MLP}^H) \tag{1}$$

$$h_t^i = LSTM^H(e_t^i, (h_{en}^H(i), W_{LSTM}^H)). \tag{2}$$

Here W_{MLP}^H and W_{LSTM}^H are trainable weights of the MLP^H and $LSTM^H$ networks. All the agents share weights for temporal encoding of coordinates.

The temporal encoding of coordinates of all the agents are then passed to the GCN layer. The agents are considered as nodes in the interaction network and their respective temporal encoding h_t^i as features of these nodes. This graph along with its node features are passed to the GCN layer to produces new aggregated features H^i for each node.

$$H^i = GCN((h_t^i, A_g), W_{GCN}). \tag{3}$$

Here W_{GCN} are the trainable weights of the GCN layer and A_g is the weight matrix of the interaction graph which is designed to have more contribution from closer agents.

The second stream (Stream2) of the encoder network does temporal encoding of the local topological context (grouping) for agents. To achieve this, the interaction graph at each time-step is used to form node embeddings n_t^i of each node/agent. These node embeddings are formed using the High-Order-Proximity-Preserving-Embeddings (HOPE)[20] algorithm which takes the weight matrix A_t of the interaction graph. These sequence of node embeddings $n_{t=1:Tobs}^i$ are then passed into LSTM network to produce temporal encoding L^i of these embeddings.

$$L^i = LSTM^L(n_t^i, (h_{en}^L(i), W_{LSTM}^L)). \tag{4}$$

Here W_{LSTM}^L are trainable weights of the $LSTM^L$ network. The feature L^i signifies the local topological context.

The third stream (Stream3) of the encoder network does temporal encoding of the global topological context of the entire interaction network (group interactions). To achieve this we represent the interaction network as weight matrices A_t at each time step. The weight matrices are passed through a $Conv$ network. The $Conv$ network consist of 2-D convolution layers and average-pooling layer, followed by a flatten layer to get a representation m_t of the entire interaction network.

$$m_t = Conv(A_t, W_{conv}). \tag{5}$$

Here W_{conv} are trainable weights of the entire $Conv$ network and A_t is the weight matrice of the interaction graphs. The sequence of graph embeddings $m_{t=1:Tobs}$ is passed through a LSTM network to produce temporal encoding G.

$$G = LSTM^G(m_t, (h_{en}^G, W_{LSTM}^G)). \tag{6}$$

W_{LSTM}^G are trainable weights of the $LSTM^G$ network. The feature G signifies the global topological context.

Features from each of the three streams is concatenated and fed to the decoder part of the seq-seq model to produce future coordinates of the agent. The definition of weight matrix A_g used in the first stream is different from the A_t that used in the second and third stream. Their specific definitions are discussed in Sect. 3.3. Our model also has an auxiliary decoder based only on the first stream. This decoder was added to reinforce the learning of the network as an auxiliary task.

We use LSTM network as decoder in the seq-seq model. Here W_{MLP}^{DP} and W_{LSTM}^{DP} are trainable weights of the MLP^{DP} and $LSTM^{DP}$ networks of primary decoder while W_{MLP}^{DA} and W_{LSTM}^{DA} are trainable weights of the MLP^{DA} and $LSTM^{DA}$ networks of auxiliary decoder.

$$X_{pred}^i = MLP^{DP}(LSTM^{DP}(HLG_i, h_{dec}^{DP}(i), W_{LSTM}^{DP}), W_{MLP}^{DP}) \tag{7}$$

$$X_{pred,H}^i = MLP^{DA}(LSTM^{DA}(H_i, h_{dec}^{DA}(i), W_{LSTM}^{DA}), W_{MLP}^{DA}) \tag{8}$$

where $HLG_i = \text{concat}(H_i, L_i, G)$. The loss of the entire network is calculated by summing up the losses of the primary and the auxiliary streams with equal weight.

3.3 Agent Interaction Network

Information sharing among agents and knowledge of their dynamic interactions are beneficial for predicting future trajectories in crowded environments. We build interaction networks to facilitate information sharing among agents and to learn about the dynamic interactions among neighboring agents. These graphs are dynamic as they change at every time-step. Our interaction graphs have undirected edges which are defined by the criteria that the distance between two edges should be less than a neighborhood distance threshold D.

We define the graph weight matrix A_g ($R^{N\times N}$) for the first stream as

$$A_g = \{a_{i,j}\} = \begin{cases} 1/d\left(X^i, X^j\right) & d\left(X^i, X^j\right) < D \\ 0 & \text{otherwise} \end{cases}. \tag{9}$$

Here $d(.)$ is the Euclidean distance function and X^i, X^j are coordinates of i-th and j-th agent. The weight matrix has stronger message passing for closer agents and weaker for further agents.

The second and third stream use the temporal adjacency matrix A_t ($R^{N\times N}$) as

$$A_t = \{\alpha_{i,j}\} = \begin{cases} d\left(X^i, X^j\right) & d\left(X^i, X^j\right) < D \\ 0 & \text{otherwise} \end{cases}. \tag{10}$$

The resulting weight matrix is defined to learn the topological features of the interaction network - e.g. the manner in which neighborhood topology changes over time. This helps to differentiate between scenarios such as traveling in a group or agents approaching/passing one another

3.4 Graph Convolutional Networks

Message passing between the neighboring agents has been shown to improve trajectory prediction by providing information about the intent (future state) of neighboring agents. In our framework, the message passing/information sharing is accomplished by using a GCN to take in node features and modify them based on the neighborhood interaction graph. Let F^0 be the input node features and A is the weight matrix of the interaction graph. The output features of the graph convolution is given by:

$$F^l = \sigma\left(D^{-0.5}\hat{A}D^{-0.5}F^{l-1}W^l\right) \tag{11}$$

where $\hat{A} = A+I$, D is the diagonal degree matrix of the interaction network, W^l are the trainable weights of the graph convolution layer and σ is the activation function. F^{l-1} are the input node features and F^l are the output node features. The identity matrix is added to the weight matrix as it helps to incorporate shared features from other nodes with it's own features.

3.5 Graph Embeddings

Temporal encoding of local and global topological context is used to learn dynamic evolution of interactions between agent. For representing the local topological context of an agent in the interaction network, we use HOPE node embeddings [20]. This algorithm produces embeddings that preserve the high order proximities (local structure) of the network. This algorithm is based on matrix factorization of the node proximity matrix. The embedding are formed by minimizing the following objective function:

$$\min ||S_{Katz} - U_s \cdot U_t^T||^2 \qquad \text{with} \qquad S_{Katz} = (I - \beta \cdot A)^{-1} \cdot \beta \cdot A. \quad (12)$$

Here S_{Katz} is the Katz Index which is a measure of high order proximity and β is a bias term. U_s and U_t are source and target embeddings.

For representing the global topological context we require whole-graph embeddings. Since techniques for whole graph embeddings are quite time consuming, we avoid their use and use the weight matrices A_t to represent whole interaction graphs. The weight matrices are further encoded using cheaper convolution layers and average pooling layers.

3.6 Implementation Details

The output size of the MLP layers is 16. The hidden state of all the LSTM units is of size 64. All the LSTM networks are single layer. We use two layers of GCN. We fix the size of node embedding to 40. We fix the value of parameter D of the interaction graph to be 10 m. We use Mean Square Error loss function with Adam optimizer. We use ReLu activations at all places. We trained our model for 200 epochs at a learning rate of 10^{-4}.

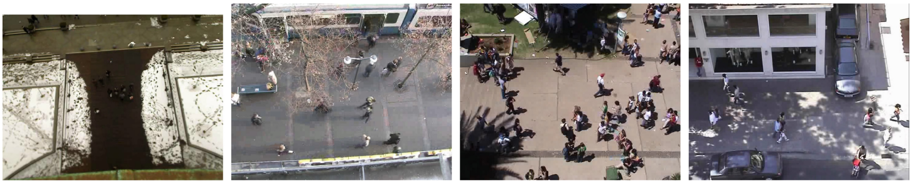
4 Experiment

We use ETH [20] and UCY [21] datasets for evaluating our model for trajectory prediction. ETH and UCY datasets have a total of five crowded scenes namely ETH, Hotel, Univ, Zara1 and Zara2 with a maximum of 51 pedestrians in a frame (Fig. 2). These datasets have a total of 1536 pedestrians (750 in ETH and 786 in UCY) performing complex interactions in real-world settings.

4.1 Evaluation Metrics

The models are evaluated following the leave-one-out policy [9,16], where the models are trained using all scenes except one (i.e four scenes) and tested on the remaining scene. The models observe 8 past time steps of a pedestrian and predict its trajectory for 12 future time steps. We calculate the individual scores for all scenes and their average for comparison amongst models.

Following the norms of previous works [9,23], we use standard metrics of Average Displacement Error (ADE) – mean L-2 distance error between predicted and real trajectory over all predicted time steps – and Final Displacement Error (FDE) – L-2 distance error between final predicted point and real trajectory – to evaluate the models.



(a) ETH (b) Hotel (c) Univ (d) Zara

Fig. 2. Standard Pedestrian Prediction Datasets. (a) ETH has mostly vertical trajectories and lots of stopped pedestrians. (b) Hotel has mostly horizontal trajectories with scene obstacles and pedestrians waiting for the train on the top. (c) Univ is a dense campus walkway with significant grouping. (d) Zara has mostly horizontal trajectories.

4.2 Baselines

We compare the performance of our model with several deterministic and stochastic models. We list the models used for comparison below:

- LSTM – This model uses only a simple LSTM network for temporal encoding of trajectory without using any contextual information.
- Social-LSTM [9] – This model uses social pooling along with LSTM network to provide social context to the model.
- CNN [25] – This model uses a fast temporal CNN encoding of a trajectory without using any contextual information.
- MATF [12] – This model uses scene context as well as social context using tensor fusion of these information.
- CVAE [26] – This is a stochastic model that uses a VAE to generate multiple plausible trajectories using an LSTM-based seq-seq model for temporal encoding of trajectories.
- SGAN [16] – This is a stochastic model that uses GAN to generate multiple plausible trajectories. It also uses social pooling for social context and LSTM for temporal encoding of trajectories.
- SoPhie [24] – This is a stochastic model that GAN for multiple predictions, social pooling for social context and LSTM for temporal encoding. In addition to these it also uses scene attention mechanism to get scene context.
- SAGCN [14] – This model uses a TCN for temporal encoding of trajectories as well as relative distance between agents. It also uses GCN for information sharing among agents.
- STGAT [13] – This model uses LSTM for temporal encoding and uses a GAT for crowd interaction modeling.

Table 1. ADE/FDE result comparison in meters

Method	ETH	Hotel	Univ	Zara01	Zara02	Average
LSTM	1.09/2.41	0.86/1.91	0.57/1.21	0.61/1.31	0.41/0.88	0.70/1.52
Social-LSTM [9]	1.09/2.35	0.79/1.76	0.67/1.40	0.47/1.00	0.56/1.17	0.72/1.54
CNN [25]	1.04/2.07	0.59/1.17	0.57/1.21	0.43/0.90	0.34/0.75	0.59/1.22
MATF [12]	1.33/2.49	0.51/0.95	0.56/1.19	0.44/0.93	0.34/0.73	0.64/1.26
CVAE [26]	0.93/1.94	0.52/1.03	0.59/1.27	0.41/0.86	0.33/0.72	0.53/1.11
SGAN [16]	0.81/1.52	0.72/1.61	0.60/1.26	0.34/0.69	0.42/0.84	0.58/1.18
SoPhie [24]	**0.70/1.43**	0.76/1.67	0.54/1.24	0.30/0.63	0.38/0.78	0.54/1.15
SAGCN [14]	0.90/1.96	**0.41/0.83**	0.57/1.19	**0.41/0.89**	0.32/0.70	**0.52**/1.11
STGAT 1V-1 [13]	0.88/1.66	0.56/1.15	**0.52/1.13**	0.41/0.91	**0.31/0.68**	0.54/1.11
GCN	1.06/2.10	0.71/1.44	0.57/1.24	0.46/0.98	0.35/0.80	0.63/1.30
GCN+Global	0.88/1.61	0.60/1.16	0.55/1.26	0.43/0.92	0.32/0.72	0.56/1.13
GCN+Local	0.90/1.73	0.61/1.18	0.56/1.28	0.43/0.90	0.32/0.72	0.56/1.16
TAGCN	0.86/1.50	0.59/1.15	0.54/1.25	0.42/0.90	0.32/0.71	0.55/**1.10**

4.3 Quantitative Results

As we can see in Table 1, our TAGCN model has results competitive with recent methods. Our model has the best average FDE score and comparable ADE score to the top performing comparison methods. Even though our model is deterministic, it performs better/comparable with some of the stochastic models (e.g. SGAN, CVAE, and SoPhie).

We performed an ablation study to understand the contributions of the different streams of the TAGCN model. GCN represents only Stream1, Stream1 + Stream2 = GCN+Local, and Stream1 + Stream3 = GCN+Global. The use of the auxiliary interaction network streams increases prediction accuracy. Incorporating global topological features gives a slight advantage over incorporating local topological features. Using both global and local topological features together further enhances the performance of our model.

Examples predictions can be seen in Fig. 3. The historical trajectory is in red, prediction in blue, and ground truth in green. TAGCN is able to correctly predict future trajectories of agents who maintain their direction to some extent, as we can see in the three horizontal trajectories, and does a good job providing stable prediction for the stopped pedestrian (Fig. 3a). However, it produces distorted predictions when the agent more aggressively changes its direction as we can see in the vertical trajectory in Fig. 3b. Note that without image overlay it is difficult to guess the reason for this direction change which is a limitation of the current model. Future work will incorporate scene context to better reason on static objects that require avoidance.

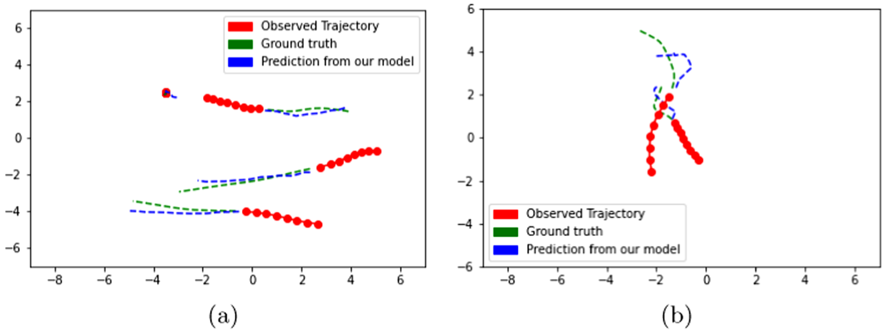


Fig. 3. Example predictions: observed trajectory (red), ground-truth (green), and TAGCN prediction (blue). (a) Strong prediction agreement for gradual maneuvers and stationary agents. (b) Poor prediction during more aggressive direction change. (Color figure online)

5 Concluding Remarks

In this work we present our model TAGCN which employs a LSTM based seq-seq model for human trajectory prediction. Our model uses GCN for information sharing amongst agents. It also uses node embeddings and graph embeddings to get local and global topological context of the evolving interactions among agents. Experimental results on the ETH and UCY datasets show our approach is quite competitive with recent pedestrian prediction networks. Further improvements could come from encoding scene context to handle agent-scene object in addition to agent-agent interactions.

References

1. Rudenko, A., Palmieri, L., Herman, M., Kitani, K.M., Gavrila, D.M., Arras, K.O.: Human motion trajectory prediction: a survey. arXiv preprint arXiv:1905.06113 (2019)
2. Helbing, D., Molnar, P.: Social force model for pedestrian dynamics.51, May 1998
3. Alahi, A., Ramanathan, V., Fei-Fei, L.: Socially-aware large-scale crowd forecasting. In: CVPR (2014)
4. Syed, A., Morris, B.T.: SSeg-LSTM: semantic scene segmentation for trajectory prediction. In: 2019 IEEE Intelligent Vehicles Symposium (IV), Paris, France, pp. 2504–2509 (2019). https://doi.org/10.1109/IVS.2019.8813801
5. Chou, F.-C., et al.: Predicting Motion of Vulnerable Road Users using High-Definition Maps and Efficient ConvNets (2019)
6. Chandra, R., et al.: Forecasting trajectory and behavior of road-agents using spectral clustering in graph-LSTMs (2019). ArXiv:1912.01118
7. Mangalam, K., Adeli, E., Lee, K.-H., Gaidon, A., Niebles, J.C.: Disentangling human dynamics for pedestrian locomotion forecasting with noisy supervision, pp. 2773–2782 (2020). https://doi.org/10.1109/WACV45572.2020.9093350

8. Wang, C., Cai, S., Tan, G.: GraphTCN: Spatio-Temporal Interaction Modeling for Human Trajectory Prediction. ArXiv, abs/2003.07167 (2020)
9. Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Li, F.-F., Savarese, S.: Social LSTM: human trajectory prediction in crowded spaces, pp. 961–971 (2016). https://doi.org/10.1109/CVPR.2016.110
10. Xue, H., Du, H., Reynolds, M.: SS-LSTM: a hierarchical lstm model for pedestrian trajectory prediction, pp. 1186–1194 (2018). https://doi.org/10.1109/WACV.2018.00135
11. Vemula, A., Muelling, K., Oh, J.: Social attention: modeling attention in human crowds. In ICRA pp. 1–7. IEEE (2018)
12. Zhao, T., et al.: Multi-agent tensor fusion for contextual trajectory prediction, pp. 12118–12126 (2019). https://doi.org/10.1109/CVPR.2019.01240
13. Yingfan, H., Bi, H., Li, Z., Mao, T., Wang, Z.: STGAT: modeling spatial-temporal interactions for human trajectory prediction (2019). https://doi.org/10.1109/ICCV.2019.00637
14. Sun, Y., He, T., Hu, J., Huang, H., Chen, B.: Socially-aware graph convolutional network for human trajectory prediction. In: 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chengdu, China, pp. 325–333 (2019). https://doi.org/10.1109/ITNEC.2019.8729387
15. Kosaraju, V., Sadeghian, A., Martín-Martín, R., Reid, I.D., Rezatofighi, S.H., Savarese, S.: Social-BiGAT: Multimodal Trajectory Forecasting using Bicycle-GAN and Graph Attention Networks (2019). ArXiv, abs/1907.03395
16. Gupta, A., Johnson, J., Li, F.-F., Savarese, S., Alahi, A.: Social GAN: socially acceptable trajectories with generative adversarial networks, pp. 2255–2264 (2018). https://doi.org/10.1109/CVPR.2018.00240
17. Bhattacharyya, A., Schiele, B., Fritz, M.: Accurate and diverse sampling of sequences based on a "Best of Many" sample objective (2018). https://doi.org/10.1109/CVPR.2018.00885
18. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph Attention Networks. In: International Conference on Learning Representations (2018)
19. Jiang, B., Zhang, Z., Lin, D., Tang, J., Luo, B.: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11313–11320 (2019)
20. Ou, M., Cui, P., Pei, J., Zhang, Z., Zhu, W.: Asymmetric transitivity preserving graph embedding, pp. 1105–1114 (2016). https://doi.org/10.1145/2939672.2939751
21. Pellegrini, S., Ess, A., Schindler, K., Van Gool., L.: You'll never walk alone: Modeling social behavior for multi-target tracking. In: 2009 IEEE 12th International Conference on Computer Vision, pp. 261–268. IEEE (2009)
22. Lerner, A., Chrysanthou, Y., Lischinski. D.: Crowds by example. In Computer Graphics Forum, vol. 26, pp. 655–664. Wiley Online Library (2007)
23. Lee, N., Choi, W., Vernaza, P., Choy, C.B., Torr, P.H.S., Chandraker, M.K. Desire: distant future prediction in dynamic scenes with interacting agents. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2165–2174 (2017)
24. Sadeghian, A., Kosaraju, V., Sadeghian, A., Hirose, N., Rezatofighi, H., Savarese, S.: SoPhie: an attentive GAN for predicting paths compliant to social and physical constraints. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1349–1358 (2019)

25. Nikhil, N., Morris, B.T.: Convolutional neural network for trajectory prediction. In: Leal-Taixé, L., Roth, S. (eds.) ECCV 2018. LNCS, vol. 11131, pp. 186–196. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-11015-4_16
26. Zhang, L., She, Q., Guo, P.: Stochastic trajectory prediction with social graph network. arXiv preprint arXiv:1907.10233 (2019)