Pedestrian Trajectory Prediction with Graph Neural Networks

Allan Wang (allanwan) [1] **Zirui Wang (ziruiw)** [1] **Wentao Yuan (wyuan1)** [1]

Abstract

Humans are capable of walking in a complex natural environment while cooperating with other stable or moving objects around. Being able to identify ones optimal moving path is a challenging task not only for humans but also for computers. Predicting the motion of pedestrians within a large system is an important problem with applications in autonomous driving and human-robot interaction. An accurate prediction can yield an optimal planning for the agent being controlled by the computer. The key to this problem is to model the complex interaction between people as well as other seen objects in the crowd. While traditional methods have utilized sequential analysis for complex time series data, some of them fail to utilize the interaction among pedestrians. In this project, we investigate the efficacy of graph neural networks, a new class of methods for interaction modeling, on the problem of pedestrian trajectory prediction.

1. Introduction

One of the goals of human-robot interaction is to enable trust between humans and robots. A key area in this domain is *social navigation* when a robot is maneuvering among pedestrians. Traditionally, mobile robots are unable to analyze the complex interactions among pedestrians, which often leads to the *freezing robot problem* (Trautman & Krause, 2010). Therefore, we attempt to analyze human navigation trajectories via prediction and empower a robot with similar behavioral capabilities in order to counter the freezing robot problem. This would not only open doors for better human-robot interaction, but also provides potentially better trafic planning tools for real-world traffic.

Humans have the innate ability to read and understand each other. When we walk in the public with other people around, we can easily prevent ourselves from hitting each other as if we can read others' minds such that we know where they are going. However, in fact, this "mind reading" is based on our understanding of a set of social rules as well as social relationships. For instance, one would give priority to senior citizens and/or people who need special care. Meanwhile, one may also try to stick together with one's friends. These different and complex common sense rules and social conventions jointly form the public crowd trajectory system.

However, predicting the motion of human while taking into account common sense behavior is a challenging problem. It requires the system being able to interpret surrounding environment as well as subtle connections among pedestrians. For instance, people will try to avoid hitting each other in the crowd from all directions or they will try to avoid huge amount of traffic at all cost. More subtle connections also involve relationships among pedestrians: people known each other will tend to walk closely together while two guys who don't know each other may try to keep a minimal safe distance. These all hidden rules guide how we as humans walk around in the larger system.

Previous methods (Alahi et al., 2016; Helbing & Molnár, 1995) have utilized different sequential data analysis to model the complex system being involved. While more old-school methods have tried to formulate the "social force" models mathematically (Helbing & Molnár, 1995), more recent approachs (Hochreiter & Schmidhuber, 1997) try to utilize sequential neural networks such as recurrent neural network (RNN) (Rumelhart et al., 1988) to handle this challenge. However, while prior work has focused on connecting each pedestrian with his/her nearby people (Alahi et al., 2016), we believe that the underlying inter-human relation graph, despite sparse, should not be restricted to neighbours only. In our project, we propose to use graph neural network (Kipf et al., 2018) to formulate connections among pedestrians and conduct trajectory prediction.

The rest of the paper will be organized as follows. We first give a thorough review of related work in the next section. In section 3, we details the proposed method together with its connection to the graph inference problem, followed by description of used dataset as well as experimental observations in section 4. Finally, we state our key observations in the conclusion as the last section.

[1]Carnegie Mellon University, Pittsburgh, PA 15213, USA.

10708 Class Project, Spring, 2019.

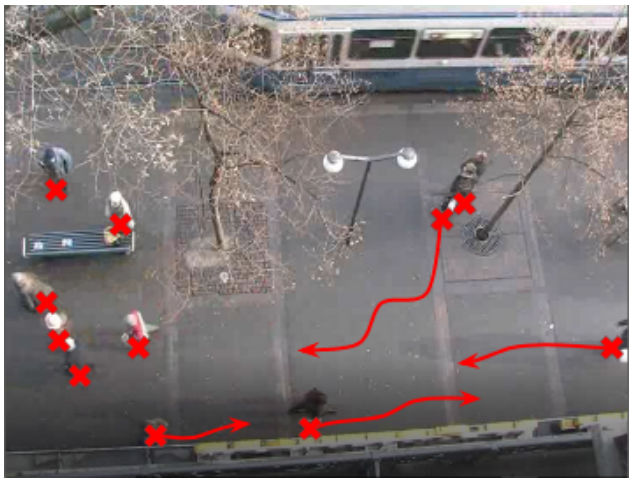

Figure 1. Predicting the the trajectories of pedestrians is challenging due to complex interactions among the crowd.

2. Related works

2.1. Time Series Analysis / Sequential Data

The time series analysis covers all sequential data with time being one of the dimensions. As time is sequential and mono-directional, these data often have dependencies on data of previous time steps. Some more traditional statistical approaches of such data includes the well known Autoregressivemoving-average model (Box & Pierce, 1970) which uses the data of prior time steps to predict for the future. The key is to construct a better weighting function for data from different time steps. Some other variants include exponential smoothing (Gardner Jr, 1985), Brown exponential smoothing, random process such as poisson-point process, and Autoregressivemoving-average model with exogenous inputs model (ARMAX) (Pham et al., 2010). These more traditional methods are still robust for many sequential data prediction tasks such as demanding forecasting.

More recent line of work of time series analysis tries to utilize the latest development of graph theory and neural networks to make robust prediction. These includes Hidden Markov Model (Fine et al., 1998), Kalman Filter (Welch et al., 1995), and Recurrent Neural Networks. These later development has been shown advantage of better capturing correlations among dimensions of data and outputs more efficient and meaningful graph inference.

2.2. Pedestrian Trajectory Prediction

Research in pedestrian behavior analysis can be dated back to two decades ago when (Helbing & Molnár, 1995) proposed the social force model. Social force models draw inspiration from physics principles and use the concept of attractive and repulsive forces to analyze the movement of pedestrians. Despite achieving early success, such simple models fail to account for the complex interpersonal interactions among pedestrians and result in inaccurate trajectory predictions.

Recent years, pedestrian trajectory problem has gained attention due to deep learning, with one of the most influential work being the Social-LSTM model (S-LSTM) proposed by (Alahi et al., 2016). S-LSTM initiated the trend of using Long Short Term Memory (LSTM) (Hochreiter & Schmidhuber, 1997), a type of Recurrent Neural Networks (RNN) (Rumelhart et al., 1988), to model the trajectory of every single pedestrian. To account for inter-pedestrian interactions, S-LSTM utilizes social pooling to combine the trajectory information of one pedestrian with those of the neighboring pedestrians. With careful training, S-LSTM is able to obtain significantly better prediction results when compared with the social force model. However, the social pooling neighbor only considers interactions with a few neighboring pedestrians and is computationally expensive.

To improve this, models developed by (Vemula et al., 2018) and (Varshneya & Srinivasaraghavan, 2017) proposed conditioning every single pedestrian's trajectories on all present pedestrian behavior, while using attention mechanism to decide which pedestrians to focus on. Attentions are weights assigned to the hidden layers of pedestrians' LSTM models. Other models such as SS-LSTM by (Xue et al., 2018) proposed to expand the input domain by further considering image patches around pedestrians. These images patches are encoded using Convolutional Neural Networks (CNN) (Krizhevsky et al., 2012) and the extracted features are further processed by LSTM networks. More recently, with the advent of Ganerative Adversarial Networks (GAN), (Gupta et al., 2018) proposed the Social-GAN model, which contains a LSTM-style encoder decoder network for the generator and another LSTM-style encoder network for the discriminator. Last but not least, the state-of-the-art SoPhie model developed by (Sadeghian et al., 2018) combined all three improvement proposals mentioned above. Despite great success, these models are all based on LSTM and can only operate on regular graph models such as images or location sequences. We believe our approach, as described below, can enable us to further explore inter-pedestrian relationships and yield more accurate trajectory predictions.

2.3. Relational Reasoning

Relational reasoning is the cornerstone of symbolic approaches to AI (Newell, 1980), where inference and predictions are made by reasoning about relations defined over a set of symbols using tools from logic and mathematics. It is known that symbolic AI suffers from the symbol

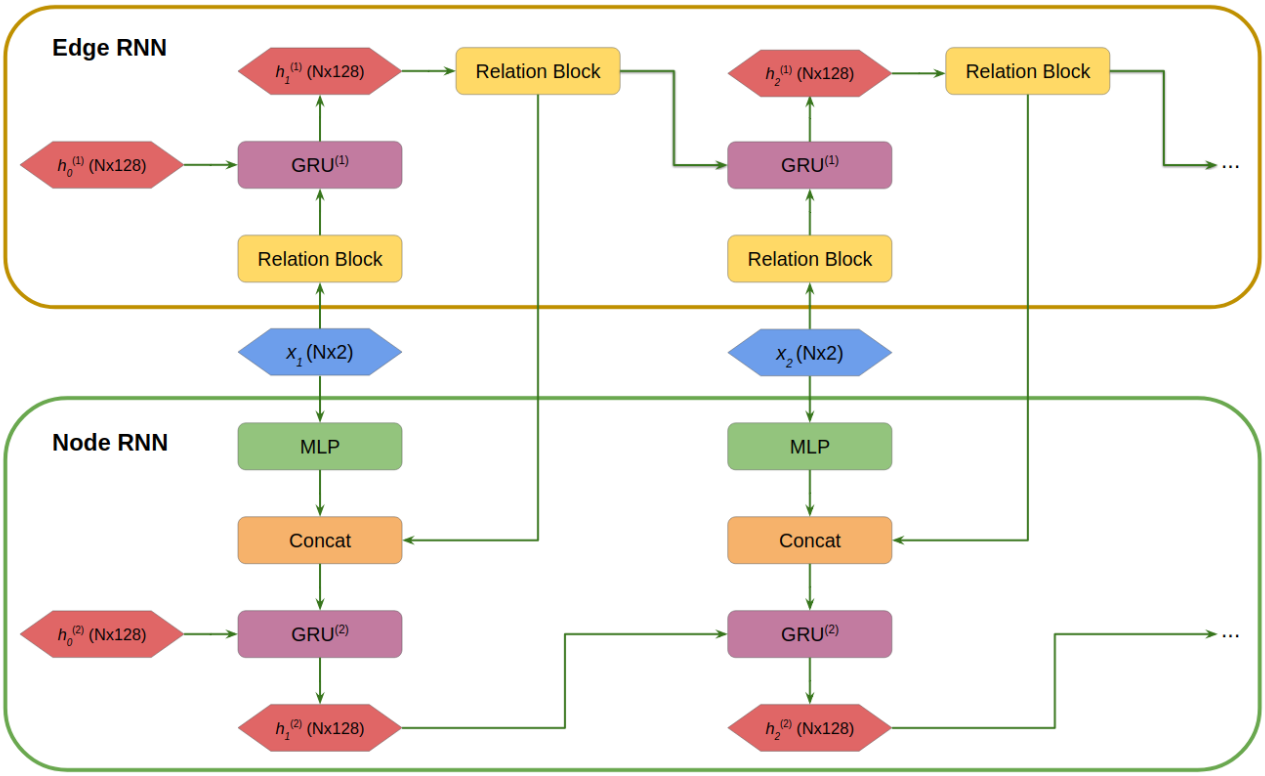

Figure 2. Architecture of Relational RNN.

grounding problem (Harnad, 1990) and thus is gradually replaced by statistical learning approaches (LeCun et al., 2015). However, it is difficult to learn complex relations from data without proper priors (Garnelo et al., 2016; Lake et al., 2017). Therefore, some recent works (Santoro et al., 2017; Kipf et al., 2018) start to explore the combination of relational reasoning with neural networks.

Graph Neural Networks (GNN) (Scarselli et al., 2009; Bruna et al., 2013; Li et al., 2015; Defferrard et al., 2016) is the main tool for relational reasoning using neural networks. GNN is a class of neural networks that operates on graph-structured data. Unlike Convolutional Neural Networks (CNN) (LeCun et al., 1995) and Recurrent Neural Networks (RNN) (Rumelhart et al., 1988) which typically operate on regular graphs such as images (2D grids) and sequences (1D grids), GNN is able to operate on graphs with more complex geometry and topology, such as 3D meshes, social networks and physical systems. The expressive power of GNNs has been demonstrated in various tasks such as classification of graph nodes (Kipf & Welling, 2016), semantic segmentation of 3D shapes (Yi et al., 2017) and modeling of interacting systems (Kipf et al., 2018).

Interaction Networks (IN) (Battaglia et al., 2016) is a particular type of GNN which models an interaction systems as a directed, complete graph, where each vertex encodes the state of an object and each edge encodes the relation between a pair of objects. Relational Networks (RN) (Santoro et al., 2017) generalizes IN to operator on feature maps of CNN and LSTM so that they can be trained jointly to perform tasks like visual question answering. Neural Relation Inference (NRI) (Kipf et al., 2018) further extends this framework by considering different types of interactions, which can be treated as an edge classification problem.

3. Method

3.1. Problem Formulation

We formulate pedestrian trajectory prediction as a sequence prediction problem. Specifically, given a set of pedestrian trajectories $\mathbf{x} = (\mathbf{x}^1, \ldots, \mathbf{x}^T)$, where $\mathbf{x}^t = (x_1^t, \ldots, x_N^t)$ denotes the locations of N pedestrians at time t, we would like to infer their positions for the next K time frame $(\mathbf{x}^{T+1}, \ldots, \mathbf{x}^{T+K})$.

The movement of a person depends on its own past trajectory as well as the trajectories of other people in the crowd. In order to model these relationships, we propose a neural network model which combines Recurrent Neural Networks (Chung et al., 2014) and Graph Neural Networks (Santoro et al., 2017), named Relational RNN (RRNN).

3.2. Relational RNN

As illustrated in Figure 2, the Relational RNN consists of two stacked Recurrent Neural Networks: Edge RNN, which is responsible for modeling the influence of other people in the crowd on the trajectory of a certain person, and Node RNN, which accounts for the dependency of the person's own past trajectory on its future movements.

At each time step t, the locations of the N pedestrians present in the current frame, $\mathbf{x}^t = \{\mathbf{x}_i^t\}_{i=1}^N$, is first passed into Edge RNN. Edge RNN processes the pedestrian locations via a relation block, which we will introduce in Sec. 3.3, and passes the processed features through a Gated Recurrent Unit (GRU) to obtain a hidden state $\mathbf{h}_t^{(1)}$. The hidden state is processed through another relation block and passed to Node RNN. Node RNN also takes the pedestrian locations $\mathbf{x}^t$ as input, but passes them independently through a Multi-layer perceptron (MLP) and concatenated the resulting features with the hidden state from Edge RNN. The concatenated features are passed through another GRU which gives a hidden state $\mathbf{h}_t^{(2)}$. If a prediction were to be made for the current time step, then a linear output layer is applied to $\mathbf{h}_t^{(2)}$.

With this stacked RNN architecture, we can effectively blend temporal and relational information. The information contained in the pedestrian's own trajectory, which is the major source of information for predicting its future movements, is modeled through Node RNN. However, the key difference between our model and an RNN that process each trajectory independently is the context information provided by Edge RNN. With the relation block, Edge RNN is able to summarize the interaction between each pair of pedestrians. This information can be helpful in cases where there are significant interaction between agents, e.g. two pedestrians running into each other will try to avoid collision. In cases where there is little or no interaction, the GRU in Node RNN can choose to ignore the context information from Edge RNN and focus on the information from each pedestrian's own past trajectory.

3.3. Relation Block

The relation block in Edge RNN is the key to modeling interactions among pedestrians. It is an instance of graph neural networks. To be specific, the input to the relation block is a set of "node embeddings", i.e. feature vectors which we associate to vertices in a graph. We represent interactions among people with "edge embeddings", i.e. features associated to edges in the graph. The two sets of embeddings communicate via the message passing operations introduced below, allowing the network block to model complex interactions among people in a crowd.

Figure 3 illustrates the structure inside the relation block.

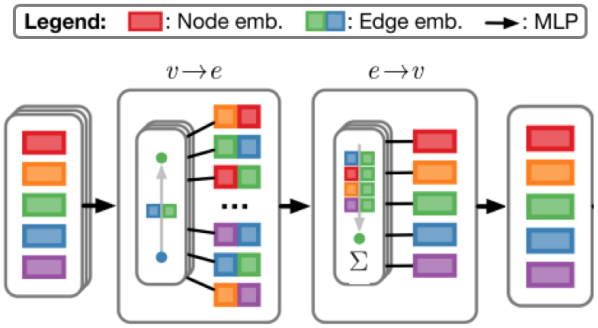

Figure 3. Illustration of the relation block, consisting of local message passing operations $v \to e$ and $e \to v$.

The key components of the relation block are two local message passing operations, defined as follows:

$$v \to e : \mathbf{u}_{i,j}^l = f_e^l([\mathbf{u}_i^l, \mathbf{u}_j^l]) \tag{1}$$

$$e \to v : \mathbf{u}_j^{l+1} = f_v^l\left(\sum\nolimits_{i \in \mathcal{N}_j} \mathbf{u}_{i,j}^l\right) \tag{2}$$

Here, $\mathbf{u}_i^l$ and $\mathbf{u}_{i,j}^l$ are the learned features of node v_i and edge $e_{i,j}$ at layer l. $\mathcal{N}_j$ denotes the set of nodes adjacent to j and $[\cdot,\cdot]$ denotes concatenation. f_e and f_v are neural networks defined on the edges and nodes respectively. These message passing operations convert between vertex and edge embeddings, which allow information to be distributed and aggregated along graph edges. With nonlinearities in f_e and f_v, the network can learn highly complex functions on the graph nodes and edges.

The message passing operations can be implemented as matrix multiplications, which makes it easy to integrate the operations into any neural network framework. In our experiments, we use fully-connected graphs as the number of pedestrians present in each frame is relatively small (on the order of 1 to 50). In the cases where the number of vertices become large, the relation block can also take a pre-defined graph or a nearest neighbour graph to keep the computation tractable. Empirically, we observed that 1 round of message passing ($l = 1$) is enough to model the interactions present in our datasets.

4. Experiments

4.1. Datasets

We conduct our study principally on two datasets, namely the ETH (Pellegrini et al., 2009) and UCY (Lerner et al., 2007) datasets. Both contains videos recording overhead view of walking pedestrians on the streets. The ETH dataset consists of two video recordings (named ETH, HOTEL) of duration 8.5 and 13 minutes respectively, with 750 pedestrians in total. The UCY dataset consists of three video record-

ings (named ZARA1, ZARA2, UCY) of duration 36s, 3.5 minutes, and 6 minutes respectively, with 785 pedestrians in total.

These datasets contain complex interactions among pedestrians such as grouping, crossing from different directions, overtaking, reacting to sudden movements etc. and are very representative of pedestrian behavior. Popular models such as (Alahi et al., 2016), (Gupta et al., 2018), (Vemula et al., 2018) all benchmark their performances on these datasets. Among the 5 video recordings, UCY has a high crowd density with as many as 52 pedestrians present at a single frame.

Along with image videos, the locations of all pedestrians are recorded. ETH and HOTEL recorded pedestrian locations in meters. ZARA1, ZARA2 and UCY recorded pedestrian locations in pixels. We performed a homographic estimation similar to (Gupta et al., 2018) to convert the pedestrian location units from pixels to meters. All videos are recorded at 25 FPS. However, we extract pedestrian information at 2.5 FPS to be consistent with (Alahi et al., 2016) and (Gupta et al., 2018). For every pedestrian, we performed bi-linear interpolations if pedestrian location information is missing at a particular frame.

Dataset	size	avg #ped	max #ped
ETH	234	8.1	25
HOTEL	445	6.4	19
ZARA1	688	9.5	51
ZARA2	989	7.7	38
UNIV	521	8.3	22

Table 1. Dataset Statistics

4.2. Experiment Setup

Evaluation Metrics. Similar to prior work (Alahi et al., 2016), (Gupta et al., 2018), (Sadeghian et al., 2018), we use two metrics to evaluate our model's accuracy:

- *Average Displacement Error (ADE):* The average L2 distance between the predicted location and the ground truth location over all prediction time steps.
- *Final Displacement Error (FDE):* The L2 distance between the predicted final destination and the ground truth final destination at the end of the prediction time period.

Baselines. We employ the following models as baselines that we hope to surpass with respect to the above proposed metrics.

- *Linear*: A simple linear regressor model that estimates linear parameters by minimizing the least square error (*ADE*).
- *LSTM*: A vanilla LSTM model with a LSTM sequence on every single pedestrian.
- *S-LSTM*: The popular Social-LSTM model proposed by (Alahi et al., 2016). This model builds on top of the vanilla LSTM by adding a social pooling layer that takes neighboring pedestrian behavior into account.

State-of-the-art models. Having realized how difficult the task is, we are no longer treating the following models as baselines. Instead, we would like to see how close our model can compete with these state-of-the-art models.

- *S-GAN*: A socially generative GAN model built on top of pedestrian LSTM sequences to predict trajectories as proposed by (Gupta et al., 2018).
- *S-GAN-P*: An advanced version of the Social-GAN model by adding a global social pooling layer to consider all present pedestrians before the decoding phase, also proposed by (Gupta et al., 2018).
- *SoPhie*: The current state-of-the-art model in pedestrian predictions proposed by (Sadeghian et al., 2018). On top of the Social-GAN model, this model takes a multi-model approach by taking in image patches around pedestrians as extra input information and also adds attention mechanisms to focus on relevant neighboring pedestrians.

Evaluation Methods. We evaluate our models in a similar fashion as (Alahi et al., 2016), (Gupta et al., 2018) and (Sadeghian et al., 2018). We use the leave-one-out approach by training on 4 sets and testing on the remaining set. When we use the term 'performance on dataset X', we imply the performance obtained by training our model on the other 4 datasets and evaluated on dataset X. We observe the trajectory for 8 time steps and predict trajectories for all present pedestrians for 12 time steps in the future.

Because the input state to our model requires a constant number of pedestrians, we first determine the maximum number of pedestrians present in a given frame across all training datasets, then we use this as the number of input states to our model. We index pedestrians and use -1 as a placeholder for nonexistent pedestrians in order to fill up the input states. Moreover, We ignore pedestrians who do not have complete information during both the observation and prediction time periods. In other words, if a pedestrian enters the frame late or exits the frame early, it will not be counted. Upon close inspection of the state-of-the-art models' codes, we discover that this preprocessing step is consistent with the other approaches.

4.3. Quantitative Examination

Evaluation Metrics The first quantitative analysis would be mainly focus on the two evaluation metrics being used, namely ADE and FDE. As shown in Table 2 (the lower the better), we can see that linear model performances are

Metric	Dataset	Linear	LSTM	S-LSTM	S-GAN	S-GAN-P	SoPhie	Ours
ADE	**ETH**	1.33	1.09	1.09	0.81	0.87	**0.70**	0.99
	HOTEL	0.39	0.86	0.79	0.72	0.67	0.76	**0.41**
	UNIV	0.82	0.61	0.67	0.60	0.76	**0.54**	0.72
	ZARA1	0.62	0.41	0.47	0.34	0.35	**0.30**	0.61
	ZARA2	0.77	0.52	0.56	0.42	0.42	**0.38**	0.44
AVG		0.79	0.70	0.72	0.58	0.61	**0.54**	0.63
FDE	**ETH**	2.94	2.41	2.35	1.52	1.62	**1.43**	1.90
	HOTEL	0.72	1.91	1.76	1.61	1.37	1.67	**0.71**
	UNIV	1.59	1.31	1.40	1.26	1.52	**1.24**	1.48
	ZARA1	1.21	0.88	1.00	0.69	0.68	**0.63**	1.03
	ZARA2	1.48	1.11	1.17	0.84	0.84	**0.78**	0.81
AVG		1.59	1.52	1.54	1.18	1.21	1.15	1.19

Table 2. Quantitative results of our model, the baseline models and our state-of-the-art models with prediction of 12 future time steps. Error reported are ADE and FDE in meters.

generally not on par with other models because it is unable to model the complex social interactions among pedestrians. Social-LSTM models step up the performance by taking neighboring pedestrians into account via social pooling. Despite it has been shown to have significant qualitative pedestrian behavior interpretations in (Alahi et al., 2016), we were unable to make Social-LSTM defeat the vanilla LSTM baseline (similar to (Gupta et al., 2018)). Social-GAN models are a further step up by approaching the task from a generative perspective. The SoPhie model, being the most sophisticated model by combining various proven architecture modules together, proves to output the most accurate results within all baselines that we used. Besides, we also observe that the results of the two evaluation metrics, ADE and FDE, are consistent across all datasets.

Due to the time constraint of this project. We have yet to fine tune the hyper-parameters of our model. The performance of our model is shown at the last column in Table 2. We can see that our model, similar to other models, performed worst on the ETH dataset. Pedestrians in the ETH dataset move vertically. Due to the fact that our dataset is small, moving patterns of pedestrians can pose a significant bias in our dataset. We can also see that our model does not perform well on the UNIV dataset. This is because the UNIV dataset is the most crowded dataset with extremely complicated interactions. Other approaches also do not perform well on the UNIV dataset.

When comparing our model's performance with other approaches, we can see that our model performed equal or better than our target baseline models (Linear, LSTM, S-LSTM). On both ADE and FDE, our model's performance only falls slightly short on the UNIV and ZARA1 datasets, but beats the baseline models on the other three datasets. When comparing our model to the state-of-the-art models, our model performs worse on the ETH and ZARA1 datasets, and is on par with the other models on the UNIV and ZARA2 datasets. Most surprisingly, our model achieves the best performance on the HOTEL dataset, even surpassing all state-of-the-art dataset performances. As a result, we conclude that we have achieved satisfactory progress in this course project.

4.4. Ablation study

Metric	Dataset	n-RNN	e-RNN	Combined
ADE	**ETH**	1.09	1.13	0.99
	HOTEL	0.86	1.01	0.41
	UNIV	0.61	1.17	0.72
	ZARA1	0.41	0.75	0.61
	ZARA2	0.52	0.55	0.44
AVG		0.70	0.92	0.63
FDE	**ETH**	2.41	2.15	1.90
	HOTEL	1.91	2.20	0.71
	UNIV	1.31	2.59	1.48
	ZARA1	0.88	1.59	1.03
	ZARA2	1.11	1.17	0.81
AVG		1.52	1.94	1.19

Table 3. Ablation study results of our model. n-RNN is the pure node-RNN model. e-RNN is the pure edge-RNN model. n-RNN performances are the same as the vanilla LSTM model.

In this section we present our ablation study on our overall model. As mentioned in the Method section, our model consists of two RNNs, the edge-RNN and the node-RNN. We present our model's ablation study by studying the model's pure edge-RNN form and pure node-RNN form respectively.

The pure node-RNN is essentially a GRU RNN model on each pedestrian. Thus, it is similar in performance to the

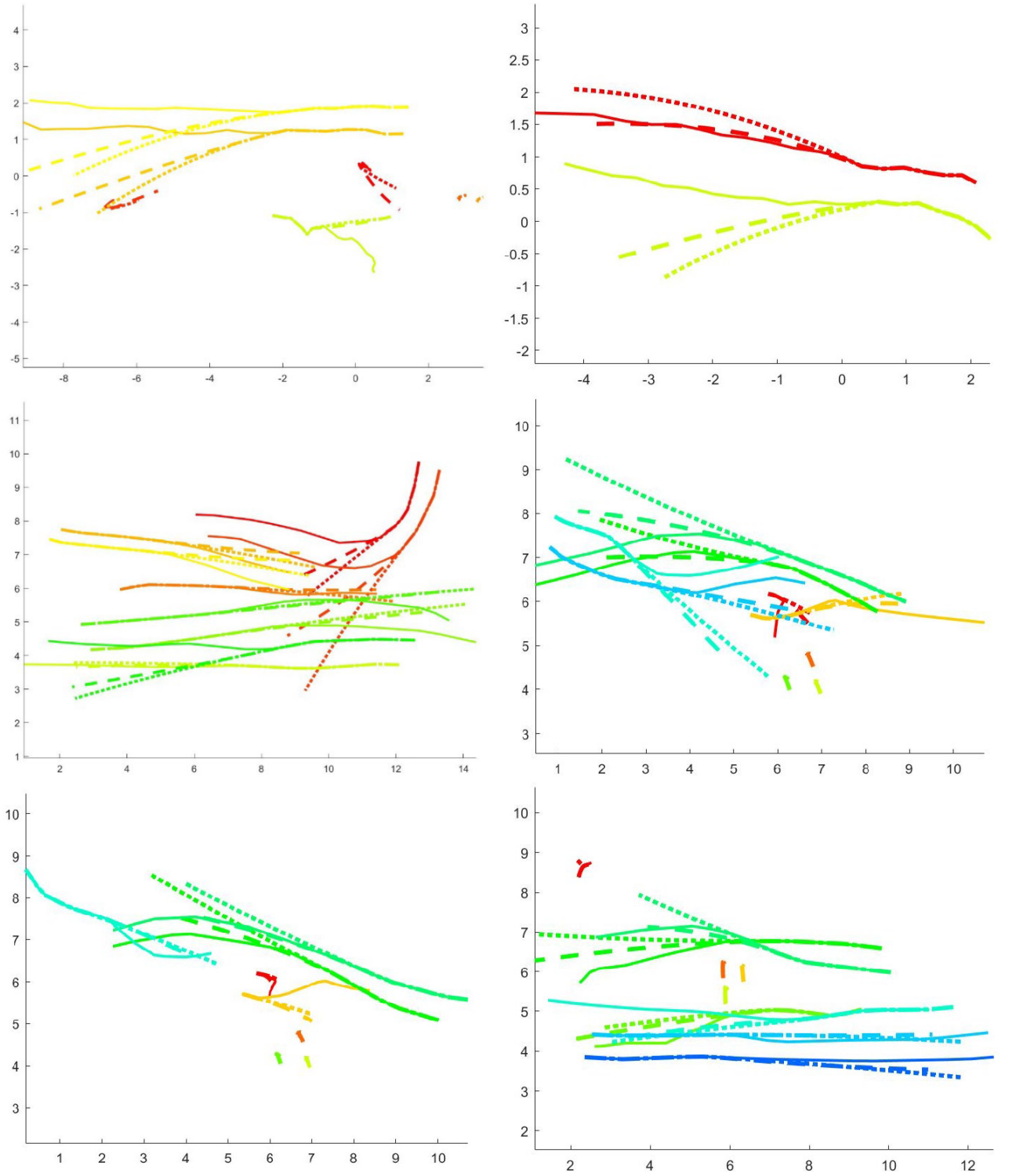

Figure 4. Qualitative results of our model vs the LSTM baseline model. The solid lines represent the ground truth trajectories. The dashed lines represent the trajectories predicted by our model. The dotted lines represent the trajectories predicted by the LSTM baseline model. Different colors represent different pedestrians. The first 8 frames of each trajectory are the observation trajectories and remain the same for all three cases. Our model's predicted lines are closer to the ground truth lines.

vanilla LSTM model. In our experiments, we found similar performances between the vanilla LSTM model and the vanilla GRU model. Here we report the vanilla LSTM model results. The pure edge-RNN model is a model that solely replies on the relationships among each pair of pedestrians.

We can see that the pure node-RNN (LSTM) model performs better than the pure edge-RNN model. This suggests that the temporal information captured by node-RNN is more important to predicting future trajectories than the context information captured by edge-RNN by itself. However, when we combine edge-RNN and node-RNN, we can see that our model imporved 10% on average ADE and 21.7% on average FDE. This suggests that the inter-pedestrian relationships captured by edge-RNN is indeed helpful and complement the temporal information captured by node-RNN, which results in superior performance.

4.5. Qualitative Examination

Some qualitative results are shown in Figure 4. We can observe a few interesting points comparing the ground truth with paths generated by our model as well as the baseline LSTM model. We use LSTM as our baseline model to compare with because it shows better performance than the Social-LSTM model.

First, we can observe that our model is able to generate more accurate path, closer to the ground truth, than the baseline model in the more straight forward case. We can see this from the top two graphs shown in 4, in which only a few pedestrians are involved in the traffic. Our generated path is closer, in some cases even completely overlap with the ground truth path, while the baseline LSTM will generate path that quickly digress from the ground truth even in the most simple two pedestrian scenarios.

Second, in a more complex system as shown in the middle two graphs, our model tends to try following previous path as much as possible, yielding sub-optimal results. For instance, the blue pedestrian made a huge turn in the graph on the right in the middle, but our generated path, together with the one generated by the baseline LSTM model, still follows the straight line of where he/she used to head to. On the other hand, this could also provide benefits as we can see that our generated paths are closer to those cases where pedestrians are less affected by the crowd, such as those green ones on the left.

Third, our model sometimes fail to avoid the collision as shown in the figures. For instance, the two blue lines actually cross each other in the right graph in the middle. However, this might due to the fact that they can cross the same point at different time steps so that a collision can be avoided. We can also observe similar behaviors of the baseline model but overall the generated path avoid collision well.

These qualitative examinations shown above demonstrate that our model is able to model the inter-pedestrian interactions to a certain degree, thus predicting trajectories that are closer to the ground truth compared to those generated by the LSTM baseline, which has no consideration of pedestrian interactions.

5. Discussion and Future Work

We have proposed a method for predicting the future trajectories of a group of pedestrians given their past trajectories. Our method leverages recent developments in graph neural networks to account for interactions among pedestrians. We evaluated our methods on standard benchmarks and showed that our methods achieve state-of-the-art performance on certain sequences. Our results show evidences that reasoning about interactions is indeed important in predicting movements of people in an social environment.

However, our method is only an initial attempt towards relational reasoning in predicting social movements. There are several key limitations which leave room for future work.

First, the inputs to our network are raw pedestrian locations, which contain very limited information. Inputs that contain much richer information such as image patches from the captured video sequence or even depth measurements from 3D sensors can potentially boost the performance of our model.

Second, the relational block treats every interaction in the same way as the parameters in the multi-layer perceptron that operate on edge embeddings are shared across all edges. Nonetheless, in reality there are many types of different interactions. For instance, a couple may be walking very close to each other while strangers will try to keep a distance from each other. A more sophisticated approach would be to first classify the type of interaction and then use different networks to process the edge embeddings of different interaction types.

Finally, self-attention models have turned out to be a strong alternative to recurrent models in many tasks that require sequence prediction such as machine translation. Thus, replacing the GRU in our proposed model with a self-attention mechanism is a promising direction tot pursue, which may lead to networks that can model interactions for longer time periods, as self-attention does not suffer from vanishing or exploding gradients. However, one bottleneck is the large amount of data and computation required for training self-attention models, which we do not have access to during the course of this project.

References

Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., and Savarese, S. Social lstm: Human trajectory prediction in crowded spaces. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Battaglia, P., Pascanu, R., Lai, M., Rezende, D. J., et al. Interaction networks for learning about objects, relations and physics. In *Advances in neural information processing systems*, pp. 4502–4510, 2016.

Box, G. E. and Pierce, D. A. Distribution of residual autocorrelations in autoregressive-integrated moving average time series models. *Journal of the American statistical Association*, 65(332):1509–1526, 1970.

Bruna, J., Zaremba, W., Szlam, A., and LeCun, Y. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.

Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

Defferrard, M., Bresson, X., and Vandergheynst, P. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*, pp. 3844–3852, 2016.

Fine, S., Singer, Y., and Tishby, N. The hierarchical hidden markov model: Analysis and applications. *Machine learning*, 32(1):41–62, 1998.

Gardner Jr, E. S. Exponential smoothing: The state of the art. *Journal of forecasting*, 4(1):1–28, 1985.

Garnelo, M., Arulkumaran, K., and Shanahan, M. Towards deep symbolic reinforcement learning. *arXiv preprint arXiv:1609.05518*, 2016.

Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S., and Alahi, A. Social gan: Socially acceptable trajectories with generative adversarial networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Harnad, S. The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42(1-3):335–346, 1990.

Helbing, D. and Molnár, P. Social force model for pedestrian dynamics. *Phys. Rev. E*, 51:4282–4286, May 1995.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997. ISSN 0899-7667.

Kipf, T., Fetaya, E., Wang, K.-C., Welling, M., and Zemel, R. Neural relational inference for interacting systems. *arXiv preprint arXiv:1802.04687*, 2018.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

Lake, B. M., Ullman, T. D., Tenenbaum, J. B., and Gershman, S. J. Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40, 2017.

LeCun, Y., Bengio, Y., et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.

LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *nature*, 521(7553):436, 2015.

Lerner, A., Chrysanthou, Y., and Lischinski, D. Crowds by example. *Computer Graphics Forum*, 26(3):655–664, 2007.

Li, Y., Tarlow, D., Brockschmidt, M., and Zemel, R. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493*, 2015.

Newell, A. Physical symbol systems. *Cognitive science*, 4 (2):135–183, 1980.

Pellegrini, S., Ess, A., Schindler, K., and van Gool, L. You'll never walk alone: Modeling social behavior for multi-target tracking. In *2009 IEEE 12th International Conference on Computer Vision*, Sept 2009.

Pham, H. T., Yang, B.-S., et al. A hybrid of nonlinear autoregressive model with exogenous input and autoregressive moving average model for long-term machine state forecasting. *Expert Systems with Applications*, 37 (4):3310–3317, 2010.

Rumelhart, D. E., Hinton, G. E., Williams, R. J., et al. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1, 1988.

Sadeghian, A., Kosaraju, V., Sadeghian, A., Hirose, N., Rezatofighi, S. H., and Savarese, S. SoPhie: An Attentive GAN for Predicting Paths Compliant to Social and Physical Constraints. *arXiv e-prints*, pp. arXiv:1806.01482, Jun 2018.

Santoro, A., Raposo, D., Barrett, D. G., Malinowski, M., Pascanu, R., Battaglia, P., and Lillicrap, T. A simple neural network module for relational reasoning. In

Advances in neural information processing systems, pp. 4967–4976, 2017.

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009.

Trautman, P. and Krause, A. Unfreezing the robot: Navigation in dense, interacting crowds. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 797–803, Oct 2010. doi: 10.1109/IROS.2010.5654369.

Varshneya, D. and Srinivasaraghavan, G. Human Trajectory Prediction using Spatially aware Deep Attention Models. *arXiv e-prints*, pp. arXiv:1705.09436, May 2017.

Vemula, A., Muelling, K., and Oh, J. Social attention: Modeling attention in human crowds. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–7, May 2018.

Welch, G., Bishop, G., et al. An introduction to the kalman filter. 1995.

Xue, H., Huynh, D. Q., and Reynolds, M. Ss-lstm: A hierarchical lstm model for pedestrian trajectory prediction. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1186–1194, March 2018.

Yi, L., Su, H., Guo, X., and Guibas, L. J. Syncspeccnn: Synchronized spectral cnn for 3d shape segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2282–2290, 2017.