Predicting Human Trajectories with LSTM using an Adaptive Attention Framework

Project

Presented in Partial Fulfillment of the Requirements for the Degree Master of Science in the Graduate School of The Ohio State University

By

Debanjan Nandi,

Graduate Program in Department of Computer Science and Engineering

The Ohio State University

July 18, 2018

Project Committee:

Dr. James W. Davis, Advisor

Dr. Eric Fosler-Lussier

Abstract

An unprecedented growth in efforts towards building autonomous vehicles and social robots over the last couple of years in human-centric environments has redefined the importance of understanding human behavior. It has now become more imperative than ever before, to understand and develop models which can understand complex, cooperative interactions between people in a crowd so that the autonomous systems can charter a safe , efficient, and socially compliant trajectories towards their destination. Previous approaches to human trajectory prediction have modeled the interactions between humans only as a function of their proximity. However, that may not be always be true. People located in our immediate vicinity and moving away from us will affect our trajectory lesser than people from the same vicinity and moving towards us, since we might collide with the latter in future. In this work, we present an approach which predicts future trajectories of people in a crowd using a data-driven architecture. We use a feed-forward, fully differentiable, and jointly trained recurrent neural network (RNN) mixture model augmented with a novel pedestrian weighting scheme to model trajectories of all humans in the crowd. Our integrated attention module has the flexibility to adapt its neighborhood of influence based on the pedestrian's behavior, and it learns the attention from the data itself. We demonstrate the performance of our model on two publicly available data-sets, and show that our model outperforms the baseline at prediction of future trajectories.

To my parents, for their unwavering support

Acknowledgments

I would first like to thank my project advisor Dr. James W. Davis, for agreeing to guide and advise me for the Master's Project. I feel privileged to have worked and learned so much under his tutelage. His door was always open whenever I ran into a troubled spot or had a question about my research or academics. He consistently allowed this project to be my own work, but steered me in the right direction whenever he thought I needed so.

I would also like to thank Dr. Eric Fosler-Lussier, at The Ohio State University as the second reader of this thesis, and I am gratefully indebted to him for his very valuable comments on this thesis.

I would also like to thank my peers in the Computer Vision Lab, The Ohio State University for their help whenever I needed them. It was a pleasure to work alongside them, and learn so many valuable things from them everyday. I feel lucky to have worked amidst such an amazing talented group of students. It was great sharing the laboratory with all of you for the last one and half year.

And finally, I would like express my profound gratitude to my friend Adyasha Maharana for her constant support since our undergraduate days together. This would not have been possible without her selfless compassion. I shall forever cherish our late night discussions on the details of my project, and be indebted to her for being a patient listener. Thank you.

Lastly, a shout out to all those ML/AI researchers who share their code, and help keep the wheels of innovation moving forward.

Vita

February 28, 1993	Born - Bankura, WB, India
August 2015	Integrated B.Tech. and M.Tech. Electronics and Electrical Communications Engineering, Indian Institute of Technology, Kharagpur, India.
August 2016 - Present	M.S. Computer Science and Engineering, The Ohio State University, Columbus, USA

Fields of Study

Major Field: Computer Science and Engineering

Table of Contents

Page

Abstract ii

Dedication iii

Acknowledgments iv

Vita v

List of Tables viii

List of Figures ix

1. Introduction 1

    1.1 Motivation 2
    1.2 Contributions 4
    1.3 Organization of Project 4

2. Background and Related Work 6

    2.1 Modeling Human Interactions for Navigation 6
    2.2 Trajectory Prediction 8

3. Approach 10

    3.1 Problem Formulation 10
    3.2 Occupancy Grid 11
    3.3 Attention Weights 14
    3.4 Social Pooling 16
    3.5 Position Estimation 17
    3.6 Training the model 18
    3.7 Inference for path prediction 19

4. Experiments 20

4.1 Datasets 20
4.2 Metrics and Baselines 20
4.3 Evaluation Methodology 21
4.4 Implementation Details 22
4.5 Quantitative Evaluation 23
4.6 Qualitative Analysis 25

5. Conclusion 28

5.1 Contributions and Significance 28
5.2 Future Work 29

Bibliography 31

List of Tables

Table **Page**

4.1 Prediction Errors (in metres) of all methods across all datasets. We report two error metrics Average Displacement Error and Final Displacement Error. Results from Alahi et al [1] could not be replicated for S-LSTM since our models were not pretrained on any synthetic datasets 24

List of Figures

Figure **Page**

3.1 Step by Step procedure of our Method. (i) Position and Velocity Vectors from data, (ii) Design Occupancy Grid Section 3.2, (iii) Calculation of Attention Weights Section 3.3, (iv) Social Pooling of hidden state of neighbors weighted by Attention Matrix Section 3.4, (v) Input to RNN, (vi) Position Estimation Section 3.5. During Inference, the same predicted position is used again as an input to the system. 11

3.2 Illustration of occupancy grid formation. (a)Change of shape of grid when $v_i \geq v_{thresh}; v_1 > v_2 > v_3$. (b) Change of shape of grid when $v_i \leq v_{thresh}; v_1 > v_2 > v_3$. (c) Radial and Angular Grid Formation 14

3.3 Point O is the location of Pedestrian i; Point Q is the location of neighbor j. **rd** and **rb** are the relative distance and velocities respectively. θ is the angle subtended between them. P represents the $point_{impact}$. $time_{impact}$ is time taken to cover PQ with **rv** velocity. W_d is dependent upon PO 15

3.4 LSTM model with shortcut connection. A shortcut connection from input trajectory position $\mathbf{x}_i^t$ to predicted mean μ allows us to model static pedestrians more efficiently. 18

4.1 Comparison of Trajectories generated by LSTM(i), and Skip-LSTM(ii) models. Shorter "tails" for static pedestrians predicted by the Skip-LSTM shows that it is able to handle identity mapping better than LSTM. The images are an annotation of a real scene from the ZARA-02 [16] dataset. All trajectories are drawn in the pixel coordinate space. 26

4.2 Illustration of Social Adaptive Model making successful trajectory predictions. The images are annotations of various real scenes from the ZARA-02 [16] dataset. All trajectories are drawn in the pixel coordinate space. 26

4.3 Illustration ofscenarios where the Social Adaptive Model fails. The images are annotations of various real scenes from the ZARA-02 [16] dataset. All trajectories are drawn in the pixel coordinate space. 27

Chapter 1: Introduction

The integration of self-driving vehicles and social robots into human society has accelerated at an unseen pace in the past couple years. With the advent of these autonomous systems, it becomes imperative that they begin to co-exist and operate along side human crowds. Towards this goal, predicting trajectories, and cooperative navigation has become one of the most important and challenging tasks in computer vision. This requires the autonomous systems to not only navigate through a crowd in a safe and efficient manner, but also in a socially compliant way. That is, the systems should be able to collaboratively avoid collisions with surrounding human beings and alter their path in a socially acceptable manner. To achieve this, the system should accurately be able to predict the future probable tracks of the other human beings in the scene and be able to plan its own path accordingly, in order to reach its destination.

Modeling and prediction of human trajectories not only finds applications in robotics and autonomous systems, but also in several other key areas of computer vision, such as target tracking Sadeghian et al [20], and activity forecasting Kitani et al [13]. Inferring trajectories of objects has been found to cover a wide range of domains including but not limited to sports analysis and biology. Activity forecasting stems from understanding the behavior of people in a social environment, and inferring their future actions from noisy visual inputs. Understanding the concept of human preference in a social setting, the set of a

large number of (uncommon) common sense rules and compliance with social conventions, such as respecting personal spaces, yielding right of the way etc., helps in predicting human activities and could also lead to identification of behavioral anomalies. Therefore, modeling and predicting human trajectories is a very important problem in the field of computer vision.

1.1 Motivation

When people navigate through a crowd they adapt their trajectory to their destination, while accommodating for other people in their vicinity. In order to learn how to navigate in a crowded setting, it is important that autonomous systems be as adaptive as humans and learn to change their trajectories based on their 'knowledge' of human-human interactions in a crowd. Early works by Helbing et al [8], Hall et al [5] in the domain of robot navigation attempted to model individual human motion patterns in crowds and predict their future trajectories. However, Trautman et al [22] showed that the independent modeling was unable to capture the complex and subtle interactions between humans in the crowd. This resulted in the path of the robot being highly sub-optimal, showing the importance of capturing crowd dynamics from an individual pedestrian's standpoint.

More recent approaches such as Trautman et al [22], Vemula et al [24], and Alahi et al [1] jointly model the distribution of future trajectories of all interacting people in the scene using a spatially local interaction model. Such a model is able to capture the dependencies between trajectories of interacting people in the scene and predict socially compliant trajectories. However, all of these approaches assume that the spatial neighborhood around the person is extremely local and constant in size, which we believe is not necessarily true in the real-world crowd settings. For example, a pedestrian would usually take into the

consideration the people in his vicinity when he is walking at an average human speed. But, if suddenly the pedestrian decides to run, he can no longer limit his attention solely on the people within his previous region of interest. He would have to increase his region of interest because now he might approach a person coming from the opposite direction and located outside his previous region of interest in lesser time than before, and thus would have to give himself an optimum reaction time to adjust his path and avoid collision. Vemula et al [25] proposed to solve the problem by considering all pedestrians in the scene while calculating the influence on one particular pedestrian. However, this approach introduces a considerable computation overload in dense crowd scenes. Moreover, it is futile to consider neighbors located far away from the pedestrian because they are more likely to have no influence on the pedestrian's trajectory.

The above approaches also give importance to the trajectory of a neighboring pedestrian based on his spatial displacement from the person whose trajectory we are trying to predict, without any consideration towards the neighbor's direction of motion, or their probability of collision, which is unlikely in a real world setting. For example, two pedestrians coming from opposite directions, would carefully observe each others path until they have crossed one another. Once they have crossed paths, they will no longer pay attention to each others trajectories because they are diverging and thus, their probability of collision is almost zero. These observations lead us to the insight that the importance a person gives to his neighbors should not only be dependent upon their spatial positions at a specific instant of time, but also their relative direction of motion and their probability of collision.

1.2 Contributions

In this work, we present an approach that tries to solve the problem of trajectory prediction and navigation in a dense human crowd by tackling the specific problems mentioned in Section 1.1. To this end, we use a feed-forward, fully differentiable, and jointly trained recurrent neural network (RNN) mixture model augmented with a novel pedestrian weighting scheme. Specifically, our contributions are as follows:

- **Adaptive local neighborhood:** We propose a neighborhood scheme which is neither constant in size for every kind of person in the scene, nor does it consider the entire scene to be its neighbor. Our proposed algorithm defines a person's neighborhood region solely based upon this person's behavior.

- **Attention Module:** We also introduce a novel attention module which determines the influence a neighbor should have on a pedestrian, based not only on their spatial displacement but also on their relative direction of motion. Instead of using a manually written function to determine the attention, we let our model learns the attention by observation from the datasets.

- **Skip LSTM:** We propose a minor modification to the existing LSTM models to tackle problems of identity mapping in deep neural networks.

1.3 Organization of Project

This project report is organized into six main parts. In the chapter 1, we have outlined the significance of trajectory prediction and navigation in crowds. In chapter 2, we discuss existing literature that addresses the challenge of modeling of human interactions in crowds, and trajectory prediction. Chapter 3 describes in details the nuances of our proposed

algorithm. In chapter 4, we report on the experiments we performed and compare our method with existing algorithms. In chapter 5 we draw a conclusion of the work done in our project, wherein we describe the significance of our contributions and also possible future works.

Chapter 2: Background and Related Work

The work done in this project is closely relevant to past literature in the domain of modeling human interactions for navigation and human trajectory prediction. In this chapter we describe the previous work done in these fields, particularly in modeling human interactions for Navigation, and Trajectory Prediction. We also describe how our work is different or similar to some of the works done in these two domains.

2.1 Modeling Human Interactions for Navigation

Modeling the dynamic interactions between pedestrians is the key to predicting their future behavior. In an early work by Helbing et al [8], the Social Forces model was proposed to model the motion of pedestrians in terms of the forces that drive humans to reach a goal and to avoid obstacles. The model incorporates two forces - the attractive forces which guide a pedestrian towards its destination, and repulsive forces that prevent collision between pedestrians. Subsequently, several approaches Helbing et al [7], Johansson et al [9] proposed an extension to the Social Forces model, and used learning algorithms to find parameters for the attractive and repulsive force functions that best fit the observed crowd behaviour. The Social Forces models were based on relative distance and used a hand-engineered potential term based on those distances. They captured simple interactions like repulsion and attraction efficiently, but failed to reflect the real-world crowd behavior

which is composed of more complex interactions like co-operation as shown in Alahi et al [1]. They did not learn human-human interactions from the observed data and hence, did not succeed at fully modelling crowd behavior.

Hall et al [5] introduced a theory based on human proximity relationships that has been used in potential field methods to model human-human interactions Svenstrup et al [21], Pradhan et al [19]. These models capture the interactions that motivate avoidance of collisions effectively, but do not model human-human or human-robot cooperation. However, modeling of such cooperation behavior is paramount to safe and efficient navigation in dense crowds Trautman et al [22], because in cases where the crowd density is high, the robot believes that there is no feasible path in its environment unless it accounts for cooperation from the crowd.

Trautman et al [22] proposed Interacting Gaussian Processes (IGP) to explicitly model the human-robot cooperation. IGP models the joint distribution of trajectories of all interacting agents in a scene, using Gaussian processes with a hand-crafted interaction potential term resulting in a probabilistic model that can capture joint collision avoidance behavior. This was further extended by Vemula et al [24] where the hand-crafted potential term was replaced by a locally trained interaction model based on occupancy grids. However, these models assume that the final destination of all pedestrians are known, which is not the case in a realistic prediction task.

The works of Kuderer et al [15], Kretzschmar et al [14] are closely related to IGP. These approaches explicitly model human-robot cooperation and jointly predict the trajectories of all agents, using feature-based representations. They use maximum entropy inverse reinforcement learning (IRL) to learn an interaction model from human trajectory database using carefully designed features such as clearance, velocity, or group membership. However,

their approaches were tested in synthetic environments with no more than four humans. In our work, we deal with crowded scenarios with an average of six humans in a single scene. Recently, Pfeiffer et al [18] have extended the maximum entropy approach to unseen and unstructured environments by using a receding horizon motion planning approach.

2.2 Trajectory Prediction

The problem of human trajectory prediction is a significant challenge in the field of computer vision and video surveillance. Kim et al [11], Joseph et al [10] learn motion patterns of pedestrians in videos using Gaussian processes and cluster human trajectories into these motion patterns. Although, these motion patterns can capture static obstacles in the scenes, they ignore human-human interactions using semantic scene information. More recently, Alahi et al [1] used Long Short-Term Memory networks (LSTM) to jointly reason across multiple agents to predict their trajectories in a scene. This work was extended in Varshneya et al [23], Bartoli et al [2] to include static obstacles in the model in addition to dynamic agents. However, these approaches assume that all the dynamic agents in a fixed sized local discretized neighborhood of a pedestrian affect the pedestrian's motion. Recent works in Fernando et al [4], Vemula et al [25] considers all the agents in an environment, rather than just local neighborhood, using attention. However, the attention model used in Fernando et al [4] is hard-wired based only on the proximity between a pedestrian and its neighbors rather than learning from the data. Vemula et al [25] tries to capture the relative importance of each person in the scene from the observed data of the crowd, using a Recurrent Neural Network (RNN) architecture based on spatio-temporal graphical models. However, Vemula et al [25] has a fallacy in assigning importance to neighboring agents. For example, it will always assign equal importance to a neighboring agent in a scene with

no other agents, no matter where the neighboring agent is located. A neighbor located on the fringes of the scene should not have as much importance as the same agent located just in front of the agent in consideration. Thus, this model tends to lose much of the spatial information in a sparse scene. They also use three different kinds of LSTM cells, introducing a huge number of learnable parameters. In our work, we build upon the model proposed by Alahi et al [1] and try to introduce an attention model while determining neighbor influence which works in both dense and sparse scenarios. We also try to solve the problem using a single LSTM cell thereby reducing a lot of computational overheads.

Chapter 3: Approach

When humans navigate through a crowded scene, they usually adapt their trajectories to the motion and behavior of other pedestrians in the scene.Alahi et al [1], Trautman et al [22], Vemula et al [24] assume that the influence is spatially local i.e. only spatially nearby neighbors within a constant neighborhood size influence the trajectory of the pedestrian in the crowd, which is not necessarily true. In this work, we propose a simple method for adapting the spatial neighborhood of a pedestrian based on its velocity. Secondly, we describe the attention model which learns the influence other neighboring pedestrians have on the concerned pedestrian. Next, we propose a modified RNN mixture architecture which not only predicts the trajectory of a pedestrian but also models other important features such as velocity, acceleration, and heading which play an important role in deciding other pedestrians' motion. Finally, we describe our complete model which uses the attention mechanism, along with the modified RNN network that simultaneously predicts the future location of all pedestrians in the scene and captures human-human interaction.

3.1 Problem Formulation

We assume that every scene is first preprocessed to obtain the spatial coordinates of all pedestrians at different time-steps. At any time instant t, pedestrian i is represented by his xy-coordinates (x_i^t, y_i^t) and his instantaneous velocity coordinates $(v_x, v_y)_i^t$. We observe

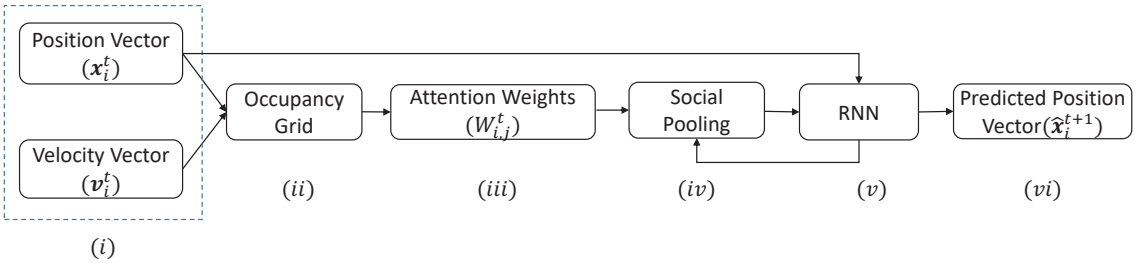

Figure 3.1: Step by Step procedure of our Method. (i) Position and Velocity Vectors from data, (ii) Design Occupancy Grid Section 3.2, (iii) Calculation of Attention Weights Section 3.3, (iv) Social Pooling of hidden state of neighbors weighted by Attention Matrix Section 3.4, (v) Input to RNN, (vi) Position Estimation Section 3.5. During Inference, the same predicted position is used again as an input to the system.

the positions of all pedestrians from time 1 to T_{obs} and predict their positions for time instants from $T_{obs}+1$ to T_{pred}. This task is similar to a sequence generation task, where the input sequence corresponds to the observed positions of a person and we are interested in generating an output sequence representing his future positions at different time-steps. Figure 3.1 illustrates the step by step procedure we follow to solve the problem.

3.2 Occupancy Grid

Human beings moving in a crowd adapt their motion based not only on their own velocity and destination, but also on the behavior of other people around them. For example, pedestrians often alter their paths when they see another person, or a group of people approaching them. Such deviations in the trajectory can not be predicted solely by observing a pedestrian in isolation without considering his neighboring crowd. Therefore we draw inspiration from the principles stated in Alahi et al [1] and develop a novel social pooling mechanism to model crowd dynamics in a more effective manner.

To capture the influence of neighbors on a pedestrian's trajectory, we construct an adaptive elliptical neighborhood region for the pedestrian, oriented in its direction of motion. However, instead of considering a constant size neighborhood, the radii of the elliptical neighborhood is dependent upon the velocity of the pedestrian for the reasons explained later in this section. We hypothesize that a pedestrian will focus primarily in its direction of motion. A fast moving pedestrian covers a fixed distance in lesser amount of time, or in other words, it travels a larger distance in the same amount of time. It is therefore imperative that the pedestrian should focus over a larger distance along its direction of motion while he is moving fast. On the other hand, a pedestrian moving slowly has the luxury to focus over a shorter distance since it has a larger reaction time. Therefore, we hypothesize that the neighborhood radius of a pedestrian along its direction of motion is proportional to its speed. Henceforth, we shall refer to this radius as the *major radius*.

Building on top of this logic, a fast moving pedestrian will pass other pedestrians located at a position that is orthogonal with respect to its direction of motion in a shorter period, and thus will pay less attention to them. Therefore, we hypothesize that the neighborhood radius along the direction which is orthogonal to its motion (*minor radius* should be inversely proportional to its speed, or in other words, inversely proportional to the major radius. This implies that

$$r_1 \times r_2 = k^2.v_{thresh}^2 \tag{3.1}$$

where r_1 is the major radius, r_2 is the minor radius, v_{thresh} is the speed which differentiates between a fast and slow pedestrian, and k is a constant multiplier term.

However, the above hypothesis implies that for a slow pedestrian, the minor radius will become larger than the major radius. This implies that a slow pedestrian would focus more in the orthogonal direction of his motion which is in contradiction with our initial

assumption that a pedestrian's primary focus is in the direction of its motion. Thus, we make a slight tweak to above equation to get the following equations:

$$\begin{aligned} r_1 &= k.v \\ r_2 &= \begin{cases} k^2.v_{thresh}^2/r_1, & \text{if } v > v_{thresh} \\ r_1, & \text{otherwise} \end{cases} \end{aligned} \tag{3.2}$$

where v is the speed of the pedestrian whose neighborhood is in consideration. Thus for a fast pedestrian, the neighborhood will be elliptical, and for a slow pedestrian, it will be circular.

After calculating the shape of the neighborhood, it is split into several cells using angular and radial grids to preserve spatial information for Social Pooling. While angular grids store the angular distance of a neighbor with respect to the pedestrian's direction of motion, the radial grids store the distance of the neighbor from the pedestrian's position. A neighbor's position is of paramount importance to a pedestrian when they are very close so that they can avoid collision and alter directions easily. Therefore we split our neighborhood into a radial grid of exponentially increasing radius as shown in Fig. 3.2. On one hand, this method provides us with a high-resolution grid space for nearby neighbors, and on the other hand, through low-resolution grid space as we move away from the pedestrian, it helps us to take into account the influence of neighbors located farther without increasing dimensionality.

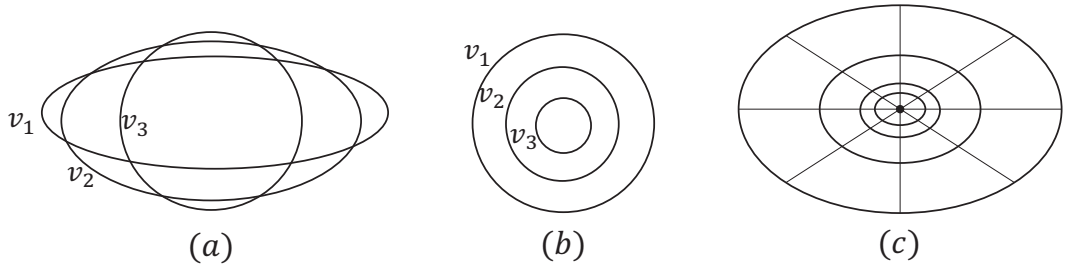

Figure 3.2: Illustration of occupancy grid formation. (a)Change of shape of grid when $v_i \geq v_{thresh}; v_1 > v_2 > v_3$. (b) Change of shape of grid when $v_i \leq v_{thresh}; v_1 > v_2 > v_3$. (c) Radial and Angular Grid Formation

3.3 Attention Weights

Let us consider two examples. For the first example, let us consider two neighbors approaching a pedestrian along the same straight line. The pedestrian, in this case, should give more priority to that neighbor with whom it expects to collide early, and then adjust its path accordingly. For the second example, we consider two neighbors coming towards the pedestrian with similar speed but along different directions. The pedestrian in this case should give more priority to the neighbor who is predicted to come closest to him and then adjust its path accordingly.

Based on the above two examples, we can hypothesize that the influence a neighbor will have on a pedestrian is a function of both "time to impact" ($time_{impact}$), and "point of impact" ($point_{impact}$). We define $point_{impact}$ as the point in the reference frame of the pedestrian with minimum distance between self and the neighbor, if both continued on their current course. We define $time_{impact}$ as the time taken by the neighbor from its current position to reach

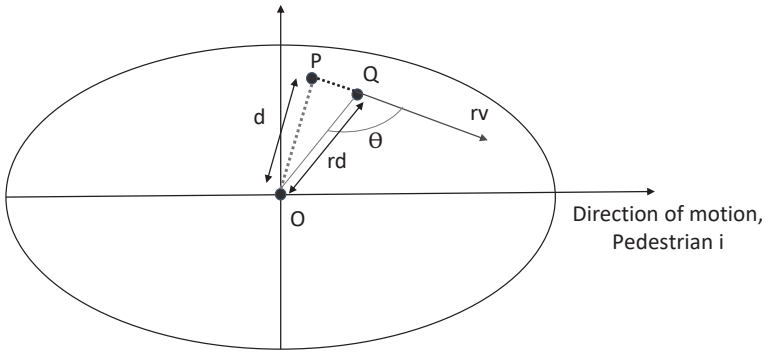

Figure 3.3: Point O is the location of Pedestrian i; Point Q is the location of neighbor j. **rd** and **rb** are the relative distance and velocities respectively. θ is the angle subtended between them. P represents the $point_{impact}$. $time_{impact}$ is time taken to cover PQ with **rv** velocity. W_d is dependent upon PO

$point_{impact}$ assuming no change in velocities of the neighbor and pedestrian.

$$\begin{aligned} time_{impact} &= \frac{\| \mathbf{rd} \|}{\| \mathbf{rv} \|}.\cos\theta \\ point_{impact} &= time_{impact}.\mathbf{rv} \end{aligned} \tag{3.3}$$

where, $\mathbf{rd} = (rd_x, rd_y)$ is the relative position vector of the pedestrian from the neighbor, $\mathbf{rv} = (rv_x, rv_y)$ is the relative speed of neighbor with respect to neighbor, θ is the angle between **rd** and **rv**, and $\| \, . \, \|$ denotes the magnitude of a vector. Solving the above equation may sometimes result in a negative value of $time_{impact}$ depending upon the value of θ. This means that the impact happened in the past. In other words, the pedestrian and neighbors are diverging and therefore, have no chance of collision. We do not calculate the effect such neighbors will have on a pedestrian's trajectory because a pedestrian is unlikely to alter his path based upon this neighbor's behavior with whom it has no chance of collision.

Thus, having obtained $time_{impact}$ and $point_{impact}$, we calculate the influence of a neighbor on a pedestrian as follows:

$$W_d = 2\pi\sqrt{\sigma_1\sigma_2}.N(p_x|0,\sigma_1^2).N(p_y|0,\sigma_2^2)$$

$$W_t = \exp^{-\alpha.time_{impact}} \tag{3.4}$$

$$W_{i,j} = W_d.W_t.\delta(time_{impact} \geq 0)$$

where, $(p_x, p_y) = point_{impact}$, $\sigma_1 = r_1/3$, $\sigma_2 = r_2/3$ (r_1, r_2 are the major and minor radii, refer Section 3.2), N is normal distribution, $\delta(.)$ is the Dirac's delta function, α is a learnable parameter which determines the slope of the decreasing function, W_d is the weight due to "point of impact", W_t is the weight due to "time of impact", and $W_{i,j}$ is the influence of the neighbor j on pedestrian i. Neighbor j must belong to the neighborhood of pedestrian i.

3.4 Social Pooling

We use LSTMs to learn an efficient hidden representation of the temporal behavior of every pedestrian in a scene. Since we need a compact representation to combine the information from neighboring states we use "Social Pooling" layers as proposed by Alahi et al [1]. The LSTM cell of a pedestrian receives the pooled hidden state information from its neighbors. While pooling the information, we try to preserve the spatial context information using the occupancy grid explained in Section 3.2.

The hidden state h_i^t of the LSTM represents the hidden state vector of pedestrian i in the scene at time t. We pool the hidden states from the neighboring pedestrians and create the Social Tensor H_i^t as follows:

$$H_i^t(m,n,:) = \sum_{j \in N_i} \mathbb{1}_{m,n}[r_{i,j}^t, \theta_{i,j}^t]\frac{W_{i,j}^t}{Z}.h_j^{t-1} \tag{3.5}$$

where,

$$Z = \sum_{j \in N_i} \mathbb{1}_{m,n}[r_{i,j}^t, \theta_{i,j}^t]W_{i,j}^t \tag{3.6}$$

where N_i is the set of neighbors corresponding to pedestrian i, h_j^{t-1} is the hidden state of the LSTM corresponding to the j^{th} neighbor at $t-1$, $r_{i,j}^t, \theta_{i,j}^t$ are the radial and angular positions respectively of neighbor j with respect to pedestrian i at time instant t, $\mathbb{1}_{m,n}[r,\theta]$ is the indicator function to determine if (r,θ) is in the (m,n) grid cell, and $W_{i,j}^t$ is the influence of the neighbor j on pedestrian i at time t.

We rasterize and embed the pooled Social Tensor into the context vector c_i^t and the positional coordinates of pedestrian i into the positional vector e_i^t. These embedding are concatenated and used as input to the LSTM cell of the corresponding trajectory at time t. Thus, we obtain the following recurrence.

$$\begin{aligned} e_i^t &= \phi(x_i^t, y_i^t; W_e) \\ c_i^t &= \phi(H_i^t; W_a) \\ h_i^t &= LSTM(h_i^{t-1}, concat(e_i^t, a_i^t); W_l) \end{aligned} \tag{3.7}$$

where $\phi(.)$ is an embedding function with a non-linearity, W_e is the positional embedding weight, and W_a is the context embedding weight. The LSTM weights are denoted by W_l.

3.5 Position Estimation

We use the hidden state of the LSTM at time t to predict the distribution of the trajectory position at time $t+1$. We assume a bi-variate Gaussian distribution parameterized by mean $\mu_i^{t+1} = (\mu_x, \mu_y)_i^{t+1}$, standard deviation $\sigma_i^{t+1} = (\sigma_x, \sigma_y)_i^{t+1}$ and correlation coefficient ρ_i^{t+1}. These parameters are obtained by passing through a fully connected linear layer W_p

$$[\mu_i^{t+1}, \sigma_i^{t+1}, \rho_i^{t+1}] = W_p h_i^t \tag{3.8}$$

He et. al [6] showed us that deep neural networks usually have the problem of approximating identity mapping, which could be solved by adopting residual learning using shortcut connections. Drawing inspiration from them, we adopt the same concept while predicting

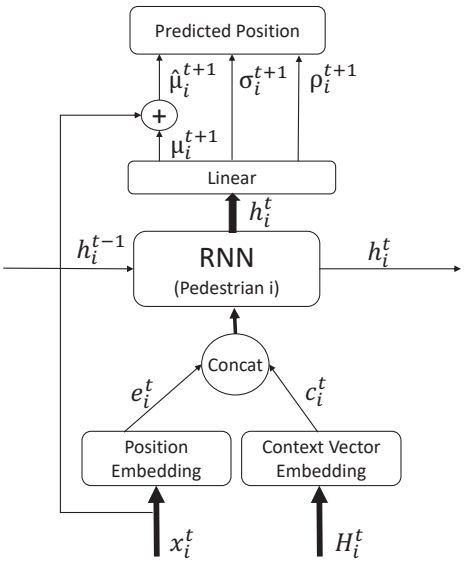

Figure 3.4: LSTM model with shortcut connection. A shortcut connection from input trajectory position $\mathbf{x}_i^t$ to predicted mean μ allows us to model static pedestrians more efficiently.

the distribution of trajectory position, by forming a skip connection from the input trajectory position to the predicted mean μ. We hypothesize that this would not only help us model static pedestrians more efficiently, but also jointly model the hidden state based on both position and velocity of the person. Thus, the updated mean is as follows:

$$\hat{\mu}_i^{t+1} = \mu_i^{t+1} + \mathbf{x}_i^t \tag{3.9}$$

where, $\mathbf{x}_i^t = (x_i^t, y_i^t)$.Figure 3.4 gives an illustration of the same.

3.6 Training the model

We jointly train the trainable variables in the model by minimizing the log-likelihood loss L_i at all predicted time-steps $t = T_{obs} + 1, \cdots, T_{pred}$ as follows:

$$L_i(W_e, W_a, W_l, W_p, \alpha) = -\sum_{t=T_{obs}+1}^{T_{pred}} \log(\mathbb{P}(x_i^t, y_i^t | \hat{\mu}_i^t, \sigma_i^t, \rho_i^t)) \tag{3.10}$$

The loss is computed over all trajectories in the training data-set and back-propagated through multiple LSTMs in a scene at every time-step.

3.7 Inference for path prediction

During test time, we initialize the model by observing the trajectory from time-steps $t = 1, \cdots, T_{obs}$ and then use it to predict the future position $(\hat{x}_i^t, \hat{y}_i^t)$ for all pedestrians by sampling from the predicted bi-variate Gaussian distribution for time-steps $t = T_{obs}+1, \cdots, T_{pred}$. However, since we want to predict the most probable path the pedestrian would take, we can simply sample the mean of the Gaussian distribution. Formally,

$$\begin{aligned} (\hat{x}_i^t, \hat{y}_i^t) &\sim N(\hat{\mu}_i^t, \sigma_i^t, \rho_i^t) \\ \text{or,} \quad (\hat{x}_i^t, \hat{y}_i^t) &= \hat{\mu}_i^t, \quad \text{for most probable path} \end{aligned} \tag{3.11}$$

From time $T_{obs}+1$ to T_{pred}, we use the predicted position $(\hat{x}_i^t, \hat{y}_i^t)$ from the previous time-step in place of the true coordinates (x_i^t, y_i^t) in Eq. 3.7. We also replace the actual coordinates with the predicted coordinates while constructing the Social Tensor H_i^t in Eq. 3.5

Chapter 4: Experiments

4.1 Datasets

We evaluate our model on two publicly available datasets: ETH [17], and UCY [16]. The ETH dataset contains two scenes each with 750 different pedestrians and is split into two sets (*ETH* and *Hotel*). The UCY dataset contains two scenes with 786 people. This dataset has three components (*ZARA-01*, *ZARA-02*, *UCY*). Thus, in total we evaluate our model on 5 crowd sets with a total of 1536 pedestrians exhibiting complex interactions such as walking together, groups crossing each other, joint collision avoidance and nonlinear trajectories, as shown in ETH [17]. The datasets were recorded at 25 frames per second, annotated every 0.4 seconds and contain 4 different scenes.

4.2 Metrics and Baselines

To compute the prediction error, we consider the following two metrics:

1. *Average Displacement Error (ADE)*: Similar to the metric used in ETH [17], this measure is the mean euclidean distance over all estimated points at each time-step between the predicted trajectory and true trajectory.

2. *Final Displacement Error (FDE)*: Introduced in Alahi et al [1], this metric computes the mean euclidean distance between the final predicted location and the final true location after T_{pred} time-steps.

While evaluating both the above metrics, we follow the inference methodology described in Section 3.7, i.e. we observe a pedestrian for time-steps $t = T_{obs} + 1, \cdots, T_{pred}$ and make predictions from time $T_{obs} + 1$ to T_{pred}. We also use the predicted coordinates as LSTM inputs instead of the actual coordinates during inference

As shown in Alahi et al [1], Social LSTM performs better than other traditional methods such as the linear model, the Social forces model [8] and Interacting Gaussian Processes [22]. Therefore, we choose Social LSTM as one of the baselines to compare the performance of our method. Thus, we compare our model against the following baselines:

1. *LSTM*: A vanilla LSTM with no pooling mechanism

2. *S-LSTM*: The method proposed by Alahi et al [1] where every person is modeled via an LSTM with the hidden states being pooled at each time-step using the Social Pooling layer.

We also do an ablation study where we compare the performance of the vanilla LSTM model against our LSTM model with shortcut connection, which we henceforth shall refer to as *Skip-LSTM*. We also refer to our full method in this section as the *Social-Adaptive-Lstm*

4.3 Evaluation Methodology

Similar to Alahi et al [1], we use a leave-one-out approach where we train and validate our approach on 4 sets, and test on the remaining set. This is repeated for all the 5 sets. For validation within each set we divide the set of trajectories into a $80-20$ split for training

and validation data. Our baselines, LSTM and S-LSTM has also been trained using the same method. We also use the same training and testing procedure for our baseline methods used for comparison.

Accurate prediction over longer horizons is important in social navigation because it results in more globally optimal behavior. In cases where the prediction is accurate for short horizon but poor for longer horizons, the resulting paths are locally optimal and could potentially lead to a non-socially compliant and reactive behavior. Since we are more interested to see if our model can learn globally optimum paths over a longer horizon, and thus model human-human interaction better, we observe a trajectory for 3.2 secs and predict their paths for the next 4.8 secs during test time. At a frame rate of 0.4 this corresponds to $T_{obs} = 8$ time-steps and $T_{pred} - T_{obs} = 12$ time-steps. This is similar to the setting used in Alahi et al [1] and is considerable since the time frame is enough to capture significant human-human interactions, and also because pedestrians remain within the video frames for only a small period of time.

4.4 Implementation Details

We use gated recurrent units (GRU) [3], which is a variant of the LSTM, as our desired choice of RNN cell in our model. The constant-multiplier term k is set at 16 and the neighborhood is divided into 12 angular and 4 radial grids. We used a fixed hidden state dimension of 128 for all LSTM models. The positional vector is embedded into a 64 dimensional vector and the context vector is embedded into a 192 dimensional vector. A batch size of 6 is used to train the network for 500 epochs using Adam optimizer [12] with an initial learning rate of 0.0005. The global norm of gradients are clipped at a value of

10 to ensure stable learning. The model was trained on a single GTX 950M GPU with a Tensorflow implementation.

4.5 Quantitative Evaluation

We compare our method on the two metrics ADE and FDE against different baselines in Table 4.1. Firstly, and most importantly, we observe that in our experiments the S-LSTM model does not outperform the vanilla LSTM model. We tried our best to reproduce the results of the paper, but to no avail. Alahi et al [1] had initially trained their model on a synthetic dataset before fine-tuning it for real datasets. However, we do not use any synthetic data to train any of our models, which could potentially lead to a worse performance.

We observe that our *skip-LSTM* model outperforms the LSTM model on all the datasets and is especially significant on the ETH dataset. The ETH dataset consists of more instances of pedestrians slowing down and almost coming to a halt, and our skip-LSTM model which was specifically designed to handle those cases, is able to predict a pedestrian's position more accurately. This shows that the shortcut connections we introduced in our RNN architecture was a useful addition, and could find application in other applications of RNNs.

Our adaptive neighborhood and attention based method, Social-Adaptive-LSTM also outperforms LSTM and S-LSTM methods in both the metrics, confirming that it is able to model human-human interaction better than the baselines. The improvement is more significant for ETH-Hotel and UCY-Zara-01 datasets which contains most pedestrians with complex non-linear paths, and people slowing down to almost a halt and other people trying to avoid colliding with them. The improvement brought forward by our model is more prominent in the results using Final Displacement Error metric. Our model predicts the final

Metric	Dataset	LSTM	S-LSTM	Skip-LSTM	Social-Adaptive
Average Displacement Error	ETH	0.66	0.68	0.59	0.52
	HOTEL	0.70	0.65	0.62	0.64
	ZARA-01	0.58	0.59	0.55	0.53
	ZARA-02	0.56	0.53	0.54	0.50
	UCY	0.76	0.77	0.75	0.73
	Average	0.65	0.64	0.61	0.58
Final Displacement Error	ETH	1.71	1.67	1.41	1.24
	HOTEL	1.45	1.43	1.28	1.35
	ZARA-01	1.36	1.44	1.31	1.22
	ZARA-02	1.28	1.30	1.30	1.07
	UCY	1.71	1.67	1.74	1.57
	Average	1.50	1.50	1.41	1.29

Table 4.1: Prediction Errors (in metres) of all methods across all datasets. We report two error metrics Average Displacement Error and Final Displacement Error. Results from Alahi et al [1] could not be replicated for S-LSTM since our models were not pretrained on any synthetic datasets

destination of the pedestrians more accurately, thereby outperforming the other models by a significant margin.

Accurate prediction over longer horizons is particularly important in social navigation as it results in more globally optimal behavior. In cases where the prediction is accurate for short horizon but poor for longer horizons, the resulting paths are locally optimal and could potentially lead to a non-socially compliant and reactive behavior. The marked improvement shown by Social Adaptive LSTM especially over the Final Displacement Error shows that our model is able to optimize paths globally and is more suited for predicting over longer horizons.

4.6 Qualitative Analysis

In Section 4.5, we showed that although our model outperforms our baseline models by a small margin in the average displacement error metric, it shows significant improvement while predicting Final Displacement error. In this section, we look at the qualitative aspects of the of our model and observe how it jointly predicts trajectory of all the pedestrians in a scene and charter a path for navigation for them.

We first take a look at how the Skip-LSTM performs against vanilla LSTM model. Figure 4.1 (i) - (ii) shows us one such example where the Skip-LSTM model performs better than the vanilla LSTM model. (ii), which has been modeled by the Skip-LSTM shows us that the static pedestrians have a very small "tail" attached to their position, when compared to (i). This indicates that the Skip-LSTM model predicts that the static pedestrians are unlikely to wander of far in the recent future. Further, it strengthens our hypothesis that the Skip-LSTM with its shortcut connection is able to handle identity mapping (predicted points is same as input points) better than a vanilla LSTM model.

Figure 4.2 shows scenarios where the Social Adaptive Model successfully models the interaction between different pedestrians in the scene and predicts an optimum path for them avoiding collision between each other. We observe that our model can successfully navigate around an oncoming pedestrian and predicts a path which is very close to the ground truth. Even when the model does not predict close to the ground truth, it still outputs some "plausible" trajectories which a pedestrian could have taken. It is also interesting to see that when groups of people come at each other from opposite directions, our model is able to successfully charter paths, which although might seem to cross each other, is optimal and avoids collision.

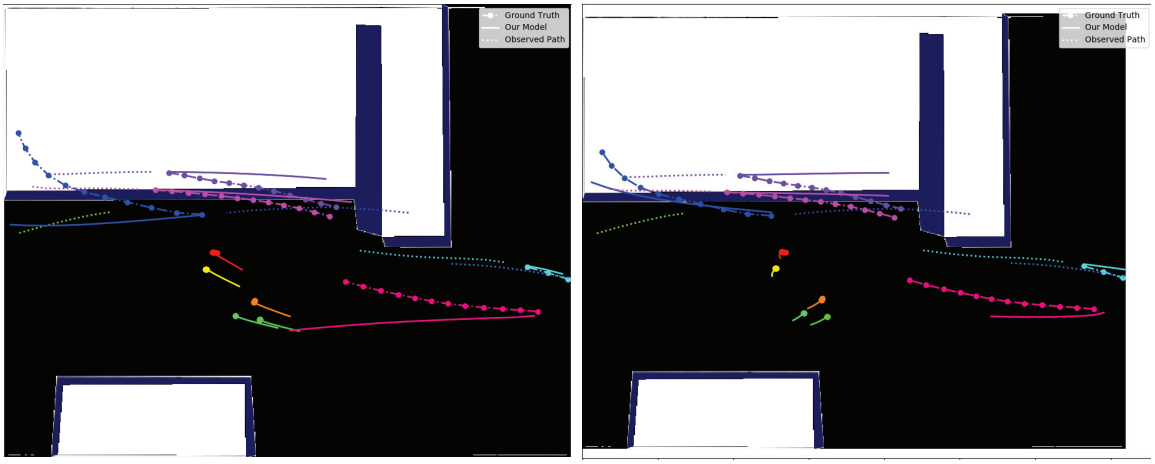

Figure 4.1: Comparison of Trajectories generated by LSTM(i), and Skip-LSTM(ii) models. Shorter "tails" for static pedestrians predicted by the Skip-LSTM shows that it is able to handle identity mapping better than LSTM. The images are an annotation of a real scene from the ZARA-02 [16] dataset. All trajectories are drawn in the pixel coordinate space.

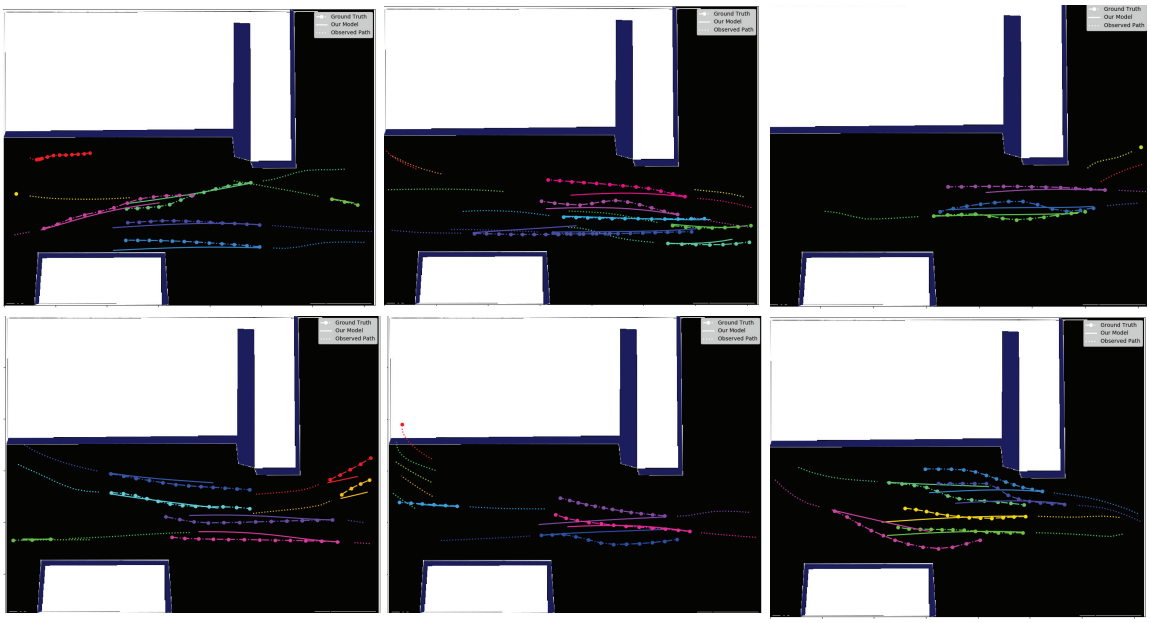

Figure 4.2: Illustration of Social Adaptive Model making successful trajectory predictions. The images are annotations of various real scenes from the ZARA-02 [16] dataset. All trajectories are drawn in the pixel coordinate space.

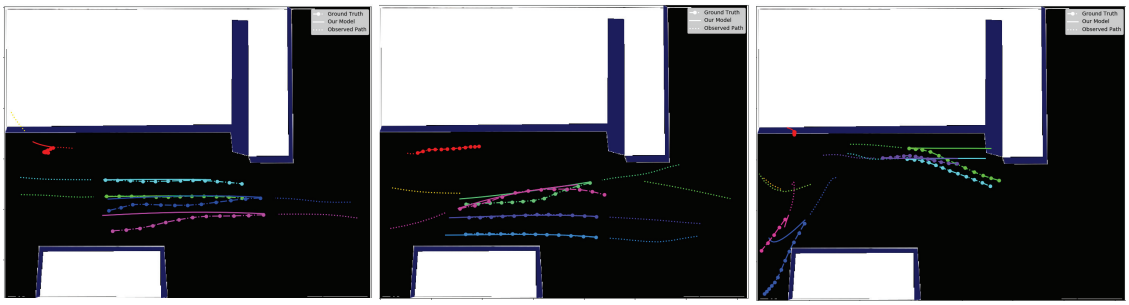

Figure 4.3: Illustration ofscenarios where the Social Adaptive Model fails. The images are annotations of various real scenes from the ZARA-02 [16] dataset. All trajectories are drawn in the pixel coordinate space.

Figure 4.3 (i) - (iii) illustrates certain scenarios where the Social Adaptive Model fails, and its predictions may result in a collision. (i) and (ii) predict two paths which overlap in the future at the same time-step indicating that these two pedestrians could collide. (iii) indicates a scenario where the model, being unaware of the static obstacles in the scene, charters a path straight into one of them.

Chapter 5: Conclusion

In this work, we discussed the challenges of modeling human-human interaction and predicting socially-compliant trajectories. With the unprecedented growth of autonomous vehicles and social robots in the past couple of years, it is now more important than ever before to understand the set of (uncommon) common sense rules and societal conventions that guide a person to navigate through a dense human crowd.

5.1 Contributions and Significance

We examined a few of the shortcomings from the works of other researchers and propose a feed-forward, fully-differentiable, and jointly trained recurrent neural network mixture model with a novel pedestrian weighting scheme, which is able to learn human-human interactions purely from the data. In summary, we proposed the following.

Firstly, we propose an **adaptive local neighborhood** scheme for social pooling. The local neighborhood is capable of adapting its size based upon a pedestrian's behavior. This allows us to not only factor in a majority of "influential" people in the scene, but also to eradicate computational overloads from considering every person in a very crowded scene.

We then introduce a novel **attention module** in our model which determines the influence a neighbor should have on predicting a pedestrian's trajectory. Our attention module is different from other works in the sense that it not only considers the spatial relation between

a pedestrian and its neighbors, but also their relative probability of collision. Instead of using a manually written function, we let our attention module learn its parameters from training on the data.

Lastly we introduce the **Skip LSTM** by making a minor modification to existing LSTM models by adding a shortcut connection from the input to the output before passing it to the network optimizer. We show that such a modification enables us to model static pedestrians more accurately than any other existing LSTM models.

We tested the efficacy of our algorithm against our baseline models on two publicly available data-sets and show that it is able to outperform them and is moderately better than the state-of-the-art. We demonstrated that our model was very efficient in performing long term predictions with a final displacement error lower than the other models. We also showed that our minor modification to the existing LSTM models, by adding a shortcut connection from the input to the output before passing it to the network optimizer, reaped benefits as it was able to model static pedestrians more accurately than other models. We also showed that in cases where our model was not able to accurately predict a pedestrian's ground truth path, it could still model a plausible, optimal, and collision-free path for the pedestrian.

5.2 Future Work

Modeling complex human interactions in dynamic environments, such as human crowds, still remains a challenging and a unsolved problem. While our proposed model in Chapter 3 takes a small step towards modeling cooperative behavior exhibited in human crowds, there are plenty of rooms for improvement. As an interesting future work, the model could be validated and verified on real robot in a dense human crowd. We could also modify our

current grid by placing the pedestrian at one of the focus of the ellipse, thus allocating more attention to his direction of motion and less to his behind. Another interesting upgrade to the grid would be to split the elliptical grid into angular grids of equal area instead of making them equi-angular as in the current approach. This would be a further improvement to the approximation of attention regions as it works in reality.

Our model considers only pedestrians within its "neighborhood", although the neighborhood is adaptable. An interesting future work could be learning to model a pedestrian's neighborhood purely from data. Another important drawback of our system is that it does not consider static obstacles in the scene while planning trajectories for pedestrians. Static obstacles play a very major role in path planning and is especially important if we introduce an autonomous system in a completely new surrounding. An interesting direction could be the integration of static object modeling during path planning.

Currently our model only considers scenes predominantly with pedestrians moving at an average speed. A challenging, but interesting future direction would be to explore ways to account for different classes of objects such as people on bicycles, or skateboards, populating a scene. Another interesting direction to tackling the problem of modeling complex human interactions could the use of Generative Adversarial Networks (GAN). The potential of GAN is huge, and it is believed that it can mimic any distribution of data, i.e. they could be used to create worlds very familiar to our own. It would be interesting to explore if a GAN can "mimic" human-human interactions and predict trajectories accurately and in a socially compliant manner.

Bibliography

[1] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 961–971, June 2016.

[2] Federico Bartoli, Giuseppe Lisanti, Lamberto Ballan, and Alberto Del Bimbo. Context-aware trajectory prediction. *CoRR*, abs/1705.02503, 2017.

[3] Kyunghyun Cho, Bart van Merriënboer, Çağlar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics.

[4] Tharindu Fernando, Simon Denman, Sridha Sridharan, and Clinton Fookes. Soft + hardwired attention: An lstm framework for human trajectory prediction and abnormal event detection. *CoRR*, abs/1702.05552, 2017.

[5] Edward T. Hall. A system for the notation of proxemic behavior. *American Anthropologist*, 65(5):1003–1026, oct 1963.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[7] Dirk Helbing and Anders Johansson. *Pedestrian, Crowd and Evacuation Dynamics*, pages 6476–6495. Springer New York, New York, NY, 2009.

[8] Dirk Helbing and Péter Molnár. Social force model for pedestrian dynamics. *Phys. Rev. E*, 51:4282–4286, May 1995.

[9] Anders Johansson, Dirk Helbing, and Pradyumn K. Shukla. Specification of the social force pedestrian model by evolutionary adjustment to video tracking data. *Advances in Complex Systems (ACS)*, 10(supp0):271–288, 2007.

[10] Joshua Mason Joseph, Finale Doshi-Velez, Albert S. Huang, and Nicholas Roy. A bayesian nonparametric approach to modeling motion patterns. *Auton. Robots*, 31(4):383–400, 2011.

[11] Kihwan Kim, Dongryeol Lee, and Irfan A. Essa. Gaussian process regression flow for analysis of motion trajectories. In *ICCV*, pages 1164–1171. IEEE Computer Society, 2011.

[12] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[13] Kris M. Kitani, Brian D. Ziebart, James Andrew Bagnell, and Martial Hebert. Activity forecasting. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part IV*, ECCV'12, pages 201–214, Berlin, Heidelberg, 2012. Springer-Verlag.

[14] Henrik Kretzschmar, Markus Spies, Christoph Sprunk, and Wolfram Burgard. Socially compliant mobile robot navigation via inverse reinforcement learning. *The International Journal of Robotics Research*, 35(11):1289–1307, 2016.

[15] Markus Kuderer, Henrik Kretzschmar, Christoph Sprunk, and Wolfram Burgard. Feature-based prediction of trajectories for socially compliant navigation. In *In Proceedings of Robotics: Science and Systems (RSS)*, 2012.

[16] Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. Crowds by example. *Comput. Graph. Forum*, 26:655–664, 2007.

[17] S. Pellegrini, A. Ess, K. Schindler, and L. van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *2009 IEEE 12th International Conference on Computer Vision*, pages 261–268, Sept 2009.

[18] Mark Pfeiffer, Ulrich Schwesinger, Hannes Sommer, Enric Galceran, and Roland Siegwart. Predicting actions to act predictably: Cooperative partial motion planning with maximum entropy models. In *IROS*, pages 2096–2101. IEEE, 2016.

[19] N. Pradhan, T. Burg, and S. Birchfield. Robot crowd navigation using predictive position fields in the potential function framework. In *Proceedings of the 2011 American Control Conference*, pages 4628–4633, June 2011.

[20] Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Tracking the untrackable: Learning to track multiple cues with long-term dependencies. *CoRR*, abs/1701.01909, 2017.

[21] M. Svenstrup, T. Bak, and H. J. Andersen. Trajectory planning for robots in dynamic human environments. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4293–4298, Oct 2010.

[22] Peter Trautman and Andreas Krause. Unfreezing the robot: Navigation in dense, interacting crowds. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010.

[23] Daksh Varshneya and G. Srinivasaraghavan. Human trajectory prediction using spatially aware deep attention models. *CoRR*, abs/1705.09436, 2017.

[24] A. Vemula, K. Muelling, and J. Oh. Modeling cooperative navigation in dense human crowds. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1685–1692, May 2017.

[25] Anirudh Vemula, Katharina Mülling, and Jean Oh. Social attention: Modeling attention in human crowds. *CoRR*, abs/1710.04689, 2017.