TECHNICAL UNIVERSITY OF MUNICH

DEPARTMENT OF INFORMATICS

Bachelor's Thesis in Engineering Science

Modeling Social Interactions for Pedestrian Trajectory Prediction on Real and Synthetic Datasets

Philipp Mondorf

TECHNICAL UNIVERSITY OF MUNICH

DEPARTMENT OF INFORMATICS

Bachelor's Thesis in Engineering Science

Modeling Social Interactions for Pedestrian Trajectory Prediction on Real and Synthetic Datasets

Modellierung von Agent-Agent Interaktionen für die Vorhersage von Fußgängertrajektorien auf realen und synthetischen Daten

Author:	Philipp Mondorf
Supervisor:	Prof. Dr. Laura Leal-Taixé
Advisor:	M.Sc. Patrick Dendorfer
Submission Date:	May 1st 2020

I hereby confirm that this bachelor's thesis is my own work and I have documented all sources and material used.

Date	Philipp Mondorf

"Prediction [...] is the Foundation of Intelligence."

– JEFF HAWKINS

Abstract

Humans moving in crowded spaces, such as sidewalks or shopping malls, socially interact with people in their vicinity. For example, they adapt their pace or alter their initial path to avoid collisions or to respect the personal space of others. These interactions are influenced by a variety of sociocultural factors and personal preferences. Due to these influences, social interactions between pedestrians are highly complex. However the ability to model these interactions is crucial to reliably predict human motion behavior in crowded scenes. With the development of autonomous moving platforms that need to share their physical environment with humans, this task has become increasingly valuable. Therefore, several trajectory prediction models have been developed that exploit social interactions between pedestrians in order to effectively predict the future movements of individuals. These models are usually evaluated and compared on datasets of real-world human motion. A critical drawback of such datasets is that the number of social interactions in real-world scenarios is inherently limited. Furthermore, these datasets often include interactions between pedestrians and obstacles in their physical environment, which additionally influences the motion behavior of individuals.

In this thesis, we present a way to conveniently evaluate the ability of a model to predict social interactions between pedestrians. In this manner, we conduct experiments on two prediction models: the Social LSTM model and the Vanilla LSTM model. We overcome the limitations of real datasets by generating synthetic datasets for which we can define the impact of social interactions on the motion of pedestrians. These hand-tailored datasets exclude interactions between pedestrians and obstacles and focus on the interactions between individuals. By comparing the performances of the models on these datasets, we show that while the Social LSTM model is capable of predicting social interactions, the Vanilla LSTM model is not. For our analysis, we introduce evaluation metrics that focus on the interactions between pedestrians. These metrics go beyond the commonly used average and final displacement error for trajectory prediction. In particular, we analyze the prediction errors in regions of trajectories that are highly influenced by social interactions. Furthermore, we analyze the collision behavior of the models' predictions and classify trajectories with respect to their degree of nonlinearity. This allows us to to compare the models' performances on motions that are differently influenced by social interactions.

Contents

1. Introduction

We humans move by walking upright - a trait known as bipedalism. Although this ability is not unique to humans, we are one of the species that has exploited its potential to its extreme [16]. While the evolution of human bipedalism, i.e. how and why it evolved, is controversial, its advantages are not [37]. For example, walking upright raises the head and allows a greater field of vision. In this manner, it enables humans to effectively navigate in complex terrain. In the last centuries, our environment has changed drastically. Many people live in big cities with well-defined walkways. Navigating in complex terrain has become associated with moving through crowded scenes while negotiating complex social interactions with multiple strangers.

Pedestrian trajectory prediction. In recent years, the development of autonomous moving platforms such as self-driving cars or social robots has become more and more important. As these technologies will share the same space as humans, predicting pedestrian trajectories will become safety-critical. Making such predictions is challenging due to the strong influence of social interactions on human motion. According to Gupta *et al.* [15], the resulting behavior of pedestrians in crowded scenes can be characterized by three main properties:

- **Interpersonality**. Humans use their innate ability to read the behavior of others when navigating through crowded scenes. They constantly predict the paths of neighboring pedestrians and adapt their motion accordingly. This causes pedestrian trajectories to become highly nonlinear.
- **Social Acceptance**. Due to cultural aspects or traffic rules, some trajectories are favored over others. This expresses in social norms like yielding right-of-way or respecting personal space.
- **Multimodality**. Given a specific scene, there are multiple plausible paths a pedestrian might take, i.e. there is no unique solution to the prediction task.

Modeling these properties in order to predict pedestrian trajectories reliably is the key task of pedestrian trajectory prediction. Recent research in this field has successfully tackled some of the above challenges. While traditional techniques have primarily addressed the aspects of interpersonality and social acceptance by using hand-crafted functions to model social interactions between pedestrians [3, 5, 19], recent achievements in deep learning have motivated researchers to revisit the problem in a new data-driven fashion [1, 15, 26, 41]. These methods have enabled great progress in addressing all of the above challenges, but suffer from one critical limitation: they strictly depend on the nature of the data provided. This dependence becomes a problem when evaluating data-driven models with respect

to a property that is not sufficiently represented in this data. In this thesis, we investigate whether existing trajectory prediction models are capable to model social interactions between individuals. These interactions significantly influence the motions of pedestrians moving in crowded scenes. It is therefore crucial to understand them to reliably predict the future movements of pedestrians. However, as the degree of social interactions in commonly used datasets of real-world human motion is limited, we can only evaluate existing trajectory prediction models on a small subset of possible motion behaviors. In order to address this problem, we adapt the Social Force model [19] to generate synthetic datasets for which we can freely define the impact of social interactions on the motion of pedestrians. We then train data-driven models on the datasets generated and infer their ability to model social interactions from the difference of their performance between each dataset. In addition, we define methods to analyze the predictions of these models. In particular, we evaluate the collision behavior of the models' predictions and define a metric to measure the error in specific nonlinear regions. We classify trajectories with respect to their degree of nonlinearity and evaluate the performances on these classes. These metrics go beyond the classical L2 loss and allow us to conclude whether a model is able to learn social interactions or not. In particular, these steps and tools are not limited to the models used in the thesis. They can be applied to evaluate any existing model for trajectory prediction.

We begin this thesis by providing a broad overview of the history of trajectory prediction. For this, we introduce important trajectory prediction models in chapter 2. We continue by explaining the main concepts of the Social Force model and then introduce the two trajectory prediction models on which we conduct our experiments, i.e. the Vanilla LSTM model and the Social LSTM model. In chapter 4, we show the limitations of real datasets and introduce methods to generate hand-tailored datasets that focus on the interactions between pedestrians. We present and discuss the results of our experiments in chapter 5, using quantitative evaluation metrics that go beyond the classical average and final displacement error. Finally, we discuss the key findings of this thesis and give an outlook on what could be done by future studies.

2. Related Work

Forecasting the motion behavior of pedestrians is a field of research with a rich history and remarkable scientific contributions from all over the world. With the development of autonomous moving platforms that need to share their space with humans, this field of research has become increasingly popular. Various models for trajectory prediction have been developed that influence the work of ongoing research. While traditional methods mainly use hand-crafted functions to model the motion behavior of individuals, data-driven techniques have received increasing interest in recent years. With the latest successes in deep learning, these models were able to overcome numerous unresolved challenges and outperformed traditional techniques. This development allows us to group trajectory prediction models into two categories: traditional knowledge-driven models that use hand-crafted functions such as energy potentials to model the behavior of pedestrians, and data-driven models that aim to learn the motion behavior of pedestrians by processing a sufficient amount of data.

Knowledge-driven models. Human motion behavior can be studied from a crowd perspective or from an individual perspective. While the former *macroscopic* approach interprets a crowd of pedestrians as one continuous entity and focuses on collective movements [17, 18, 20], the latter *microscopic* perspective regards this crowd as a set of individuals. In this thesis, we focus on the latter. One prominent example for this approach is the Social Force model by Dirk Helbing and Peter Molnár [19]. In their work, Helbing and Molnár model the internal motivation of an individual to move with attractive and repulsive forces. The influence of this work has been huge and it has often been revisited, for example in [28, 33, 39, 40, 51], to name only a few. Other microscopic models use different approaches to model the motions of pedestrians. For example, Bedau *et al.* [5] use cellular automata with a small rule set in order to model the collective motion behavior of pedestrians in bi-directional flow. Antonini *et al.* [3] use a discrete choice framework to model the short-term motion behavior of individuals with a choice set based on a combination of speed regimes and directions. Further knowledge-driven models aim to describe the motion of human crowds by using continuum dynamics. Respectively, Hughes introduces a model in [22, 23] in which he defines an evolving potential function that guides pedestrians, represented by a density field, towards their destinations. Treuille *et al.* [46] revisit Hughes' work by transforming the continuous crowd field into a particle representation, in order to create an efficient simulation for pedestrian dynamics.

While a variety of different approaches exists to describe the motion behavior of pedestrians with knowledge-driven models, they all resemble each other in one limitation. These models are based on a finite set of hand-crafted rules and functions that represent the current knowledge about human motion behavior. As state-of-the-art research in trajectory prediction

shows [31, 32], we are yet far away from entirely understanding this complex behavior, including the personal motives and preferences of pedestrians and their interactions with each other. Therefore, data-driven methods have been used over the past few years to address the challenges of trajectory prediction. These models aim to learn human motion behavior by observing a sufficient amount of data and extracting key features from this data.

Data-driven models. Pedestrian trajectory prediction is highly sequential. It comprises of iteratively predicting the next movement of a pedestrian based on previously made observations and predictions. With the advent of neural networks, the sequential nature of this task has motivated researchers to use Recurrent Neural Networks (RNNs) and especially their variants, i.e. Long Short-Term Memory (LSTM) [21] and Gated Recurrent Units (GRUs) [9], for the prediction. These networks are a rich class of dynamic models that have proven to be very successful for various sequence generation problems like speech recognition [8, 14, 44], machine translation [4, 45] or image/video captioning [11, 48, 49]. However, while RNNs are successful in learning and generalizing the properties of isolated sequences [13], they lack in jointly predicting multiple correlated sequences. Furthermore, they are deterministic models that predict only one single future trajectory for each pedestrian in a scene. Yet, human motion behavior is multimodal. Given a specific scenario, there are multiple feasible paths a pedestrian might take. Deterministic models neglect the multimodal character of human motion and tend to learn the average behavior of pedestrians. Contrarily, stochastic models aim to learn a distribution of future movements instead of a single trajectory. One example are Variational Autoencoders (VAEs) [25]. These generative models are trained by maximizing the lower bound of a data likelihood. In this manner, Lee *et al.* [29] introduce a stochastic model based on a recurrent conditional VAE that learns to generate multiple feasible samples of a trajectory and subsequently refines them. However, training a VAE can become difficult due to intractable probabilistic computations. Therefore, Goodfellow *et al.* [12] propose an alternative approach: Generative Adversarial Networks (GANs), where the training procedure is replaced by a minimax game between a generative model and a discriminative model. Gupta *et al.* [15] use this concept and propose a GAN-based model that predicts plausible trajectories by training adversarially against a recurrent discriminator. Other stochastic models [2, 26] use variants of GANs, i.e. BicycleGAN [53] and InfoGAN [7], that prevent the models from mode collapsing and, therefore, ensure diverse results.

Social interaction models. Pedestrians moving in crowded scenes socially interact with people in their vicinity. They adapt their movements according to the behavior of others. In order to predict human motion reliably, these interactions have to be taken into account. In this manner, various models and techniques have been developed that exploit social interactions between pedestrians. Alahi *et al.* [1] propose a social pooling layer that allows their LSTM-based model to jointly predict trajectories of neighboring pedestrians. This pooling layer connects neighbors by aggregating the hidden states of their respective LSTM networks at every time step of the prediction process. Gupta *et al.* [15] use a Multilayer Perceptron (MLP), followed by a max pooling to aggregate information across neighboring pedestrians in the generation step. Furthermore, they show that this technique performs comparatively well

and is computationally more efficient than the proposed social pooling layer in [1]. Inspired by the latest successes with attention-based models [35, 50], Sadeghian *et al.* [41] present a social attention mechanism that aggregates information across different interactions between pedestrians and learns to extract information about the most influencing neighbors. Amirian *et al.* [2] use a similar attention-based pooling scheme, but additionally defines hand-crafted interaction features that serve as a prior for the pooling process. Kosaraju *et al.* [26] use a Graph Attention Network (GAT) [47] to learn the feature representations that encode the interactions between pedestrians.

Overall, we see that there are numerous concepts and models that address the problem of trajectory prediction. While some of these models focus on the social interactions between individuals [1, 15], others look exclusively at the interactions between pedestrians and obstacles in their spatial environment [36, 43]. Moreover, there are many models that consider both impacts [29, 41, 53]. In this thesis, we focus on the interactions between pedestrians and, therefore, neglect spatial interactions between pedestrians and obstacles. Furthermore, we concentrate on two trajectory prediction models: the Vanilla LSTM model and the Social LSTM model. However, the presented concepts and tools in this work can be applied to any of the trajectory prediction models mentioned above.

One critical limitation of data-driven models is that they strictly depend on the data that is provided. This can become a problem when we want to evaluate models with respect to a property that is not sufficiently represented in this data. This problem is barely addressed in the works mentioned above. Only [2] generates a hand-tailored dataset to study the problem of mode collapsing. In this thesis, we show that the amount of social interactions between pedestrians is limited in commonly-used datasets. Hence, we design synthetic datasets that are specifically oriented to the evaluation of learning social interactions.

3. Objectives of the thesis

We begin this chapter by stating the main objectives of this thesis. This is followed by a section about social interactions, their influence on pedestrian trajectories and how we plan to identify them. We then introduce important methods and concepts that are later used in this thesis. In particular, we explain the mathematical concepts of the Social Force model and present two additional models for pedestrian trajectory prediction on which we conduct our experiments, the Vanilla LSTM model and the Social LSTM model.

3.1. Main objectives

Pedestrians moving in crowded scenes socially interact with people in their vicinity and adapt their motions accordingly. Modeling these interactions between pedestrians is crucial in order to accurately predict human motion behavior. Over the years, various models have been developed that exploit these interactions in order to reliably predict the motion behavior of pedestrians.

The main objective of this thesis is to deduce whether and to what extent existing models for trajectory prediction are able to predict these social interactions. For this, we conduct experiments on two models: the Social LSTM model and the Vanilla LSTM model. While the Vanilla LSTM model predicts each trajectory independently of the motions of others, the Social LSTM model introduces a social pooling layer that aggregates information across neighboring pedestrians. Due to these properties, we expect the Vanilla LSTM model to be not capable of predicting the interactions between individuals. In order to conclude the extent to which the social pooling layer allows to predict these interactions, we compare the performances of both models on datasets with variable amounts of social interactions present. In datasets of real-world human motion, the level of these interactions is inherently limited. We overcome this problem by presenting a method to generate synthetic datasets for which we can freely define this level. Furthermore, we propose evaluation metrics that focus on the interactions between pedestrians. These metrics go beyond the commonly used average and final displacement error for pedestrian trajectory prediction and ensure a rich and detailed analysis.

3.2. Social interactions

Before we begin to discuss whether a trajectory prediction model is capable of predicting social interactions or not, we first need to define what social interactions are and how we plan to identify them. As mentioned in chapter 1, human motion is interpersonal. Pedestrians plan

their paths by considering the behavior of others while keeping their goals in mind. When they navigate through crowded scenes, they interact with neighboring pedestrians and adapt their motions accordingly. For example, they change their pace or alter their path in order to avoid collisions. These interactions are influenced by a variety of sociocultural factors and personal preferences that affect, for example, on which side pedestrians favorably pass each other or at which distance they begin to invade the personal space of neighbors [10]. Due to these influences, social interactions between pedestrians are highly complex. However, as they significantly affect the way humans move in crowded scenes, modeling these interactions is crucial in order to predict the respective motions accurately. In the following, we define social interactions more systematically by considering the specific movements of pedestrians and their respective trajectories.

In this manner, let us for now assume a scenario with only one pedestrian l present, i.e. there are no other persons or obstacles in the scene. At time t this pedestrian has a position $\vec{x}_l^t$ and a destination $\vec{o}_l$. We further assume that this person wants to reach its destination $\vec{o}_l$ by taking the fastest route possible. If we denote the distance vector between the current position $\vec{x}_l^t$ and the destination $\vec{o}_l$ as $\vec{e}_l^t$, this vector will give us the direction of the fastest way. In an empty scenario like this, the observed movements of pedestrian l would strictly follow this direction. Therefore, we would obtain an undisturbed, linear trajectory X_l.

Definition. Let us now assume that instead of one pedestrian, there are multiple pedestrians in the scene. At time t, each pedestrian i has a position $\vec{x}_i^t$ and a destination $\vec{o}_i$ with $i \in 1, \ldots, N$, for N pedestrians present. While walking, it is likely that some pedestrians need to pass each other. In order to avoid collisions or to comply with social norms, they need to adapt their paths according to the motion of neighbors. We define social interactions, or human-human interactions, as an adaption of a pedestrian's path caused by the behavior of others. These interactions are, dependant on the number of pedestrians involved, bi- or multi-directional. It is therefore possible that the motion of a pedestrian is simultaneously influenced by multiple neighbors.

Trajectory prediction. In this thesis, we want to deduce whether existing trajectory prediction models are capable to model these interactions in order to accurately predict human motion behavior in crowded scenes. These models aim to predict the future trajectories of pedestrians, given a history of previous motions. In particular, they receive as input for each pedestrian in the scene a sequence of xy-coordinates that represents the pedestrian's past positions. Based on these observations, they predict subsequent sequences of xy-coordinates that describe the pedestrians' future trajectories. If we assume N pedestrians in a scene, the model input can be described as $X = X_1, X_2, \ldots, X_N$, where X_i denotes the input sequence of xy-coordinates for pedestrian i. In particular, this sequence is defined as $X_i = (\vec{x}_i^1, \vec{x}_i^2, \ldots, \vec{x}_i^{T_{obs}})$, where $\vec{x}_i^t = (x_i^t, y_i^t)$ denotes the position of pedestrian i at time t, for $t = 1, \ldots, T_{obs}$. Similarly, we denote the future trajectories of all pedestrians (ground truth) as $Y = Y_1, Y_2, \ldots, Y_N$, where the output sequence of pedestrian i is defined as $Y_i = (\vec{y}_i^{T_{obs}+1}, \vec{y}_i^{T_{obs}+2}, \ldots, \vec{y}_i^{T_{pred}})$, with $\vec{y}_i^t = (x_i^t, y_i^t)$ for $t = T_{obs} + 1, \ldots, T_{pred}$. The predictions of a model are denoted as $\hat{Y} = \hat{Y}_1, \hat{Y}_2, \ldots, \hat{Y}_N$, where the predicted sequence of pedestrian i is defined as $\hat{Y}_i = (\hat{\vec{y}}_i^{T_{obs}+1}, \hat{\vec{y}}_i^{T_{obs}+2}, \ldots, \hat{\vec{y}}_i^{T_{pred}})$ respectively.

Influence of social interactions on pedestrian trajectories. Any interaction between pedestrians causes their trajectories to deviate from the direct connection $\vec{e}_i^t$ between the actual position $\vec{x}_i^t$ and the destination $\vec{o}_i$. The corresponding path is disturbed and becomes nonlinear. Trajectory prediction models, have to consider these deviations in order to reliably predict the future trajectories of pedestrians. Therefore, they need to be capable of modeling social interactions.

Humans can react differently to people in their vicinity. For example, when two strangers come unintentionally close, they strongly interact and momentarily change their motion. On the other hand, they only slightly interact with people that are still far away or not completely in their field of view. We observe that the stronger pedestrians socially interact with each other, the higher is the influence on their initial paths. This reflects in an increasing degree of nonlinearity of the respective trajectories. In order to make well-founded statements about the ability of a model to predict social interactions, we need to identify where social interactions occur and focus on these regions. The commonly used metrics for the performance evaluation of pedestrian trajectory models, i.e. the Average Displacement Error (ADE) and Final Displacement Error (FDE), are not sufficient for this task. The ADE evaluates the average euclidean distance between prediction and ground truth over all time steps of the prediction process. It therefore neglects the specific shape of a predicted trajectory and does not weight the performance of a model in important regions, i.e. where social interactions take place. This can produce misleading results as linear predictions that entirely neglect social interactions may fit a given trajectory better than curved predictions that try to model these interactions when it comes to an average over all positions. The FDE only analyzes the error of the final predicted position. As social interactions take place within a certain time span, a single position is not sufficient to represent these interactions. Therefore, the FDE is only of limited significance for the stated objective. A convenient way to evaluate the prediction of social interactions is to focus on nonlinearities. A trajectory that is not influenced by any interactions is linear. As these interactions increase, the trajectory becomes increasingly nonlinear. We evaluate the predictions of the models by analyzing the ADE in specific nonlinear regions of the trajectory. Furthermore, we classify the trajectories of a dataset into linear, gradually and highly nonlinear trajectories. With this, we obtain groups of trajectories that are not at all, gradually or significantly influenced by social interactions. We then evaluate and compare the respective performances of the models on these groups. Finally, we analyze the predictions' collision behavior. Models that are capable of learning social interactions should predict collision behavior that is close to that of the given dataset.

3.3. Methods

In this thesis, we tackle the mentioned limitations of datasets of real-world human motion by generating synthetic datasets for which we can control the impact of social interactions on the motion of pedestrians. For this, we use the Social Force model [19]. This knowledge-driven model uses attractive and repulsive forces to describe the different influences on human motion. By comparing the performances of existing trajectory prediction models on the

datasets generated, we evaluate their ability to predict social interactions. In this manner, we conduct experiments on two models: the Vanilla LSTM model and the Social LSTM model [1]. In the following, we explain the concepts of these models.

3.3.1. The Social Force model

In 1995, Dirk Helbing and Petér Monlár proposed a pedestrian motion model that uses forces to model the behavior of individuals. According to Helbing and Molnár, these *social forces* are not directly exerted by the pedestrians' environment. Instead, they can be interpreted as an internal motivation of individuals to perform certain movements. Simulations show that with this concept, it is possible to realistically describe collective effects of pedestrian motion behavior in crowded scenes [19]. We use this model to generate hand-tailored datasets that focus on the interactions between pedestrians.

In the proposed work, the resultant force describing the total motivation of a pedestrian α to perform a certain movement is composed of the following attractive and repulsive forces:

- **Acceleration term towards destination**. If a pedestrian is not disturbed, he/she will walk into the *direction of his/her destination* $\vec{e}_\alpha(t)$ with a *desired speed* v_α^0. Any deviation from this velocity leads to a subsequent acceleration in order to approach v_α^0 again within a certain *relaxation time* τ_α. If we denote $\vec{v}_\alpha(t)$ as the actual velocity at time t, this constitutes in the following acceleration term:

$$\vec{F}_\alpha^0(\vec{v}_\alpha(t), v_\alpha^0\ \vec{e}_\alpha) := \frac{1}{\tau_\alpha}\ (v_\alpha^0\ \vec{e}_\alpha - \vec{v}_\alpha(t)) \tag{3.1}$$

- **Repulsive forces between pedestrians**. Due to social interactions between individuals, the motion of a pedestrian is influenced by others. A pedestrian normally feels increasingly uncomfortable the closer a stranger gets. This results in *repulsive effects*. If we denote the distance between two pedestrians α and β as $\vec{r}_{\alpha\beta}$, these effects can be modeled by a monotonically decreasing *repulsive potential* $V_{\alpha\beta}(\|\vec{r}_{\alpha\beta}\|)$[1]:

$$\vec{f}_{\alpha\beta}(\vec{r}_{\alpha\beta}) = -\nabla_{\vec{r}_{\alpha\beta}} V_{\alpha\beta}(\|\vec{r}_{\alpha\beta}\|) \tag{3.2}$$

- **Repulsive forces between pedestrians and obstacles**. A pedestrian also keeps a certain distance from buildings or other obstacles. The closer the pedestrians walks to an obstacle, the more likely it becomes that he/she is getting injured. Therefore, an obstacle B evokes *repulsive effects*, too. If we denote the respective *repulsive potential* as $U_{\alpha B}(\|\vec{r}_{\alpha B}\|)$ we get:

$$\vec{F}_{\alpha B}(\vec{r}_{\alpha B}) = -\nabla_{\vec{r}_{\alpha B}} U_{\alpha B}(\|\vec{r}_{\alpha B}\|) \tag{3.3}$$

[1] Note that in this case the repulsive potential has circular isoclines

- **Attractive forces**. Pedestrians are sometimes attracted by certain persons (e.g. friends, colleagues, etc.) or obstacles (e.g. shops, landmarks, etc.), denoted as i. This results in additional *attractive effects* that can be modeled by a monotonically increasing *attractive potential* $W_{\alpha i}(\|\vec{r}_{\alpha i}\|, t)$:

$$\vec{f}_{\alpha i}(\vec{r}_{\alpha i}, t) = -\nabla_{\vec{r}_{\alpha i}} W_{\alpha i}(\|\vec{r}_{\alpha i}\|, t) \tag{3.4}$$

  In this case, the main difference is that this form of *attractiveness* usually decreases with time t. This has to be taken into account when designing the *potential* $W_{\alpha i}$.

The model also considers that attractive and repulsive forces only affect the motion of pedestrians when they are perceived, i.e. when they move into the field of view. Situations that do not lie within the *angle of sight* $2\,\phi$ will have weaker influences c with $0 < c < 1$ on the pedestrian. This is modeled by direction dependent weights:

$$w(\vec{e}, \vec{f}) := \begin{cases} 1 & \text{if } \vec{e} \cdot \vec{f} \geq \left\|\vec{f}\right\| \ cos(\phi) \\ c & \text{otherwise} \end{cases} \tag{3.5}$$

where $\vec{e}$ denotes the desired direction of movement and $\vec{f}$ the respective force. With this, the final forces are obtained as:

$$\begin{aligned} \vec{F}_{\alpha\beta}(\vec{e}_\alpha, \vec{r}_\alpha - \vec{r}_\beta) &:= w(\vec{e}_\alpha, -\vec{f}_{\alpha\beta})\, \vec{f}_{\alpha\beta}(\vec{r}_{\alpha\beta}) \\ \vec{F}_{\alpha i}(\vec{e}_\alpha, \vec{r}_\alpha - \vec{r}_i, t) &:= w(\vec{e}_\alpha, -\vec{f}_{\alpha i})\, \vec{f}_{\alpha i}(\vec{r}_{\alpha i}, t) \end{aligned} \tag{3.6}$$

The resultant $\vec{F}_\alpha(t)$, denoting the pedestrian's total motivation for movement, is then the sum of all effects:

$$\vec{F}_\alpha(t) := \underbrace{\vec{F}_\alpha^0(\vec{v}_\alpha, v_\alpha^0 \vec{e}_\alpha)}_{\substack{\text{acceleration term} \\ \text{towards dest.}}} + \underbrace{\sum_\beta \vec{F}_{\alpha\beta}(\vec{e}_\alpha, \vec{r}_\alpha - \vec{r}_\beta)}_{\substack{\text{force} \\ \text{between pedestrians}}} + \underbrace{\sum_B \vec{F}_{\alpha B}(\vec{e}_\alpha, \vec{r}_\alpha - \vec{r}_B^{\,\alpha})}_{\substack{\text{force} \\ \text{between ped. and obstacles}}} + \underbrace{\sum_i \vec{F}_{\alpha i}(\vec{e}_\alpha, \vec{r}_\alpha - \vec{r}_i, t)}_{\text{attractive forces}} \tag{3.7}$$

The equation of motion

If we denote the preferred velocity of a pedestrian α at time t as $\vec{w}_\alpha(t)$, the Social Force model can be defined as:

$$\frac{d\vec{w}_\alpha}{dt} := \vec{F}_\alpha(t) \tag{3.8}$$

In order to complete this model, a relation between the actual velocity $\vec{v}_\alpha(t)$ and the preferred velocity $\vec{w}_\alpha(t)$ needs to be introduced. Since the speed of a pedestrian is limited by

a *maximum acceptable speed* v_α^{max}, the realized motion is finally given by:

$$\frac{d\vec{r}_\alpha}{dt} = \vec{v}_\alpha(t) := \vec{w}_\alpha(t)\, g\left(\frac{v_\alpha^{max}}{\|\vec{w}_\alpha\|}\right) \tag{3.9}$$

with

$$g\left(\frac{v_\alpha^{max}}{\|\vec{w}_\alpha\|}\right) := \begin{cases} 1 & \text{if } \|\vec{w}_\alpha\| \leq v_\alpha^{max} \\ \dfrac{v_\alpha^{max}}{\|\vec{w}_\alpha\|} & \text{otherwise} \end{cases} \tag{3.10}$$

This concludes the Social Force model. In this thesis, we use the model's capability to realistically describe collective effects of pedestrian motion behavior in crowded scenes in order to generate customized datasets that focus on the interactions between pedestrians. For this, we neglect attractive effects (Equation 3.4) and the repulsive forces between pedestrians and obstacles (Equation 3.3). For the repulsive forces between pedestrians, we choose a monotonically decreasing exponential potential. By varying the shape of this potential, we can freely control the impact of social interactions on the motion behavior of pedestrians in the datasets generated. This allows us to generate various datasets with different amounts of social interactions. A detailed description of this procedure is given in section 4.2.

3.3.2. The Vanilla LSTM model

Given a scene of pedestrians moving in crowded space, we often observe people completely altering their path or unexpectedly stopping. Traditional models like the Social Force model try to reproduce this complex behavior by defining a set of hand-crafted functions that seek to model the different properties of human motion. These functions are often inspired by existing concepts of physics and represent the current knowledge about human motion behavior. In this manner, traditional models could successfully describe collective effects of human motion. However, they are yet not able to fully capture all its properties, including the variety of unwritten rules, personal preferences and cultural aspects that influence the motion of pedestrians [10]. This motivates researchers to use data-driven models. These models do not depend on hand-crafted functions. Instead, they learn the key features of human motion behavior by processing a sufficient amount of suitable data. An important advantage of these models is that they are able to model properties that are yet not entirely understood or formalized. In this manner, data-driven models have been developed to learn the complex interactions between pedestrians. In the following, we introduce the Vanilla LSTM model: a data-driven model that uses a Recurrent Neural Network to predict human motion behavior.

The task of predicting the future movements of pedestrians can be understood as a sequence generation problem. As mentioned in section 3.2, this task consists of generating an output sequence of xy-coordinates that represents the future trajectory of a pedestrian, while receiving an input sequence of observed positions. The Vanilla LSTM model, is based on a LSTM encoder-decoder architecture. It represents a classical sequence-to-sequence model

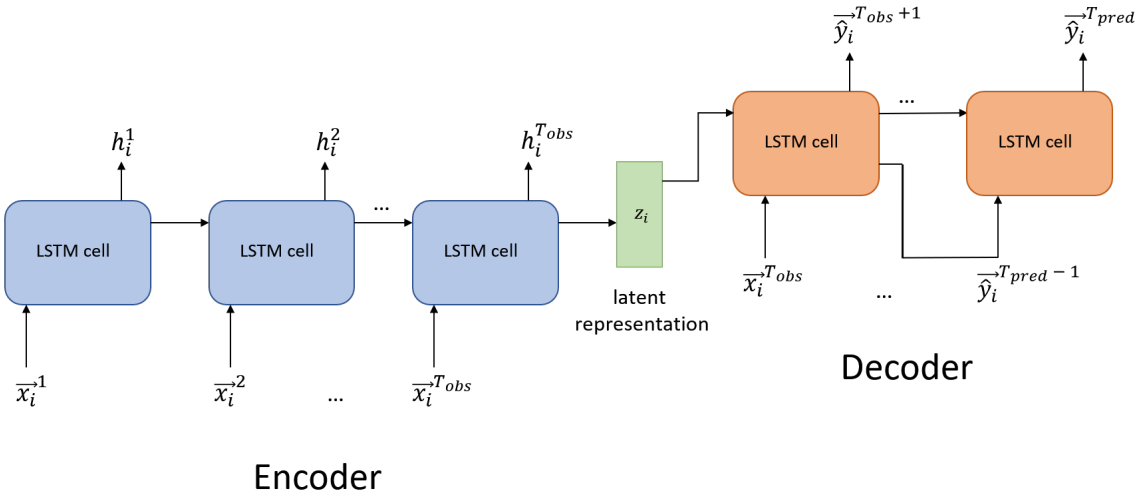


Figure 3.1.: Conceptional overview of the Vanilla LSTM model's encoder-decoder architecture for pedestrian i. The LSTM cells of the encoder take as input the observed motions $\vec{x}_i^t$ and learn to represent the state of the pedestrian in the compressed latent variable z_i. The decoder of the model uses this information, together with the last observed position $\vec{x}_i^{T_{obs}}$, to predict the pedestrian's future positions $\vec{\hat{y}}_i^t$.

[45], as the input and output sequences may differ in length. The model's architecture is visualized in Figure 3.1. It consists of three parts: the LSTM encoder, an intermediate latent representation z and the LSTM decoder.

LSTM encoder. In the encoder, one LSTM for each pedestrian learns the state of a person and represents it in a compressed latent variable z_i. The encoder consists of a stack of several LSTM cells, where each one accepts one position of the observed trajectory $\vec{x}_i^t$ and a hidden state of the previous LSTM cell h_i^{t-1}. It is advisable to embed the coordinates of the observed trajectory into a vector e_i^t before inserting them into a LSTM cell. If we denote W_e as the weights of the LSTM encoder and $LSTM(\cdot)$ as the output of a LSTM cell, the hidden states h_i^t can be computed using the formula:

$$\begin{aligned} e_i^t &= f\left(\vec{x}_i^t\,;\, W_{fe}\right) \\ h_i^t &= LSTM\left(e_i^t,\, h_i^{t-1}\,;\, W_e\right) \end{aligned} \tag{3.11}$$

where f is an embedding function with weights W_{fe}.

Latent representation. This compressed representation is the final hidden state of the LSTM encoder. It encapsulates the information of the whole input sequence and helps the decoder to make appropriate predictions. Furthermore, it acts as the initial hidden state of the LSTM decoder.

LSTM decoder. The LSTM decoder predicts the pedestrians' next movements based on the latent representation z and the last observed positions $\vec{X}^{T_{obs}}$. In order to predict the complete future trajectory of a pedestrian i, it then iteratively predicts the missing positions by considering the previous hidden state and the last predicted position $\vec{y}_i^{t-1}$ [45]. Any hidden state of the LSTM decoder is computed by using the following formula:

$$\begin{aligned} e_i^t &= f\left(\vec{y}_i^{t-1};\, W_{fd}\right) \\ h_i^t &= LSTM\left(e_i^t,\, h_i^{t-1};\, W_d\right) \end{aligned} \tag{3.12}$$

where f is again an embedding function with weights W_{fd}. The predicted positions $\vec{y}_i^t$ are obtained by applying a linear layer to the respective hidden position h_i^t in order to match the desired dimensions.

The parameters of the Vanilla LSTM model are learned by minimizing the Average Displacement Error (ADE) for all predicted trajectories in a dataset. If we assume N trajectories of the pedestrians in a given dataset, the ADE can be described by the following formula:

$$ADE = \frac{1}{N}\frac{1}{T_{pred} - (T_{obs}+1)}\sum_{i=1}^{N}\sum_{t=T_{obs}+1}^{T_{pred}}\left\|\vec{y}_i^t - \vec{\hat{y}}_i^t\right\| \tag{3.13}$$

3.3.3. The Social LSTM model

The Vanilla LSTM model uses one LSTM network for each pedestrian in a given scene in order to learn the pedestrian's state. As these LSTMs operate in isolation, the model predicts each trajectory independently of others and does not regard the movements of neighboring individuals. However, pedestrians socially interact with people in their vicinity. Therefore, their motions are influenced by the behavior of neighbors. Several models and concepts have been developed that aim to model these interactions in order to predict the future trajectories of pedestrians accurately [2, 15, 27, 42]. One example is the Social LSTM model proposed by Alahi *et al.* in 2016 [1]. This model uses a similar architecture as the Vanilla LSTM model, but adds a social pooling layer that connects the LSTM networks of neighboring individuals. In this work, we evaluate the extent to which this pooling scheme allows the Social LSTM model to exploit the motion behavior of neighboring pedestrians. In the following, we describe the concepts of this pooling scheme in more detail.

Social pooling of hidden states. As shown in Figure 3.2, at every time step during the prediction process, each LSTM cell shares its hidden-state information with LSTM cells of neighboring pedestrians. This is realized by a pooling layer that partially preserves spatial information through a grid-based pooling method. The hidden state $h_i^t \in \mathbb{R}^D$ of a LSTM cell captures the latent representation of pedestrian i at time t. This compressed information is shared with neighbors by building a *Social hidden-state tensor* H_i^t. If we denote the set of

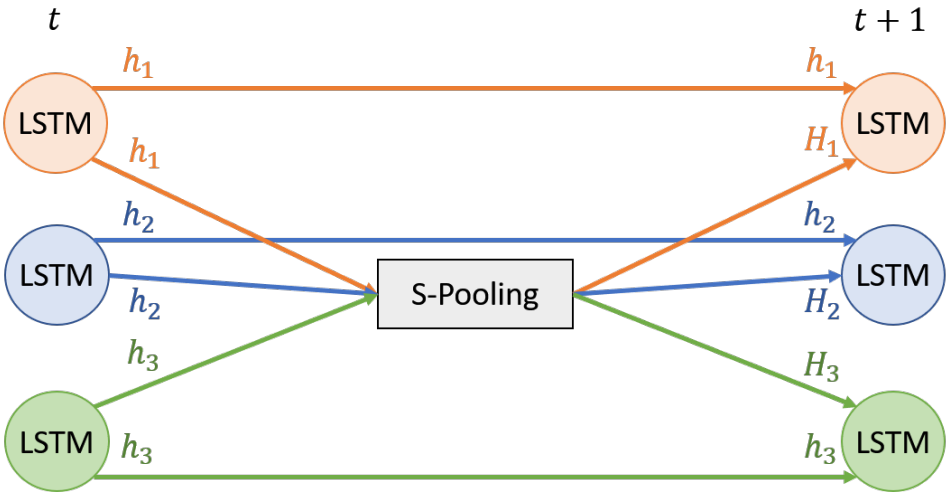


Figure 3.2.: Conceptional overview of the social pooling layer for three neighboring pedestrians. The hidden state information h_i^t of each pedestrian in a vicinity is passed to the pooling module at each time step of the prediction process. Once the hidden-state tensor H_i^t is calculated, it is passed to the LSTM cells of the corresponding pedestrians.

pedestrians in a certain *neighborhood* N_o of person i as $\mathcal{N}_i$, this tensor is defined as:

$$H_i^t(m,n,:) := \sum_{j \in \mathcal{N}_i} \mathbf{1}_{m\,n}[x_j^t - x_i^t,\, y_j^t - y_i^t]\, h_j^{t-1} \qquad H_i^t \in \mathbb{R}^{N_o \times N_o \times D} \tag{3.14}$$

where h_j^{t-1} is the hidden state of the LSTM corresponding to pedestrian j at time $t-1$ and $\mathbf{1}_{m\,n}[x,y]$ is an indicator function which ensures that the spatial information is preserved. In particular, the indicator function checks whether the person j is in the cell (m,n) of the grid. This operation is depicted in Figure 3.3. It visualizes the pooling process for a pedestrian (represented by a black dot) who is surrounded by three neighbors. In the example, the *neighborhood size ns* and *grid size gs* of the pooling scheme is defined such that the grid is divided into four equally spaced cells $\mathcal{C}_k$ with $k = 1, \ldots, 4$. Due to the spatial circumstances in

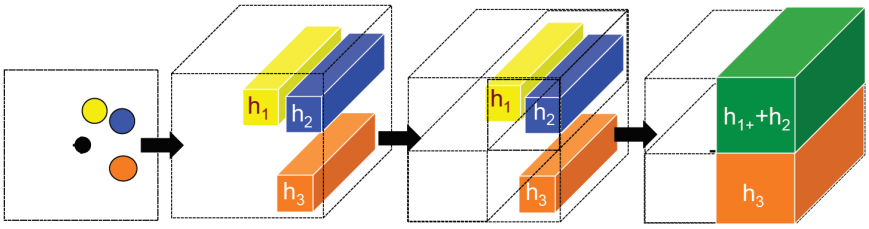


Figure 3.3.: Grid-based pooling method. The subdivision of a neighborhood N_o in multiple cells $\mathcal{C}$ with size *ns* preserves the spatial information of the scene [1] (Alahi *et al.*, 2016).

the given scene, pedestrians 1 and 2 belong to the same cell, whereas neighbor 3 is assigned a different cell. As shown in the last step of the pooling scheme, the hidden states of each cell are finally merged. This partially preserves the spatial information of the given scene.

The pooled Social hidden-state tensor H_i^t of pedestrian i is concatenated with the respective hidden state information of the previous LSTM cell h_i^{t-1} and embedded into a vector d_i^t. Together with the embedding of the previously predicted position $\vec{y}_i^{\,t-1}$, the obtained vector d_i^t is then inserted into the next LSTM cell to obtain the hidden state information h_i^t. This introduces the following recurrence:

$$\begin{aligned} e_i^t &= f\left(\vec{y}_i^{\,t-1};\, W_{fd}\right) \\ d_i^t &= f\left([h_i^{t-1}, H_i^t];\, W_{fh}\right) \\ h_i^t &= LSTM\left(e_i^t,\, d_i^t;\, W_d\right) \end{aligned} \tag{3.15}$$

where f denotes an embedding function with weights W_{fd}, or W_{fh} respectively. The LSTM weights are denoted as W_d. As for the Vanilla LSTM, the predicted positions $\vec{y}_i^t$ are finally obtained by applying a linear transformation to the respective hidden position h_i^t to match the desired dimensions.

The parameters of the Social LSTM model are learned by minimizing the average displacement error for all trajectories in a training set, as described in Equation 3.13. Note that in comparison to the Vanilla LSTM model in subsection 3.3.2, the hidden states of multiple LSTM networks are coupled by the social pooling layer. The model thus needs to backpropagate through multiple LSTMs in a scene at every time step [1]. Therefore, the training of the Social LSTM model is computationally more expensive than the respective training of the Vanilla LSTM model.

As mentioned before, data-driven models like the Vanilla LSTM model or the Social LSTM model learn the motion behavior of pedestrians by extracting key features from the data they process. A critical drawback of these models is that they strictly depend on the nature of the data provided. In particular, they can only learn properties of human motion behavior that is represented in the respective datasets. In the next chapter, we will therefore analyze datasets that are commonly used to train and evaluate data-driven models for trajectory prediction and discuss their suitability for the evaluation of models that aim to predict social interactions.

4. Datasets

In this chapter, we present two publicly available datasets of real-world human trajectories and discuss their suitability for the evaluation of data-driven models that aim to learn social interactions. We further describe how the Social Force model can be adapted in order to generate hand-tailored synthetic datasets that focus on the interactions between pedestrians and demonstrate the usability of this approach. We conclude this chapter by giving an outlook on how we can address human-space interactions when generating synthetic data with this approach.

4.1. Real datasets

There exists a variety of publicly available datasets of real-world human trajectories that can be used for pedestrian trajectory prediction. Examples can be found in [34, 40, 52], to name a few. However, most of the works mentioned in chapter 2 evaluate their models on two particular datasets: ETH [38] and UCY [30]. As shown in [38], these datasets cover complex motion behaviors, such as group crossings or collision avoidance. The datasets are labeled manually at a frame rate of 2.5 fps in order to obtain real world coordinates every 0.4 seconds. They cover four different scenarios with five sets of data. In particular, the ETH dataset comprises two sets of data, denoted as ETH and Hotel, with 750 pedestrians present. The UCY dataset includes three different sets of data, i.e. Zara01, Zara02 and Univ, containing 786 pedestrians in total. A more detailed overview of the properties of each dataset can be found in Table 4.1. We can observe that the Univ dataset includes the highest number of pedestrians while comprising the smallest number of frames. Therefore, we expect crowded scene-behavior for this dataset.

In this thesis, we predict a trajectory by observing 3.2 seconds (8 time steps) of motion and forecasting the next 4.8 seconds (12 time steps). The complete trajectory of a pedestrian thus consists of 8 seconds (20 time steps). In the given datasets, we regularly find pedestrians that

Table 4.1.: Overview of the properties of the ETH and UCY datasets.

	Dataset	**Number of frames**	**Number of pedestrians**
ETH	ETH	1934	360
	HOTEL	1807	390
UCY	UNIV	540	434
	ZARA01	902	148
	ZARA02	1052	204

are only present for a shorter period of time or leave the scene for a moment and return later. This results in trajectories that consist of less than 20 time steps or miss information at certain points. There are several approaches to solve this problem, such as padding or interpolating the data. However, this can introduce non-negligible errors that produce misleading results. Therefore, we only evaluate the performance of a model on complete trajectories that consist of at least 20 consecutive time steps. This results in a significant reduction of usable trajectories in a given dataset. In Table 4.2, we show that the amount of suitable trajectories in each dataset is significantly less than the total number of trajectories present. In particular, each trajectory that consists of at least 20 consecutive time steps is counted as a suitable trajectory. The total number of trajectories also includes trajectories that consist of less than 20 time steps. As an example, a trajectory that consists of 16 consecutive time steps is represented in the total number of trajectories but does not count as a suitable trajectory. A trajectory that consists of 40 consecutive time steps represents two suitable trajectories, since we can split it into two paths, each comprising 20 consecutive time steps.

Table 4.2.: Comparison between the total number of trajectories in the datasets and the amount of suitable trajectories for the prediction task.

Dataset	Total number trajectories	Nr. of suitable trajectories	% of suitable trajectories
ETH	649	297	45.76 %
HOTEL	511	145	28.38 %
UNIV	1302	903	69.35 %
ZARA01	320	178	55.63 %
ZARA02	572	374	65.38 %

We note that especially for the Hotel dataset, the number of suitable trajectories for the prediction task is significantly lower than the total number of trajectories. Overall, there is no dataset for which the amount of suitable trajectories is higher than 70% of the total number of trajectories present. This is a significant limitation for the evaluation of data-driven models on these datasets.

4.1.1. Performance evaluation on real datasets

In order to evaluate the performance of the Vanilla LSTM model and Social LSTM model on these datasets, we train and validate the models on 4 sets of data and test them on the remaining set. Similarly to previous works [1, 2, 15], we use the following two evaluation metrics:

1. Average Displacement Error: evaluates the average euclidean distance between prediction $\vec{\hat{y}}_i^t$ and ground truth $\vec{y}_i^t$ over all time steps t of the prediction process for all suitable trajectories N in a dataset:

$$ADE = \frac{1}{N} \frac{1}{T_{pred} - (T_{obs} + 1)} \sum_{i=1}^{N} \sum_{t=T_{obs}+1}^{T_{pred}} \left\| \vec{y}_i^t - \vec{\hat{y}}_i^t \right\| \tag{4.1}$$

2. Final Displacement Error: evaluates the euclidean distance between the prediction of the final position $\vec{\hat{y}}_i^{T_{pred}}$ and the respective ground truth data $\vec{y}_i^{T_{pred}}$ for all suitable trajectories N in a dataset:

$$FDE = \frac{1}{N} \sum_{i=1}^{N} \left\| \vec{y}_i^{T_{pred}} - \vec{\hat{y}}_i^{T_{pred}} \right\| \tag{4.2}$$

Implementation details. In order to obtain an adequate comparison between the two models, we use similar configurations for both models. In particular, we use an embedding dimension of 32 for the spatial coordinates before using them as input to the LSTM cells of the models. We define the dimension of the hidden state tensor of the encoder to be 64 and the decoder to be 32. In addition, we use a layer with ReLU (Rectified Liner Units) nonlinearity between these states to match the respective dimensions. For the Social LSTM model, we set the neighborhood size ns to be 10 meters and use a grid size of 10 meters, such that each cell C_k spans one square meter. Furthermore, we use a layer with ReLU nonlinearity on top of the calculated hidden state tensor H_i^t before concatenating it with the hidden state information of the previous LSTM cell h_i^{t-1}, as described in subsection 3.3.3. The hyper-parameters are chosen for each dataset separately, based on the evaluation of the corresponding validation set. For the optimization, we use Adam [24].

The evaluation results for both models are shown in Table 4.3. As it can be seen, the Vanilla LSTM model outperforms the Social LSTM model on the UCY datasets (Zara01, Zara02 and Univ). While both models have the same average displacement error for the ETH dataset, the Social LSTM model performs slightly better on this dataset with respect to the FDE. For the Hotel dataset, the Social LSTM model outperforms the Vanilla LSTM model.

In this thesis, we focus on the capacity of a model to predict social interactions. Therefore, the number of social interactions present in each dataset is crucial. As shown in [38], the ETH and UCY datasets cover complex motion behavior including social interactions between multiple pedestrians. However, the evaluation of these datasets, depicted in Table 4.3, does not

Table 4.3.: Prediction errors of the Vanilla LSTM model and the Social LSTM model for the ETH and UCY datasets. The ADE and FDE values are separated by a slash, where the first value indicates the ADE. All values represent meters. The last row denotes the average values over all datasets.

	Dataset	**Vanilla LSTM**	**Social LSTM**
ETH	ETH	1.22 / 2.60	1.22 / **2.54**
	HOTEL	0.37 / 0.71	**0.28** / **0.55**
UCY	UNIV	**0.61** / **1.22**	0.69 / 1.42
	ZARA01	**0.47** / **0.99**	0.56 / 1.15
	ZARA02	**0.36** / **0.75**	0.45 / 0.92
	AVG	**0.61** / **1.25**	0.64 / 1.32

clearly indicate whether one of the analyzed models is capable of learning social interactions or not. In particular, we would assume a model that is not capable of modeling social interactions to perform poorly on datasets with many social interactions present. Yet, we do not know the specific number of social interactions present in the given datasets.

To address this problem, we generate hand-tailored datasets for which we can define the intensity and range of the social interactions between pedestrians. This allows us to compare the performance of the models on datasets with variable amounts of social interactions present. By generating synthetic datasets, we also overcome the above-mentioned problem of limited suitable trajectories. Generating synthetic datasets allows us to create an almost unlimited amount of data and, therefore, an almost unlimited amount of suitable trajectories.

4.2. Generation of synthetic datasets

Due to the problems mentioned in section 4.1, we generate synthetic datasets that are focused on social interactions between pedestrians. For this, we adapt the Social Force model introduced in subsection 3.3.1. This model uses attractive and repulsive forces to describe the motion behavior of pedestrians. As described in Equation 3.7, the resulting force that governs the motion of a pedestrian is composed of four components: an acceleration term towards the destination, a repulsive force modeling human-human interactions, a repulsive force describing human-space interactions and an attractive force towards specific regions of interest. By manipulating the repulsive force between pedestrians, we can conveniently define the influence of social interactions in the modeled data. As we focus on these interactions, we neglect the impact of obstacles or attractive effects in a scene. The resultant force in Equation 3.7 then simplifies to:

$$\vec{F}_\alpha(t) := \underbrace{\vec{F}_\alpha^{\,0}(\vec{v}_\alpha, v_\alpha^0 \vec{e}_\alpha)}_{\substack{\text{acceleration term} \\ \text{towards dest.}}} + \underbrace{\sum_\beta \vec{F}_{\alpha\beta}(\vec{e}_\alpha, \vec{r}_\alpha - \vec{r}_\beta)}_{\substack{\text{force} \\ \text{between pedestrians}}} \tag{4.3}$$

where the accelerating term towards the destination is described by Equation 3.1 and the repulsive force between pedestrians is defined by the gradient of a repulsive potential, as described in Equation 3.2. The closer two pedestrians get to each other, the stronger they interact. In order to model this effect, we choose the repulsive potential to be a monotonically decreasing exponential function:

$$V_{\alpha\beta}(\left\|\vec{r}_{\alpha\beta}\right\|) := V^0 \, e^{\frac{-\left\|\vec{r}_{\alpha\beta}\right\|}{\sigma}} \tag{4.4}$$

where V^0 and σ are free parameters and $\left\|\vec{r}_{\alpha\beta}\right\|$ denotes the distance between two pedestrians α and β. Note that by varying the values of V^0 and σ, we can control the shape of this potential and, therefore, the impact of social interactions on the motion of pedestrians in the dataset.

4.2.1. Influence of V^0 and σ on social interactions

In order to specify the magnitude and range of social interactions in the dataset generated, we manipulate V^0 and σ in Equation 4.4. In particular, V^0 denotes the initial amplitude of the repulsive potential at a distance of zero. It therefore directly influences the magnitude of social interactions. The higher the value of V^0, the stronger pedestrians interact with each other. The value of σ determines how fast the repulsive potential in Equation 4.4 decreases. Table 4.4 shows for different values of σ, at which distance the repulsive potential $V(\|\vec{r}\|)$ has decreased to 10% of its initial value V^0. We can observe that with rising values of σ, the range of the potential and, thus, the range of pedestrians influencing each other increases.

Table 4.4.: Increasing values of σ represent rising distances for which the repulsive potential drops to 10% of its initial value V^0.

σ	**r [m] for $V(r) = 0.1V^0$**
0.2171	0.5
0.4343	1.0
0.8686	2.0
1.303	3.0
1.7371	4.0
2.171	5.0
2.6058	6.0

Hence, these parameters allow us to efficiently determine the impact of social interactions on the motion behavior of pedestrians. As an example, Figure 4.1 depicts scenes from two datasets with different values for V^0 and σ, in which the values of the repulsive potential is visualized. We can see that the dataset on the left, generated with low values for V^0 and

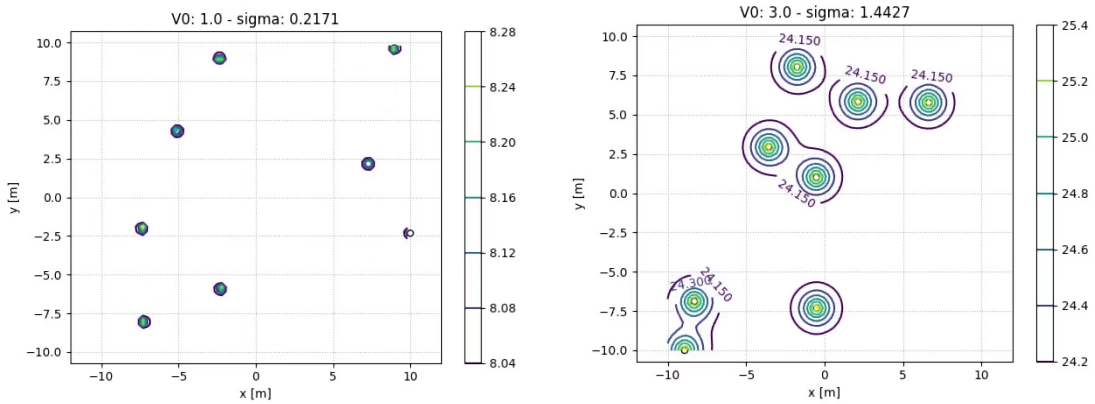


Figure 4.1.: Scenes from two different datasets simulated by our adaption of the Social Force model. While the left scene represents a dataset with low values $V^0 = 1$ and $\sigma = 0.2171$, the right scene is from a dataset with comparatively high values $V^0 = 3$ and $\sigma = 1.4427$. The visualized repulsive potentials differ in magnitude and range respectively.

σ, results in a weak and narrow repulsive potential. Contrarily, the dataset on the right, generated with high values for V^0 and σ, leads to a strong and far-ranging potential. In this manner, we create datasets for 35 different combinations of values for V^0 and σ. The particular values can be found in Table 4.5. These datasets are characterized by a specific amount of social interactions present. Each dataset describes a scenario for which a predefined number of pedestrians move in a square of size 20x20 square meters. As we focus on social interactions between pedestrians, we omit obstacles in the scenes, such that the motion of the pedestrians is only influenced by other individuals. Each pedestrian in the dataset enters the square at one randomly selected side and leaves it at another randomly selected side. The specific entry and exit locations are uniformly distributed over the respective sides. The positions of all pedestrians are updated according to Equation 3.9, where we define $dt := \Delta t = 0.4$ seconds, similar to the real-world data described in section 4.1. The respective coordinates are labeled and saved in a text file according to the datasets of real-world human motion. Every time a pedestrian leaves the scene, a new pedestrian enters the scene at one side of the square. This ensures a constant number of pedestrians N_{const} at all time steps t. The pedestrians have an initial velocity $\vec{v}_\alpha^{init}$ that is directed towards their destination. The respective magnitude is uniformly distributed on the interval $[v_{min}, v_{max})$, where we define $v_{min} = 0.4\frac{m}{s}$ and $v_{max} = 1.2\frac{m}{s}$. For the relaxation time in Equation 3.1, we choose $\tau_\alpha = 0.5$ according to the suggestions made in [19]. Note that smaller values of τ_α lead to a more aggressive gait that does not fit our preferences. In general, we limit the speed of a pedestrian by $v_\alpha^{max} = 1.3\, v_\alpha^0$. In order to take head movements of the pedestrians into account, we choose the effective angle of sight in Equation 3.5 to be $2\phi = 200°$. Other pedestrians outside this angle are assumed to have an influence of $c = 0.5$.

By evaluating the performances of the Vanilla LSTM model and the Social LSTM model on these datasets, we can systematically analyze their ability to learn and predict social interactions. This is done in chapter 5. In the following, we give a brief outlook on how the Social Force model can be adapted to generate datasets that additionally take the interactions between pedestrians and obstacles in their physical environment into account.

Table 4.5.: Specific values of V^0 and σ used for the generation of the datasets.

V^0 **values**
0
1
2
4
6

σ **values**
0.2171
0.4343
0.8686
1.303
1.7371
2.171
2.6058

4.2.2. Simulating obstacles for human-space interactions

In this thesis, we focus on human-human interactions and neglect human-space interactions between obstacles and pedestrians. Therefore, we omit obstacles in the datasets we generate. However, human-space interactions are of great interest for many application areas. In this section, we briefly describe how the impact of obstacles can be taken into account while generating synthetic datasets with the introduced Social Force model. This could be of interest for ongoing future works on this subject.

In order to model the impact of obstacles in the datasets generated, we need to consider the repulsive effects between pedestrians and obstacles (see Equation 3.3) in the resultant force of the Social Force model:

$$\vec{F}_\alpha(t) = \underbrace{\vec{F}_\alpha^{\,0}(\vec{v}_\alpha, v_\alpha^0 \vec{e}_\alpha)}_{\substack{\textit{acceleration term} \\ \textit{towards dest.}}} + \underbrace{\sum_\beta \vec{F}_{\alpha\beta}(\vec{e}_\alpha, \vec{r}_\alpha - \vec{r}_\beta)}_{\substack{\textit{force} \\ \textit{between pedestrians}}} + \underbrace{\sum_B \vec{F}_{\alpha B}(\vec{e}_\alpha, \vec{r}_\alpha - \vec{r}_B^{\,\alpha})}_{\substack{\textit{force} \\ \textit{between ped. and obstacles}}} \tag{4.5}$$

Similar to the repulsive effects between pedestrians, the respective forces are modeled by the gradient of a monotonically decreasing repulsive potential $U_{\alpha B}(\|\vec{r}_{\alpha B}\|)$. Based on the choice of this potential, the impact of human-space interactions on pedestrians is defined. It is desirable that the data generated to model a specific situation or property is as realistic as possible. In order to model scenarios of real-world datasets, we implement a method that reads in segmented scene context information and embeds it in the process of data generation. In particular, the segmentation of a real-world image is processed by approximating the boundary boxes of specific obstacle classes. These approximations are then used to define the borders of obstacles B in Equation 3.3. An example of this process is depicted in Figure 4.2. As it can be seen, the scene context from the Zara01 scenario on the left is segmented and then embedded in the process of generating a corresponding synthetic dataset, as depicted on the right. This allows for the generation of synthetic datasets based on real-world scenarios.

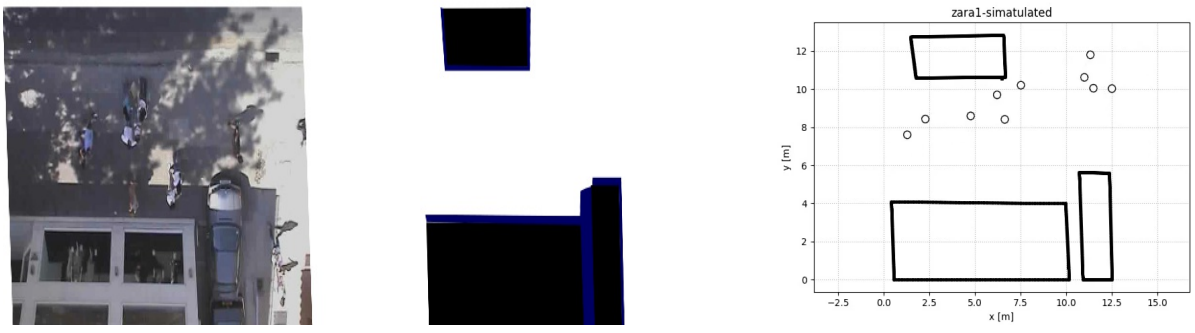


Figure 4.2.: An image of the Zara01 scenario from the UCY dataset (image on the left) is segmented in order to obtain the boundaries of the obstacles in the scene (middle image). This segmentation is then used to simulate a corresponding dataset with our adaption of the Social Force model (image on the right).

5. Experiments

We begin this chapter by explaining the general configurations of the experiments we conduct on the Vanilla LSTM model and the Social LSTM model in order to analyze their capability of learning social interactions. For a detailed analysis, we introduce various evaluation metrics that go beyond the classical average and final displacement error. These metrics allow us to show quantitatively that the Social LSTM model outperforms the Vanilla LSTM model in predicting trajectories that are influenced by social interactions. A subsequent qualitative evaluation supports these results. In particular, we can show that while the Vanilla LSTM model is not sensitive to social interactions, the Social LSTM model is capable of predicting these interactions. The respective experiments were implemented in Python using PyTorch. The implementation to reproduce the results or to generate the corresponding datasets can be found at: https://github.com/PMMon/Thesis_Social_Interactions.

As described in section 4.2, we train and evaluate the introduced models on synthetic datasets that focus on the interactions between pedestrians. In particular, we generate datasets that in total represent 35 different combinations of the values for V^0 and σ. With this, we obtain datasets that gradually differ in the amount of social interactions present. A detailed overview of these combinations can be found in Table 5.1. Note that the impact of social interactions on the pedestrians increases with rising values of V^0 and σ. Each dataset represents a scenario, in which a predefined constant number of pedestrians N_{const} move in a square of 20x20 meters. By varying this number, we are able to define the density of pedestrians in a scene. For each pair of V^0 and σ, we generate three sets of data: a set on which we train the models, a validation set and a test set on which we evaluate the models. An overview of the configurations of these sets is given in Table 5.2.

Table 5.1.: Detailed overview of the different combinations of the values for V^0 and σ. For each pair of these values, we generate datasets on which we evaluate the models introduced.

V^0	σ						
6	0.2171	0.4343	0.8686	1.303	1.7371	2.171	2.6058
4	0.2171	0.4343	0.8686	1.303	1.7371	2.171	2.6058
2	0.2171	0.4343	0.8686	1.303	1.7371	2.171	2.6058
1	0.2171	0.4343	0.8686	1.303	1.7371	2.171	2.6058
0	0.2171	0.4343	0.8686	1.303	1.7371	2.171	2.6058

Phase	Nr. Frames	Space of Square [m]
Train	9000 (1h)	20x20
Val	1800 (12 min)	20x20
Test	1800 (12 min)	20x20

Table 5.2.: General configurations of the different sets of data. The time span between two frames is 0.4 seconds. As the training set comprises 9000 frames, it represents one hour of motion behavior. The validation and test set are limited to 12 minutes of motion.

Evaluation Methodology. As mentioned in section 3.2, the introduced models predict the future paths of pedestrians based on the observation of previous motions. Similar to prior work [1, 15, 27], we observe each trajectory for 8 time steps (3.2 seconds) and predict the next 12 time steps (4.8 seconds). Respectively, we set $T_{obs} = 8$ and $T_{pred} = 20$.

5.1. Evaluation Metrics

In order to conclude the extent to which a trajectory prediction model is capable of predicting social interactions between pedestrians, we introduce various evaluation metrics. In addition to the commonly used average and final displacement error, we propose new metrics that focus on the interactions between pedestrians. In this manner, we evaluate the average displacement error in nonlinear regions, measure the error on classified trajectories and analyze the collision behavior of the predicted paths.

5.1.1. Average and final displacement error

The most commonly used evaluation metrics for pedestrian trajectory prediction are the average displacement error (Equation 4.1) and the final displacement error (Equation 4.2). Similar to previous studies [1, 2, 15, 27], we use these metrics to evaluate the overall performance of the introduced models on the datasets generated. However, as mentioned in section 3.2, these metrics are not sufficient to make well-founded statements about the ability of a model to predict social interactions. Therefore, we introduce additional evaluation metrics.

5.1.2. Average nonlinear displacement error

As mentioned in section 3.2, social interactions between pedestrians cause their respective trajectories to become nonlinear. In particular, while weak interactions have only a small impact on the motion of pedestrians, strong interactions result in distinct nonlinear regions. This motivates us to create a metric that measures the average displacement error in nonlinear regions of the trajectories. In order to identify these regions, we calculate the curvature of a trajectory at discrete points.

Menger curvature. Note that in this thesis, we observe each trajectory for 8 time steps and predict the next 12 time steps. Hence, the future trajectory $Y_i = (\vec{y}_i^{T_{obs}+1}, \vec{y}_i^{T_{obs}+2}, \ldots, \vec{y}_i^{T_{pred}})$

of pedestrian i comprises 12 different positions. We use a three-point circle approximation scheme, denoted as Menger curvature, to calculate the curvature of the trajectory at ten different positions $\vec{y}_i^t$ with $t \in \{T_{obs}+2, \ldots, T_{pred}-1\}$.

Let $\vec{y}_i^{t-1}$, $\vec{y}_i^t$, $\vec{y}_i^{t+1}$ be three consecutive points of the future trajectory Y_i. As depicted in Figure 5.1, we define R to be the radius of a circle passing through these points. We further denote S to be the area of the triangle $\triangle \vec{y}_i^{t-1}\, \vec{y}_i^t\, \vec{y}_i^{t+1}$ that connects these points, and define a, b and c as the sides of the triangle:

$$a := \left\| \vec{y}_i^t - \vec{y}_i^{t-1} \right\| \qquad b := \left\| \vec{y}_i^{t+1} - \vec{y}_i^t \right\| \qquad c := \left\| \vec{y}_i^{t+1} - \vec{y}_i^{t-1} \right\| \tag{5.1}$$

We then calculate the curvature k_i^t of the future trajectory of pedestrian i at position $\vec{y}_i^t$ by using the formula for the Menger curvature [6]:

$$k_i^t := \frac{1}{R} = \frac{4S}{abc} \qquad \text{for } t \in \{T_{obs}+2, \ldots, T_{pred}-1\} \tag{5.2}$$

Note that as we use a three-point approximation scheme, we do not calculate the curvature at the first and last point of the future trajectory Y_i. Hence, we obtain a 10-tuple of curvature values $k_i = \left(k_i^{T_{obs}+2}, \ldots, k_i^{T_{pred}-1}\right)$ for each pedestrian i in the scene. In addition, we define a threshold value td that allows us to identify regions of the trajectory, for which the curvature is higher than or equal to the threshold: $k_i^t \geq td$. Finally, we obtain the average nonlinear displacement error by calculating the ADE for these regions, excluding positions of a trajectory with $k_i^t < td$.

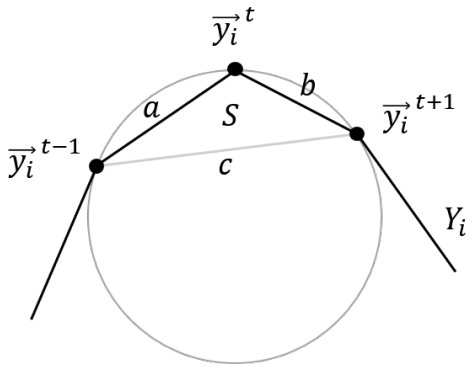


Figure 5.1.: Schematic overview of the circle approximation scheme that we use to calculate the curvature k_i^t at position $\vec{y}_i^t$ of the trajectory.

5.1.3. ADE and FDE on classified trajectories

We observe that while the path of a pedestrian who is not influenced by social interactions remains linear, the trajectory of a pedestrian who is influenced by these interactions becomes nonlinear. In order to compare the performance of the models on these different motions, we classify the future trajectories Y of a dataset with respect to their degree of nonlinearity.

We then evaluate the average and final displacement error for each class. This allows us to compare the performance of a model on uninfluenced trajectories with its performance on motions that are highly influenced by social interactions.

Classification scheme. As described in subsection 5.1.2, we calculate the curvature of a future trajectory Y_i at ten different positions, obtaining a 10-tuple of curvature values $k_i = (k_i^{T_{obs}+2}, \dots, k_i^{T_{pred}-1}) = (k_i^{10}, \dots, k_i^{19})$ for each pedestrian i. Based on these values, we heuristically define the following four trajectory-classes:

1. *Strictly linear*: A trajectory Y_i is strictly linear when all its respective curvature values are low. In particular, we define the set of strictly linear trajectories SL to be:

$$SL \coloneqq \{\, Y_i \in Y \mid k_i^t \leq 0.11 \quad \forall\, t \in \{T_{obs}+2, \dots, T_{pred}-1\} \,\} \tag{5.3}$$

2. *Linear*: A trajectory Y_i appears to be linear when there are no two consecutive points $\vec{y}_i^t$ and $\vec{y}_i^{t+1}$ for which the trajectory has high curvature values. However, there can be isolated points where the trajectory has a slightly higher curvature value than for the strictly linear case. In particular, we define the set of linear trajectories L to be:

$$\begin{aligned} L \coloneqq \{\, Y_i \in Y \mid\ & k_i^t \leq 0.4 \quad \forall\, t \in \{T_{obs}+2, \dots, T_{pred}-1\} \wedge \\ & (\forall\, t \in \{T_{obs}+2, \dots, T_{pred}-2\} : 0.11 < k_i^t \leq 0.4 \implies k_i^{t+1} \leq 0.11) \,\} \end{aligned} \tag{5.4}$$

3. *Gradually nonlinear*: A trajectory Y_i is gradually nonlinear when it has high curvature values at three or more consecutive points. In order to distinguish between gradually and highly nonlinear trajectories, we limit the curvature values of gradually nonlinear trajectories to a certain maximum. In particular, we define the set of nonlinear trajectories GNL to be:

$$\begin{aligned} GNL \coloneqq \{\, Y_i \in Y \mid\ & k_i^t < 0.7 \quad \forall\, t \in \{T_{obs}+2, \dots, T_{pred}-1\} \wedge \\ & (\exists\, t \in \{T_{obs}+2, \dots, T_{pred}-3\} :\ 0.2 \leq k_i^t, k_i^{t+1}, k_i^{t+2} < 0.7) \,\} \end{aligned} \tag{5.5}$$

4. *Highly nonlinear*: A trajectory Y_i appears to be highly nonlinear when it has high curvature values at three or more consecutive points $\vec{y}_i^t$, $\vec{y}_i^{t+1}$ and $\vec{y}_i^{t+2}$. We define the set of highly nonlinear trajectories HNL as:

$$HNL \coloneqq \{\, Y_i \in Y \mid \exists\, t \in \{T_{obs}+2, \dots, T_{pred}-3\} : 1.0 \leq k_i^t, k_i^{t+1}, k_i^{t+2} \,\} \tag{5.6}$$

This classification scheme allows us to conveniently label future trajectories with respect to their degree of nonlinearity, or in other words, how strongly they were influenced by social interactions. Note that while L, GNL and HNL are mutually exclusive, SL is a subset of L, i.e. $SL \subseteq L$. Furthermore, the union of the above-defined sets does not include all future trajectories Y, i.e. $Y \setminus (L \cup GNL \cup HNL) \neq \varnothing$.

(a) strictly linear (b) linear (c) gradually nonlinear (d) highly nonlinear

Figure 5.2.: Exemplary trajectories of the four different trajectory-classes defined above.

We therefore introduce a fifth class *Other* and assign the remaining future trajectories to it:

$$Other := Y \setminus (L \cup GNL \cup HNL) \tag{5.7}$$

At this point we want to stress that the goal of this classification scheme is not to classify all future trajectories to a distinguishable class, such that the introduced *Other*-class becomes obsolete. Instead, we want to evaluate the performance of the models on sets of trajectories that vary in their degree of nonlinearity. In this manner, the classification scheme extracts specific trajectories that share the above-defined characteristics.

5.1.4. Collision behavior

The collision of two or more pedestrians is a rare event due to social interactions between individuals. Humans normally respect the personal space of others and avoid close contact with strangers. Yet, whenever pedestrians collide, they momentarily react and strongly interact with each other. This results in a highly complex alternation of their respective paths. Collisions, therefore, have a significant impact on the motion of pedestrians. Models that are capable of learning social interactions should predict collision behavior that is close to that of the given dataset. Hence, we create a metric that evaluates the collision behavior of the predicted trajectories of a model and compare it with the collision behavior of the ground truth data.

Two pedestrians collide with each other when their distance drops below a certain threshold value r_{coll}. For the collision behavior of the ground data, we calculate the euclidean distance r^t between all pedestrians in a scene N_{const} for each time step of the prediction process $t = \{T_{obs} + 1, \ldots, T_{pred}\}$. This is done for each prediction cycle. In order to focus on pedestrians that interact with each other, we neglect distances that do not lie in the interval $r = [0, r_{max}(\sigma)]$, where the value of $r_{max}(\sigma)$ is given by the σ-value of the dataset and the mapping defined in Table 4.4. Once the values are calculated, we plot the distribution of distances between pedestrians. In order to obtain the respective distribution for the model's predicted motion behavior, this process is repeated, but now the euclidean distance $\widetilde{r}^t$ between the predicted positions of all pedestrians in a scene is calculated. Finally, these distributions are compared. The corresponding collision behavior can be obtained by analyzing the distributions for the region $r < r_{coll}$.

5.2. Quantitative Evaluation

We begin the quantitative evaluation of the Vanilla LSTM model and the Social LSTM model by analyzing their overall performance on synthetic datasets with varying amounts of social interactions present. For this, we evaluate the average and final displacement error on 35 different datasets. Each dataset is generated according to the method described in section 4.2, using a high number of pedestrians in each scene $N_{const} = 14$ and a unique combination of the values for V^0 and σ, as defined in Table 5.1. For the Vanilla LSTM model and the Social LSTM model, we use the same configurations as for the evaluation on datasets of real-world human motion. A detailed description of these configurations can be found in subsection 4.1.1.

In order to conveniently characterize the datasets generated with respect to the impact of social interactions present, we classify the future trajectories Y of a dataset according to the in subsection 5.1.3 introduced classification scheme and calculate the following weighted sum ws:

$$ws := w_1 \cdot \frac{|L|}{|L \cup GNL \cup HNL|} + w_2 \cdot \frac{|GNL|}{|L \cup GNL \cup HNL|} + w_3 \cdot \frac{|HNL|}{|L \cup GNL \cup HNL|} \tag{5.8}$$

where $|X|$ denotes the cardinality of a set X and the weights are defined as follows:

- linear: $w_1 = 0$
- gradually nonlinear: $w_2 = 0.5$
- highly nonlinear: $w_3 = 1.0$

This weighted sum represents the distribution of classified trajectories in a dataset. It therefore gives a first impression of the influence of social interactions on the motion of the pedestrians in a dataset. As an example, a value of $ws = 0$ indicates that all successfully classified trajectories of a dataset are linear. Higher values for ws imply that there are also gradually or highly nonlinear trajectories in the dataset. In particular, the closer the value of ws gets to one, the higher the amount of highly nonlinear trajectories in the dataset. Note that as trajectories that are assigned to the *Other*-class do not represent an interpretable trajectory-class, we neglect them for the calculation of ws. The weighted sum for each dataset is depicted in Figure 5.3. We can see a clear trend that with rising values of V^0 and σ, the value of ws and, therefore, the amount of nonlinear trajectories in the dataset increases. This confirms that the created datasets differ in their amount of social interactions. While datasets with high values of V^0 and σ comprise many social interactions that cause trajectories to become highly nonlinear, datasets with low values for V^0 and σ barely include these interactions, such that the motion of the pedestrians is predominantly linear. This is reflected in the values of ws.

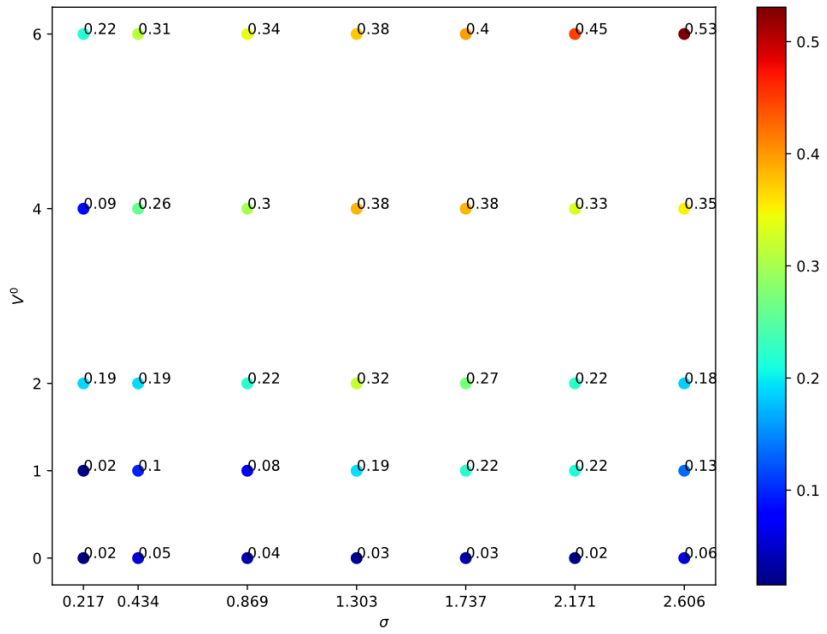
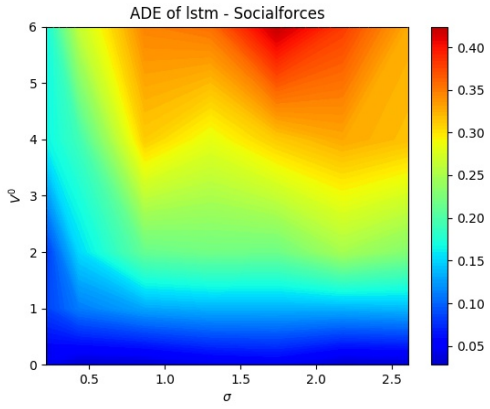


Figure 5.3.: Weighted sums calculated for each generated dataset according to Equation 5.8. For increasing values of V^0 and σ, we observe rising values of ws. This shows that the number of gradually and highly nonlinear trajectories in a dataset increases with increasing values of V^0 and σ.

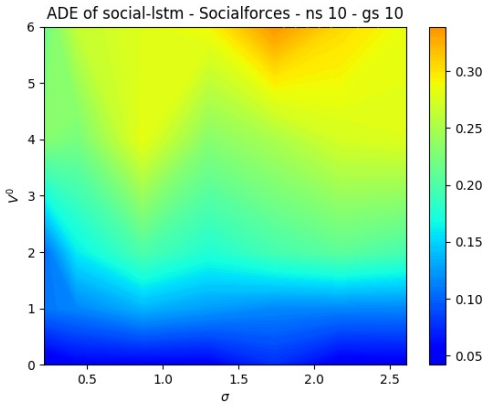
Average and final displacement error

The average and final displacement error of the Vanilla LSTM model and the Social LSTM model are calculated for all datasets. The respective results are depicted in Figure 5.4. We observe that while the Vanilla LSTM model performs comparatively well on datasets with weak or spatially-limited social interactions, the performance significantly decreases on datasets with strong interactions. As we analyze the results of the model's average displacement error in Figure 5.4a, we can particularly see that with rising values of V^0 and σ the ADE rapidly increases. This trend also reflects in the final displacement error of the model, as illustrated in Figure 5.4c. Note that the final displacement error is in general higher than the average displacement error of the model.

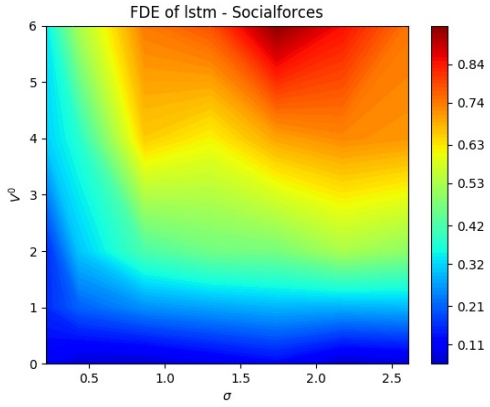
In comparison, we illustrate the ADE of the Social LSTM model in Figure 5.4b. While we can observe a similar trend towards higher average displacement errors for a rising impact of social interactions in the datasets, we note that the decrease in the performance is not as significant as for the Vanilla LSTM model. In particular, while both models perform approximately equivalent on datasets with low social interactions, the Social LSTM model clearly outperforms the Vanilla LSTM model on datasets with strong and far-ranging interactions. This difference also shows in the final displacement error of the two models. We can see that the FDE of the Social LSTM model is significantly lower than the FDE of the Vanilla LSTM model for datasets with strong interactions between the pedestrians.



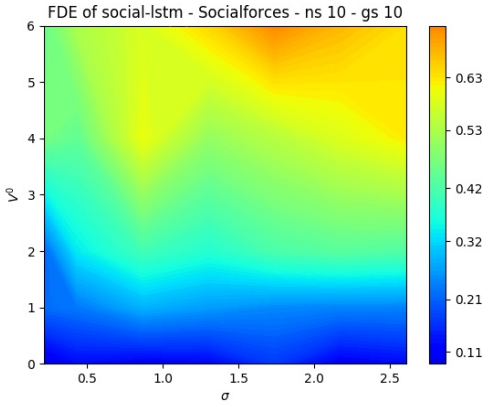
(a) ADE of Vanilla LSTM model



(b) ADE of Social LSTM model



(c) FDE of Vanilla LSTM model



(d) FDE of Social LSTM model

Figure 5.4.: Average and final displacement error of both models for 35 different synthetic datasets. Low errors are represented in blue, while high errors are depicted in red. The Social LSTM model clearly outperforms the Vanilla LSTM model on datasets with strong social interactions between the pedestrians.

A detailed overview of the particular values for the presented results can be found in Table A.1 and Table A.2 in the appendix.

Average nonlinear displacement error

We saw that with respect to the overall average and final displacement error, the Social LSTM model clearly outperforms the Vanilla LSTM model on datasets with strong social interactions. In the following, we want to gain more insights on how well these models predict social interactions. For this, we use the average nonlinear displacement error introduced in subsection 5.1.2. When pedestrians interact with each other, they adapt their motions

according to the behavior of others. Therefore, social interactions between pedestrians cause their respective trajectories to become nonlinear. The average nonlinear displacement error evaluates the error of a model in specific nonlinear regions of the trajectories. In order to identify these regions, a threshold value td is defined. The corresponding error is obtained by calculating the ADE for the identified regions, excluding the points of the trajectories for which the curvature is below the threshold.

We perform this evaluation on a dataset with strong and far-ranging social interactions. In particular, we define $V^0 = 6$, $\sigma = 2.6058$ and $N_{const} = 14$. In Figure 5.5, we illustrate the distribution of curvature values k_i, calculated for all future trajectories Y_i of the dataset. We observe that this distribution is exponentially decaying with rising curvature value k_i^t. While approximately 33% of all curvature values are close to zero, the number of values above $k = 1.6$ is almost negligible. However, there is a considerable number of points, for which the curvature of a trajectory lies in the interval $K = [0.1, 1.6]$. According to this distribution, we define a set of threshold values $td \in \{0.0, 0.1, \ldots, 1.5, 1.6\}$ and calculate the average nonlinear displacement error for each value of this set.

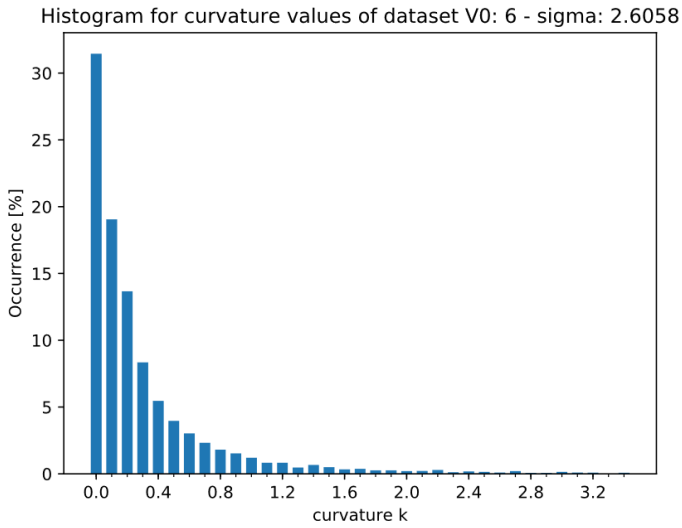


Figure 5.5.: Distribution of calculated curvature values k for the trajectories in the analyzed dataset.

Note that with rising value of td, we exclude an increasing number of points of the trajectories at which the curvature is below the threshold $k_i^t < td$. The respective results for both models are illustrated in Figure 5.6. We can see that the average nonlinear displacement error of both models increases with rising values of td. This aligns with our expectation, since we neglect trivial linear movements and focus on complex motions as we increase the threshold value td. Although the performance of both models decreases with rising values of td, the Social LSTM model clearly outperforms the Vanilla LSTM model for all threshold values. In particular, we observe that even for highly nonlinear regions of a trajectory, i.e.

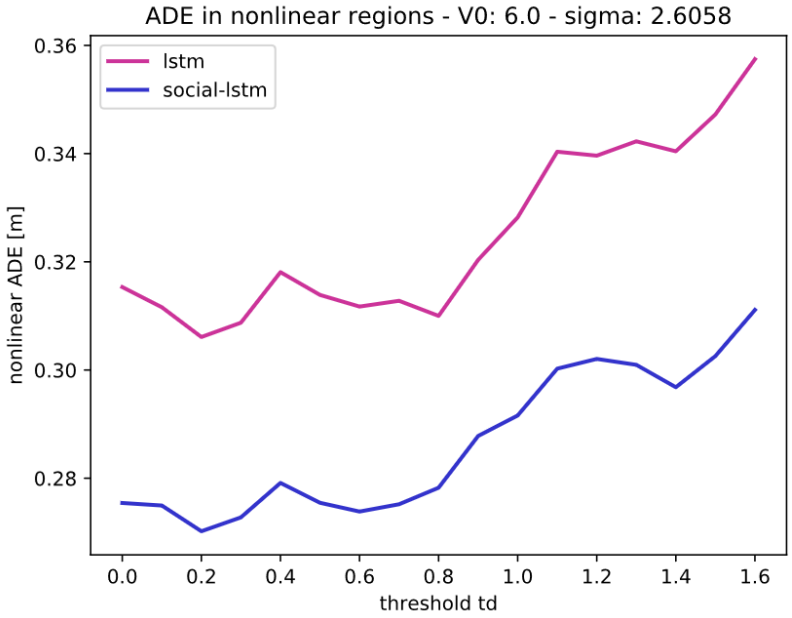


Figure 5.6.: Average nonlinear displacement error of the Vanilla LSTM model and the Social LSTM model for different threshold values td. The Social LSTM model consistently outperforms the Vanilla LSTM model for all threshold values.

$td > 1.0$, the error of the Social LSTM model is below the overall ADE of the Vanilla LSTM model, which averages to 0.33 meters for this dataset (see Table A.1). This shows that the Social LSTM model performs comparatively well even on complex motion behavior that is highly influenced by social interactions.

Despite these results, analyzing only isolated regions of a trajectory can sometimes lead to misleading results, as a model might predict trajectories that fit the respective regions accurately, but fail to approximate the remaining positions. Hence, we additionally evaluate the performance of the models on entire trajectories that are classified with respect to their degree of nonlinearity.

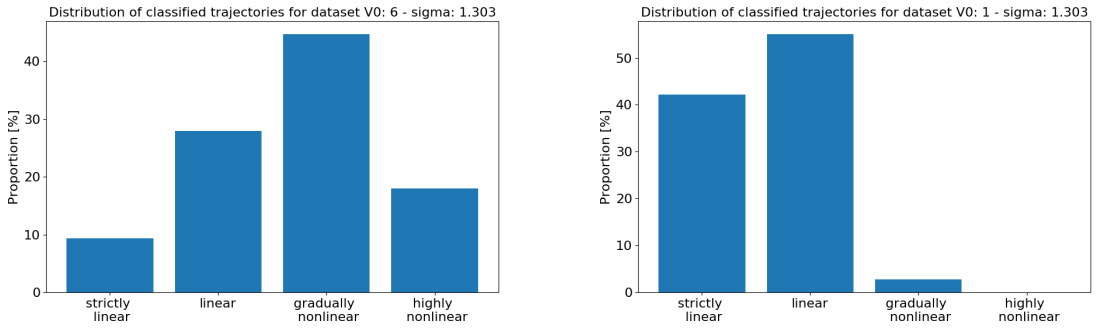
ADE and FDE on classified trajectories

We classify the trajectories of each dataset generated according to the classification scheme described in subsection 5.1.3. This allows us to compare the performances of the models on classes of trajectories that are differently influenced by social interactions. In this regard, we analyze the following two datasets:

1. **Dataset A**. High density of pedestrians in each scene with $N_{const} = 20$. The social interactions between the pedestrians are characterized by a strong magnitude $V^0 = 6$ and a moderate range $\sigma = 1.303$.

2. **Dataset B**. Comparatively low density of pedestrians with $N_{const} = 8$. The social interactions between the pedestrians are characterized by a weak magnitude $V^0 = 1$ and a moderate range $\sigma = 1.303$.

We note that while dataset A is characterized by strong social interactions and a crowded scene configuration, dataset B comprises weak social interactions and a low number of pedestrians for each scene. Hence, we expect dataset A to contain a high number of gradually and highly nonlinear trajectories. In comparison, we expect dataset B to comprise a high number of linear trajectories. Figure 5.7 shows the distribution of successfully classified trajectories for both datasets. As expected, we observe that most classified trajectories of dataset A are gradually nonlinear, while the amount of strictly linear trajectories is low. In contrast, for dataset B, we observe a high number of linear trajectories, while the amount of gradually nonlinear trajectories is almost negligible. In particular, there are no highly nonlinear trajectories in this dataset.



(a) Dataset A

(b) Dataset B

Figure 5.7.: Distribution of successfully classified trajectories for dataset A and dataset B. The bars represent the amount of *strictly linear*, *linear*, *gradually* and *highly nonlinear* trajectories in the datasets. Note that we neglect the *Other*-class, since it does not represent an interpretable trajectory-type.

The evaluation of the average and final displacement error of the Vanilla LSTM model and the Social LSTM model for all future trajectories Y of dataset A is depicted in Table 5.3. It is apparent that the Social LSTM model clearly outperforms the Vanilla LSTM model. This corresponds to the previous results for datasets with strong social interactions.

Table 5.3.: Average and final displacement error of the Vanilla LSTM model and the Social LSTM model for dataset A. The Social LSTM model clearly outperforms the Vanilla LSTM model.

Metric	**LSTM**	**S-LSTM**
ADE	0.53	**0.40**
FDE	1.13	**0.84**

In order to analyze the prediction behavior of these models in more detail, we evaluate their performance on each previously defined trajectory-class. Figure 5.8 illustrates the average displacement error of both models for the classified trajectories of dataset A. We observe that while the models perform approximately equivalent on linear trajectories, the Social LSTM model clearly outperforms the Vanilla LSTM model on gradually and highly nonlinear trajectories. We can see that the ADE of the Vanilla LSTM model is almost one third higher for gradually trajectories than for strictly linear trajectories. In comparison, the performance of the Social LSTM model is equivalent for both classes. The Vanilla LSTM model performs particularly poorly on highly nonlinear trajectories. Here, the ADE of the model is almost two thirds higher than the respective error on strictly linear trajectories. The average displacement error of the Social LSTM model only increases by 19% respectively.

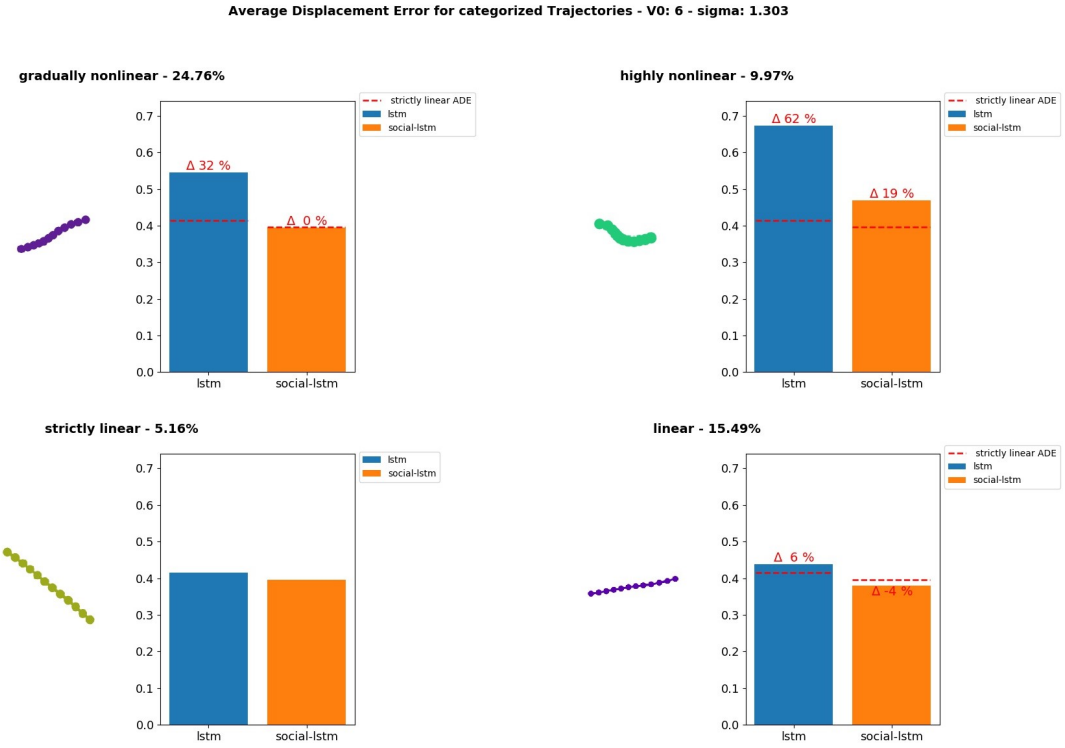


Figure 5.8.: ADE of the Vanilla LSTM model and the Social LSTM model on the classified trajectories of dataset A. The red values at the top of the bars indicate the relative increase or decrease of the ADE with respect to the error on strictly linear trajectories. The Social LSTM model especially outperforms the Vanilla LSTM model on nonlinear trajectories.

According to the results of dataset A, Table 5.4 shows the performance of the models on all future trajectories Y of dataset B. We observe that the Vanilla LSTM model slightly outperforms the Social LSTM model. However, both models perform approximately equivalent on this dataset. This corresponds to our previous results on datasets with few social interactions between pedestrians.

Table 5.4.: Average and final displacement error of the Vanilla LSTM model and the Social LSTM model for dataset B.

Metric	LSTM	S-LSTM
ADE	**0.09**	0.11
FDE	**0.18**	0.22

The overall results of both models are reflected in their performances on the classified trajectories. As shown in Figure 5.9, the Vanilla LSTM model slightly outperforms the Social LSTM model for all classes. However, while the Vanilla LSTM model performs particularly well on strictly linear trajectories, the difference in the performances between the two models is less for gradually nonlinear trajectories. In particular, while the ADE of the Vanilla LSTM model increases by more than 70% for gradually nonlinear trajectories compared to the error for strictly linear trajectories, the respective error of the Social LSTM model only increases by 46%.

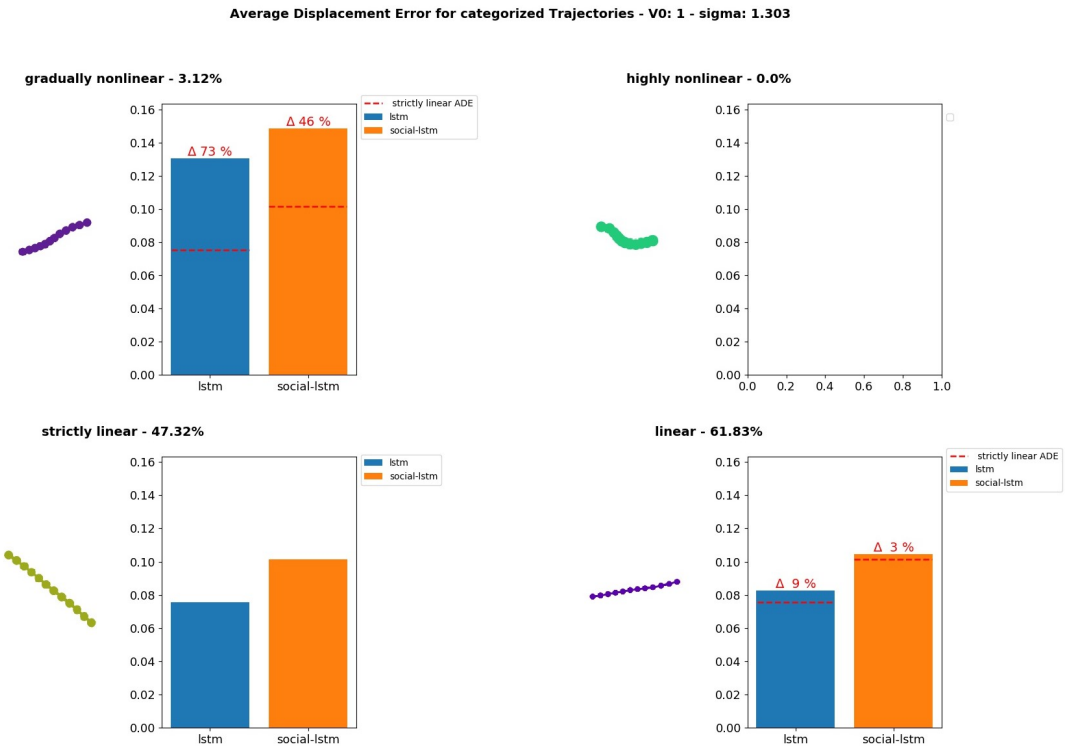


Figure 5.9.: ADE of both models on the classified trajectories of dataset B. The Vanilla LSTM model outperforms the Social LSTM model on all trajectory-classes. However, especially on gradually nonlinear trajectories, the two models perform almost equivalent.

For both datasets, we observe similar results for the FDE on the respective trajectory-classes.

A detailed overview of the respective results can be found in Figure A.4 and Figure A.5 in the appendix. In summary, we see that the Social LSTM model predicts gradually and highly nonlinear trajectories that are influenced by social interactions significantly better than the Vanilla LSTM model. This shows especially for datasets with strong social interactions and reflects in the models' overall performance on these datasets. Both models perform similarly on linear trajectories that are only slightly or not at all influenced by social interactions. However, for datasets with almost no social interactions, we observe that the Vanilla LSTM model slightly outperforms the Social LSTM model. A possible explanation is that for these datasets, the more complex Social LSTM model is slightly overfitting.

Collision behavior

Pedestrians rarely collide with each other, as they normally respect personal space and keep a certain distance to strangers. Models that are capable of learning social interactions should be able to predict trajectories that reproduce this avoidance behavior. As mentioned in subsection 5.1.4, we evaluate the collision behavior of the models by analyzing the distribution of distances between pedestrians. For this, we define a collision to have occurred if the distance between two pedestrians is less than $r_{coll} = 1.0$ meters.

In this manner, we analyze two crowded datasets with strong social interactions that differ only in the range of the respective forces. The specific parameters of these datasets can be found in Table 5.5. Note that we expect more collisions for dataset A than for dataset C, as the value of σ is lower and the corresponding social interactions are not as far-ranging.

Table 5.5.: Overview of the parameters of dataset A and dataset C.

Dataset	N_{const}	V^0	σ
Dataset A	20	6	1.303
Dataset C	20	6	2.6058

We compare the results of both models and the ground truth data in Table 5.6. The given values represent the amount of collisions between pedestrians during the prediction process in the respective datasets. We observe that for both datasets, the Social LSTM model predicts collision behavior that is close to that of the ground truth data. Contrarily, we see that the Vanilla LSTM model predicts almost twice as many collisions as actually given in the datasets.

Table 5.6.: Amount of collisions between the pedestrians during the prediction process. While the Social LSTM model predicts a frequency of collisions that is close to the ground truth data, the Vanilla LSTM model predicts collisions more frequently than they occur in the actual data.

	LSTM	**S-LSTM**	**Ground Truth**
Dataset A	12.3%	7.9%	6.8%
Dataset C	3.0%	1.6%	1.5%

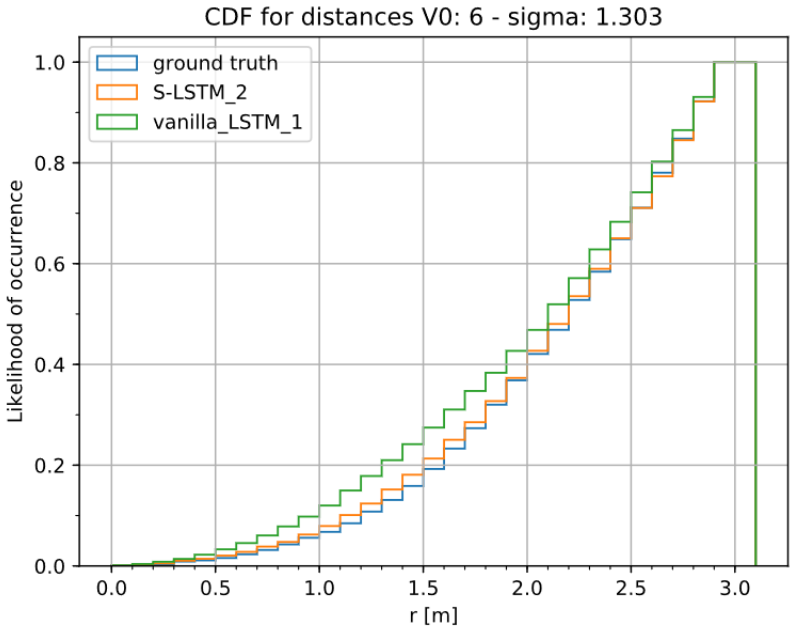


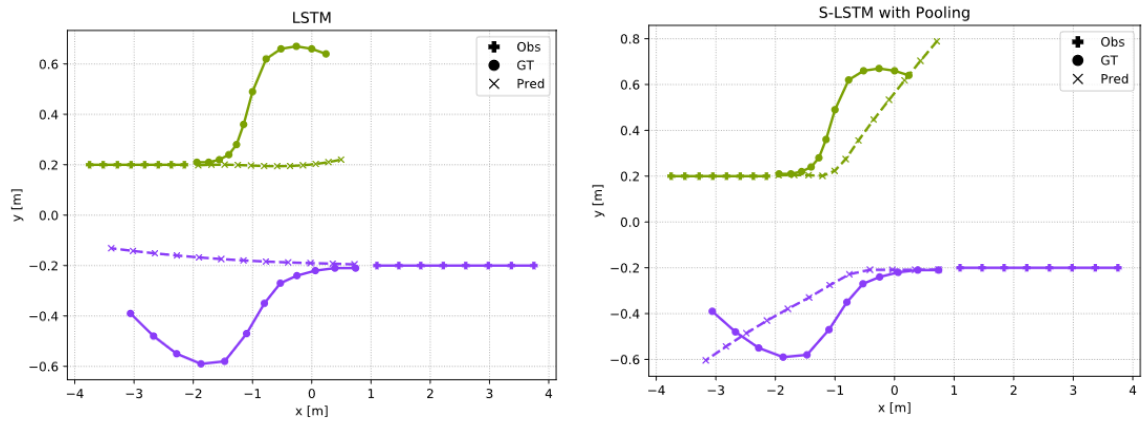
Figure 5.10.: The figure illustrates the cumulative distribution function for the euclidean distance between the pedestrians in dataset A for the ground truth data and both models. The CDF of the Vanilla LSTM model clearly deviates from the CDF of the ground truth data.

For a more detailed analysis, we plot the Cumulative Distribution Function (CDF) for the distances between the pedestrians of dataset A in Figure 5.10. Note that in order to focus on pedestrians that are influenced by social interactions, we exclusively consider the interval $r = [0, 3]$. The respective distributions of the Vanilla LSTM model and the Social LSTM model reflect the results in Table 5.6. We observe that the CDF of the Vanilla LSTM model clearly deviates from the distribution of the ground truth data. In particular, we can see that the pedestrians predicted by the model are in general closer to each other, compared to the ground truth data. In comparison, the CDF of the Social LSTM model almost completely fits the distribution of the actual data.

5.3. Qualitative Evaluation

The quantitative evaluation in section 5.2 shows that the Social LSTM model outperforms the Vanilla LSTM model on motion behavior that is significantly influenced by the social interactions between pedestrians. In this section, we aim to gain more insight on the actual prediction behavior of these models. Therefore, we qualitatively compare the predictions of the models on social scenes where pedestrians interact with each other.

In order to qualitatively evaluate the collision avoidance capacity of the models, we present an example scenario in Figure 5.11, in which two pedestrians pass each other and avoid



(a) Vanilla LSTM model

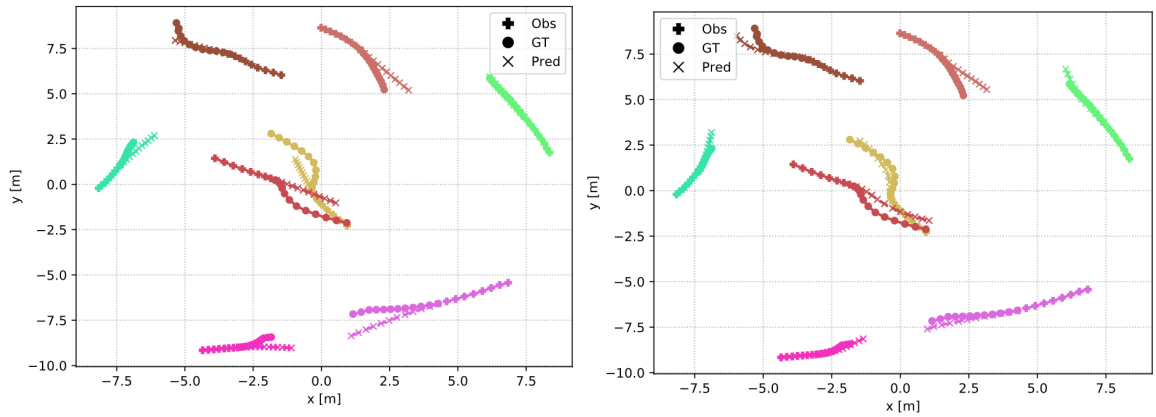
(b) Social LSTM model

Figure 5.11.: Example scenario of two pedestrians passing each other and altering their paths in order to avoid a close contact. The predicted trajectories are indicated by stars while the true trajectories are denoted by circles. The predictions are visualized for (a) the Vanilla LSTM model and (b) the Social LSTM model.

close contact by altering their paths respectively. As it can be seen, the Vanilla LSTM model is not able to reproduce the avoidance-behavior of the pedestrians. Instead, it predicts two predominantly linear trajectories that are not influenced by the motion of neighboring individuals. As the model does not take the motion behavior of neighboring pedestrians into account, it fails to predict the actual interactions between the pedestrians. In comparison, we see that the Social LSTM model predicts two trajectories that clearly avoid a possible collision. The model considers the motion of neighboring pedestrians and replicates the corresponding avoidance-behavior. These observations match the results of the quantitative evaluation of the models' collision behavior.

In Figure 5.12, we illustrate the prediction results of the Vanilla LSTM model and the Social LSTM model for an example scenario of a dataset generated with strong social interactions. In particular, we define $V^0 = 6$, $\sigma = 2.171$ and $N_{const} = 8$. In the centre of the scene we observe two pedestrians strongly interacting with each other. Similar to the prediction behavior depicted in Figure 5.11, we see that the Vanilla LSTM does not consider the impact of neighboring pedestrians in the scene. Therefore, it mainly predicts linear trajectories, which in the case of the two pedestrians in the centre would lead to a collision. Contrarily, we observe that the Social LSTM model predicts a motion behavior that considers the movements of neighboring individuals. For the two pedestrians in the centre, it predicts trajectories that reflect the social interactions between the pedestrians and avoid a possible collision.

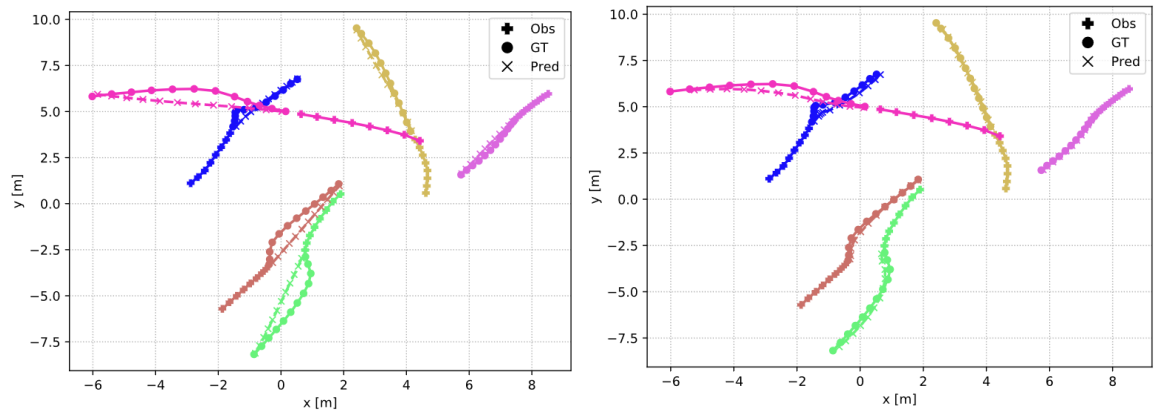
Both models often fail to predict accurate trajectories because they lack information about the destination of the pedestrians. As a consequence, the models seem to fail in predicting suitable trajectories, although they consider social interactions between pedestrians during the prediction process. In order to focus on the performance of these models on social interactions, we analyze how the prediction behavior changes when we provide the models



(a) Vanilla LSTM model (b) Social LSTM model

Figure 5.12.: Comparison between the predicted trajectories of the Vanilla LSTM (a) and the Social LSTM model (b) for a dataset with strong social interactions between the pedestrians. The Social LSTM model predicts trajectories that reflect social interactions and avoid collisions.

with information about the final position $\vec{y}_i^{T_{pred}}$ of each pedestrian i. An example of the respective results for a dataset with strong and far-ranging social interactions ($V^0 = 6$; $\sigma = 2.6058$) is illustrated in Figure 5.13. We can see that both models accurately predict the final location of each pedestrian. However, similar to previous results, we observe that the Vanilla LSTM model mainly predicts linear trajectories towards the final position of each pedestrian and fails to model social interactions between individuals. In comparison, the Social LSTM model is able to accurately predict the observed motions.



(a) Vanilla LSTM model (b) Social LSTM model

Figure 5.13.: Predictions of the Vanilla LSTM model (a) and the Social LSTM model (b) for a dataset with strong and far-ranging social interactions between pedestrians. Both models are provided with information about the destination of each pedestrian in the scene. While the Vanilla LSTM model neglects social interactions between pedestrians, the Social LSTM model accurately predicts these interactions

6. Conclusion and outlook

In this chapter, we summarize the main results and discuss the key findings of this work. Furthermore, we give an outlook on possible future studies. This thesis presents a method to create customized datasets that can be used to evaluate specific properties of human motion. In this manner, we generated datasets for which we defined the impact of social interactions on the motion of pedestrians. By analyzing the performances of the Vanilla LSTM model and the Social LSTM model on these datasets, we were able to evaluate their ability to predict social interactions. We showed that while both models perform similarly on datasets with weak social interactions, the Social LSTM model clearly outperforms the Vanilla LSTM model on datasets with strong social interactions. Furthermore, we saw that the Social LSTM model performs significantly better than the Vanilla LSTM model on trajectories that are gradually or highly influenced by social interactions. This becomes particularly apparent for datasets in which social interactions have a strong impact on the motion of pedestrians. Additionally, we analyzed the collision behavior within the datasets and compared it with the predicted collision behavior of both models. We saw that while the Vanilla LSTM model predicts collisions between pedestrians much more frequently than they occur in the actual data, the Social LSTM model is able to reproduce the collision behavior of the dataset.

A qualitative evaluation of the trajectories predicted by both models supports these results. In particular, we saw that the Vanilla LSTM model fails to predict the avoidance behavior of two pedestrians passing each other. In comparison, we showed that the Social LSTM model is able to predict trajectories that avoid such a collision. Furthermore, we saw that while the Vanilla LSTM model predominantly predicts linear trajectories, the Social LSTM model predicts nonlinear trajectories that consider the impact of neighboring pedestrians. Given these results, we conclude that the Social LSTM model is capable of predicting social interactions between pedestrians. In comparison, we saw that the Vanilla LSTM model is not sensitive to neighboring individuals and, thus, not able to predict these interactions.

In summary, this thesis presented a way to evaluate trajectory prediction models with respect to their capability of predicting social interactions between pedestrians. For future studies, we suggest evaluating additional trajectory prediction models on the datasets generated, using the evaluation metrics proposed in this thesis and the presented results as baselines. In particular, it might be interesting to investigate how other models that exploit social interactions perform compared to the Social LSTM model. Once several models are compared in this manner, the results could be used in order to identify the architectural requirements of a model that is able to precisely predict social interactions. Furthermore, we suggest using the methods introduced to generate hand-tailored datasets that simulate human-space interactions in order to evaluate and compare trajectory prediction models that aim to predict interactions between pedestrians and obstacles.

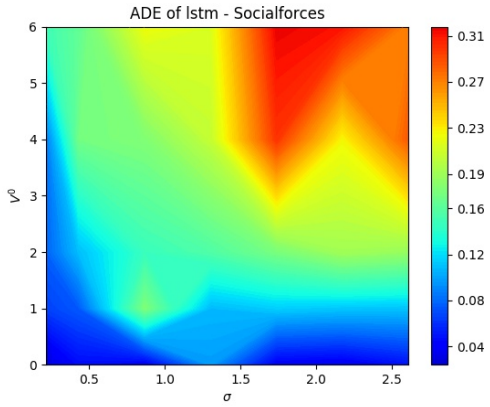
A. Appendix

Vanilla LSTM model							
V^0	σ						
	0.2171	0.4343	0.8686	1.303	1.7371	2.171	2.6058
6	**0.17** / **0.33**	0.26 / 0.53	0.35 / 0.74	0.36 / 0.78	0.42 / 0.94	0.38 / 0.85	0.33 / 0.74
4	**0.16** / **0.31**	**0.19** / **0.37**	0.31 / 0.66	0.28 / 0.62	0.31 / 0.70	0.33 / 0.72	0.32 / 0.71
2	**0.08** / **0.15**	**0.15** / **0.30**	0.21 / 0.42	0.22 / 0.46	0.22 / 0.47	0.25 / 0.53	0.23 / 0.49
1	**0.08** / **0.16**	0.11 / **0.21**	**0.13** / **0.25**	0.13 / 0.27	0.13 / 0.29	0.13 / 0.28	0.14 / 0.29
0	0.04 / 0.11	**0.03** / **0.07**	**0.03** / **0.06**	**0.04** / **0.09**	**0.05** / **0.12**	**0.03** / **0.06**	**0.03** / **0.07**

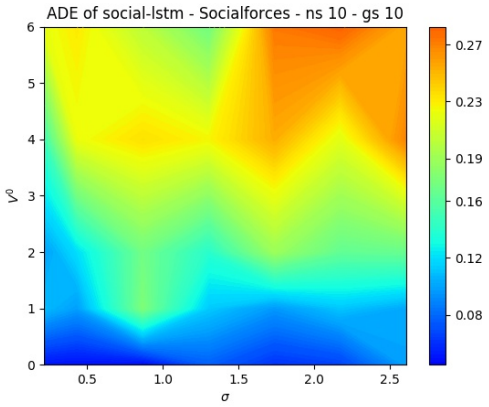
Table A.1.: Values for the ADE and FDE of the Vanilla LSTM model on 35 different datasets, generated with a constant number of pedestrians in each scene of $N_{const} = 14$. The values of the ADE and FDE are seperated by a slash, where the first value indicates the ADE. All values represent meters. The performance of the Vanilla LSTM model is compared with the performance of the Social LSTM model in Table A.2.

Social LSTM model							
V^0	σ						
	0.2171	0.4343	0.8686	1.303	1.7371	2.171	2.6058
6	0.22 / 0.45	0.26 / 0.53	**0.28** / **0.58**	**0.29** / **0.65**	**0.34** / **0.73**	**0.30** / **0.68**	**0.29** / **0.64**
4	0.23 / 0.47	0.22 / 0.44	**0.28** / **0.60**	**0.23** / **0.50**	**0.25** / **0.54**	**0.27** / **0.59**	**0.28** / **0.62**
2	0.10 / 0.21	0.16 / 0.34	**0.20** / **0.41**	**0.18** / **0.37**	**0.20** / **0.42**	**0.21** / **0.44**	**0.20** / **0.43**
1	0.11 / 0.22	0.11 / 0.23	0.14 / 0.29	0.13 / **0.26**	**0.12** / **0.25**	**0.11** / **0.24**	**0.11** / **0.24**
0	0.04 / **0.09**	0.05 / 0.13	0.05 / 0.11	0.05 / 0.12	0.08 / 0.17	0.05 / 0.11	0.05 / 0.13

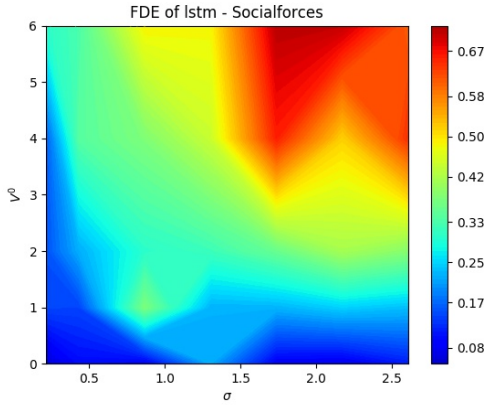
Table A.2.: Values for the ADE and FDE of the Social LSTM model on 35 different datasets, generated with a constant number of pedestrians in each scene of $N_{const} = 14$. The values of the ADE and FDE are seperated by a slash, where the first value indicates the ADE. All values represent meters. The performance of the Social LSTM model is compared with the performance of the Vanilla LSTM model in Table A.1.



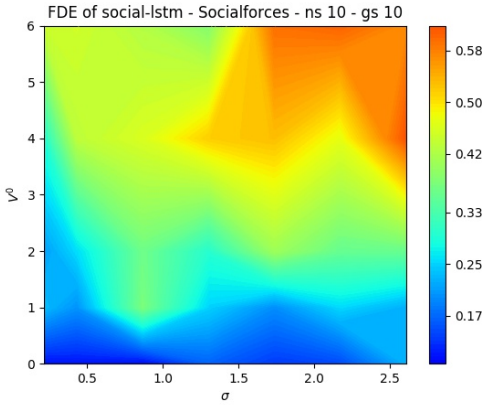
(a) ADE of Vanilla LSTM model



(b) ADE of Social LSTM model



(c) FDE of Vanilla LSTM model



(d) FDE of Social LSTM model

Figure A.1.: Average and final displacement error of both models for 35 different synthetic datasets with $N_{const} = 8$. Low errors are represented in blue, while high errors are depicted in red. The Social LSTM model outperforms the Vanilla LSTM model on datasets with strong social interactions between the pedestrians. The specific values can be found in Table A.3 and Table A.4. Note that the difference in performance is not as significant as for the datasets with a high density of pedestrians with $N_{const} = 14$ that comprise many social interactions between pedestrians.

Vanilla LSTM model							
V^0	σ						
	0.2171	0.4343	0.8686	1.303	1.7371	2.171	2.6058
6	**0.15** / **0.30**	**0.16** / **0.33**	0.22 / 0.49	0.22 / 0.49	0.32 / 0.71	0.31 / 0.70	0.27 / 0.61
4	**0.08** / **0.16**	**0.17** / **0.34**	**0.18** / **0.38**	**0.21** / **0.44**	0.30 / 0.66	0.23 / 0.51	0.28 / 0.64
2	**0.08** / **0.15**	**0.11** / **0.22**	**0.15** / **0.30**	0.16 / 0.32	**0.17** / **0.36**	0.19 / 0.41	0.19 / 0.38
1	**0.07** / **0.15**	**0.07** / **0.14**	0.18 / 0.37	0.11 / 0.23	0.11 / 0.23	0.11 / 0.25	0.11 / 0.23
0	**0.03** / **0.05**	0.05 / **0.10**	**0.04** / **0.08**	0.10 / 0.22	**0.04** / **0.08**	**0.04** / **0.08**	**0.05** / **0.11**

Table A.3.: Values for the ADE and FDE of the Vanilla LSTM model on 35 different datasets, generated with a comparatively low constant number of pedestrians in each scene of $N_{const} = 8$. The values of the ADE and FDE are seperated by a slash, where the first value indicates the ADE. All values represent meters. The performance of the Vanilla LSTM model is compared with the performance of the Social LSTM model in Table A.4.

Social LSTM model							
V^0	σ						
	0.2171	0.4343	0.8686	1.303	1.7371	2.171	2.6058
6	0.22 / 0.44	0.23 / 0.45	**0.20** / **0.42**	**0.17** / **0.38**	**0.27** / **0.59**	**0.28** / **0.60**	**0.25** / **0.57**
4	0.15 / 0.29	0.22 / 0.43	0.24 / 0.46	0.23 / 0.52	**0.25** / **0.53**	**0.22** / **0.47**	**0.27** / **0.62**
2	0.10 / 0.21	0.12 / 0.26	0.17 / 0.36	**0.14** / **0.30**	0.19 / 0.41	**0.16** / **0.36**	**0.17** / **0.36**
1	0.11 / 0.23	0.10 / 0.20	0.18 / 0.37	0.11 / 0.25	**0.10** / **0.20**	0.11 / 0.25	**0.10** / **0.22**
0	0.05 / 0.11	0.05 / 0.11	0.05 / 0.10	**0.08** / **0.17**	0.06 / 0.14	0.07 / 0.15	0.10 / 0.23

Table A.4.: Values for the ADE and FDE of the Social LSTM model on 35 different datasets, generated with a comparatively low constant number of pedestrians in each scene of $N_{const} = 8$. The values of the ADE and FDE are seperated by a slash, where the first value indicates the ADE. All values represent meters. The performance of the Social LSTM model is compared with the performance of the Vanilla LSTM model in Table A.3.

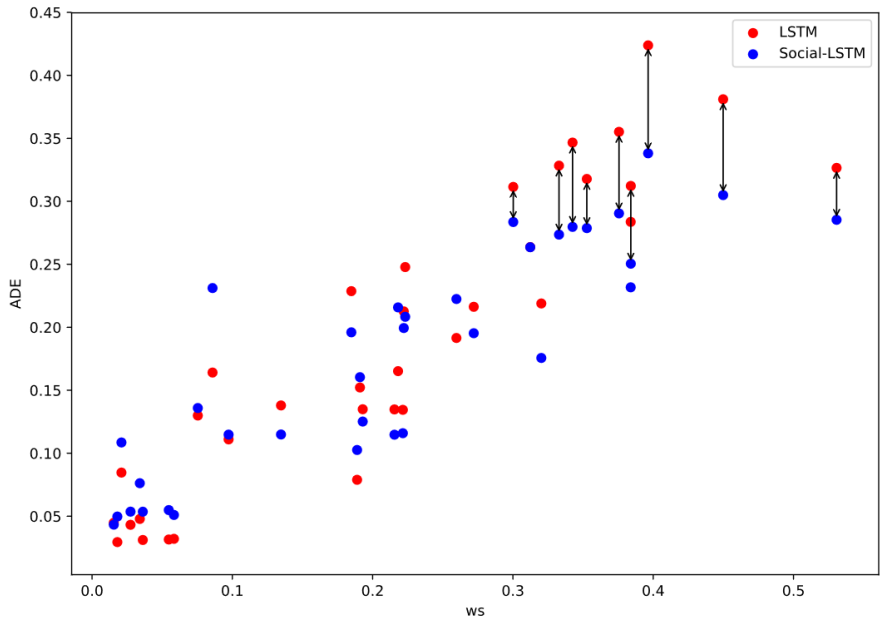


Figure A.2.: Average displacement error of the Vanilla LSTM model and the Social LSTM model for increasing values of *ws*. The arrows indicate the increasing difference in the performance of the two models for high values of *ws*.

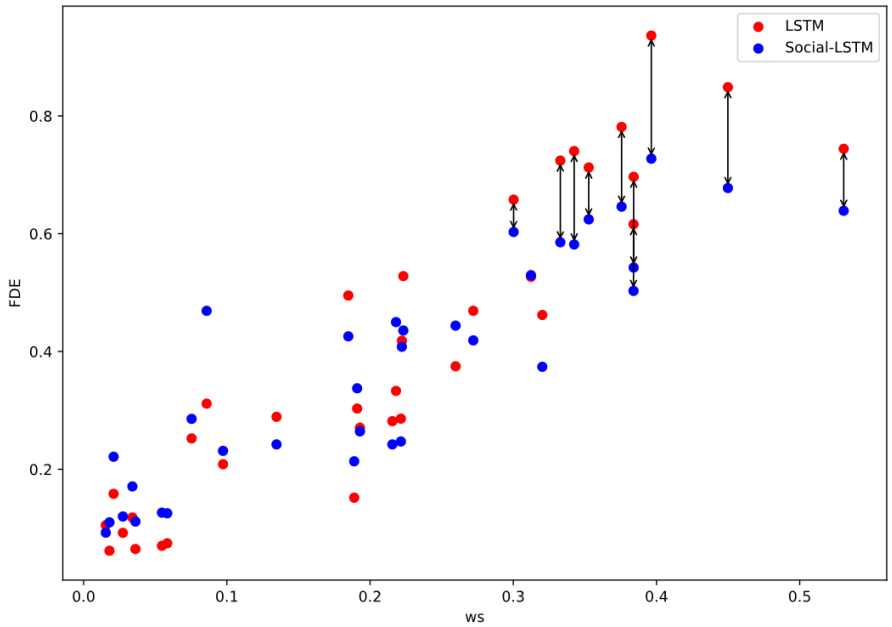


Figure A.3.: Final displacement error of the Vanilla LSTM model and the Social LSTM model for increasing values of *ws*. The arrows indicate the increasing difference in the performance of the two models for high values of *ws*.

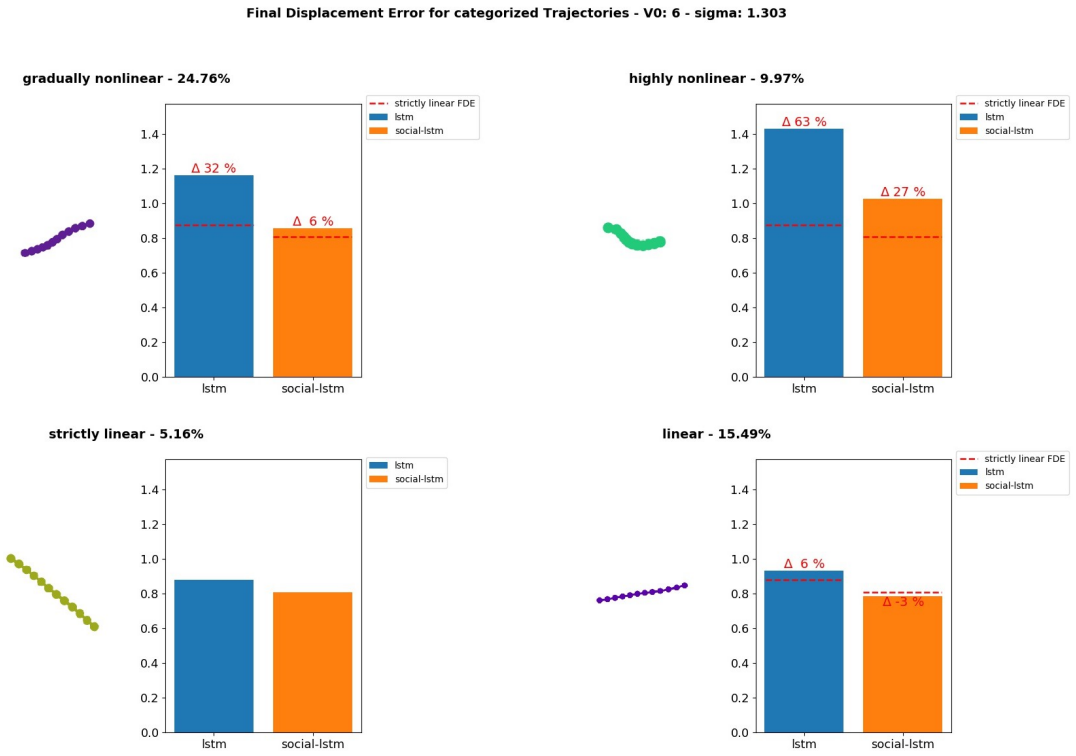


Figure A.4.: FDE of the Vanilla LSTM model and the Social LSTM model on the classified trajectories of dataset A. The red values at the top of the bars indicate the relative increase or decrease of the FDE with respect to the error on strictly linear trajectories.

Trajectory-class	Vanilla LSTM model	Social LSTM model
Overall	0.53 / 1.13	**0.40** / **0.84**
Strictly linear	0.41 / 0.88	**0.40** / **0.81**
Linear	0.44 / 0.93	**0.38** / **0.78**
Gradually nonlinear	0.55 / 1.16	**0.40** / **0.86**
Highly nonlinear	0.67 / 1.43	**0.47** / **1.03**
Other	0.53 / 1.11	**0.39** / **0.84**

Table A.5.: Average and final displacement error of the Vanilla LSTM model and the Social LSTM model on different trajectory-classes of dataset A. ADE and FDE are separated by a slash, where the first value denotes the ADE. All values represent meters.

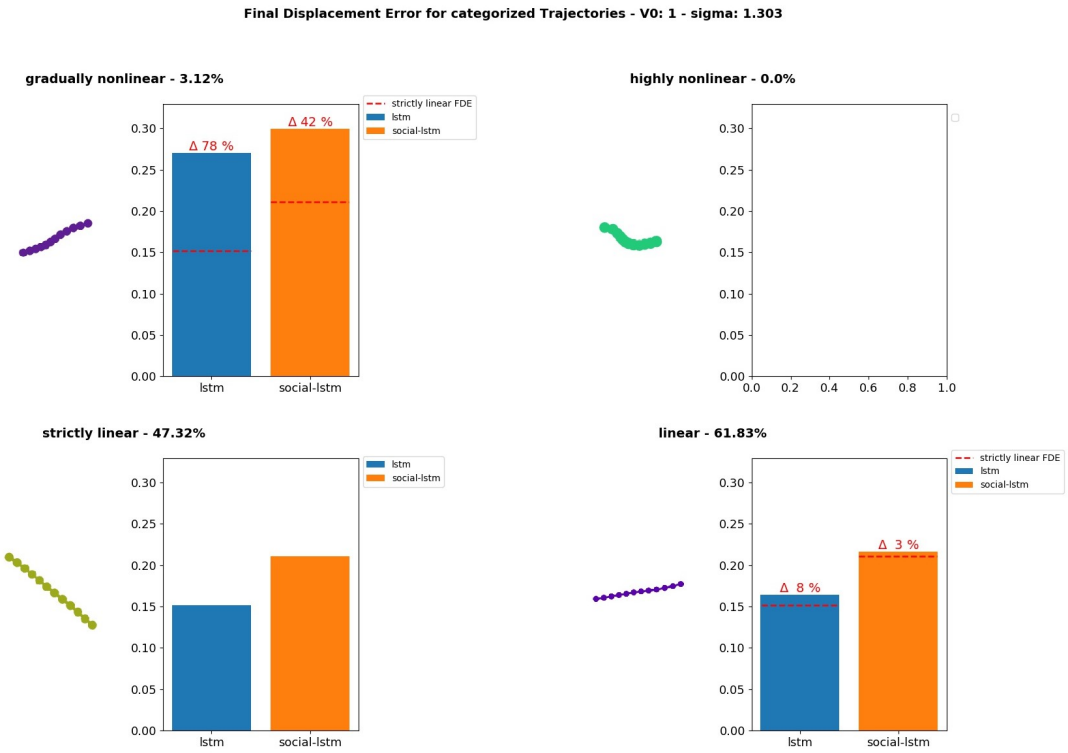


Figure A.5.: FDE of the Vanilla LSTM model and the Social LSTM model on the classified trajectories of dataset B. The red values at the top of the bars indicate the relative increase or decrease of the FDE with respect to the error on strictly linear trajectories.

Trajectory-class	Vanilla LSTM model	Social LSTM model
Overall	**0.09** / **0.18**	0.11 / 0.22
Strictly linear	**0.08** / **0.15**	0.10 / 0.21
Linear	**0.08** / **0.16**	0.10 / 0.22
Gradually nonlinear	**0.13** / **0.27**	0.15 / 0.30
Highly nonlinear	-	-
Other	0.10 / **0.20**	0.10 / 0.22

Table A.6.: Average and final displacement error of the Vanilla LSTM model and the Social LSTM model on different trajectory-classes of dataset B. ADE and FDE are separated by a slash, where the first value denotes the ADE. All values represent meters.

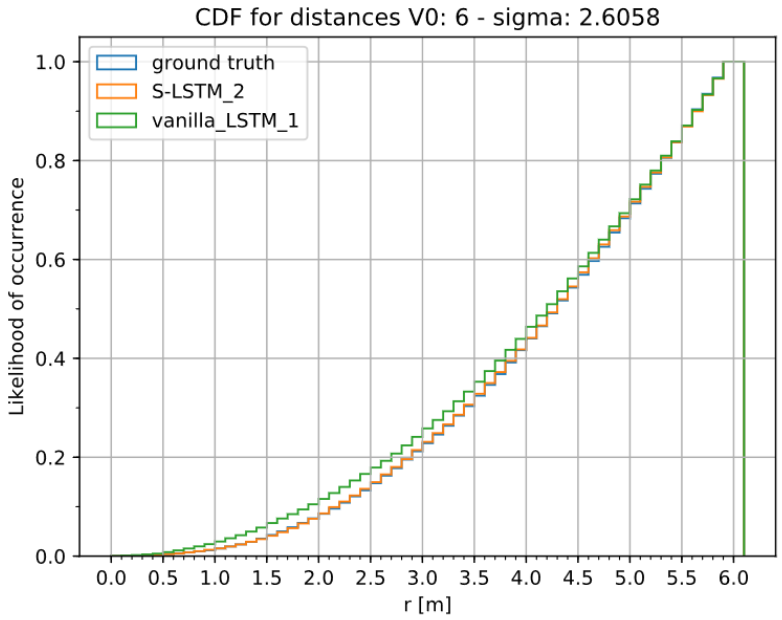


Figure A.6.: Cumulative distribution function of the euclidean distance between the pedestrians in dataset C for the ground truth data and both models. The CDF of the Vanilla LSTM model deviates from the distribution of the ground truth data especially for low distances. The CDF of the Social LSTM model accurately fits the distribution of the actual data.

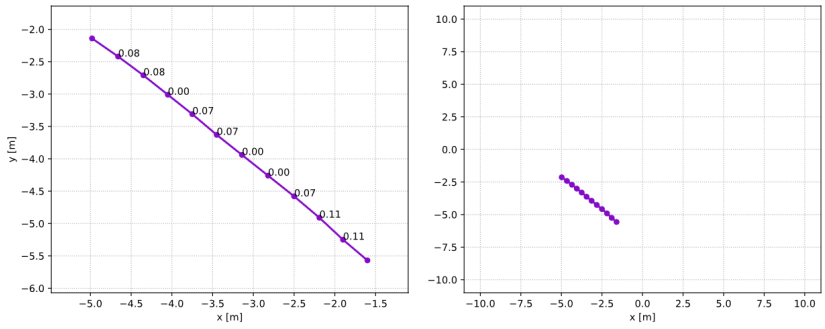


Figure A.7.: Example of a strictly linear trajectory. The values depicted next to the points of the trajectory denote the respective curvature values at these positions.

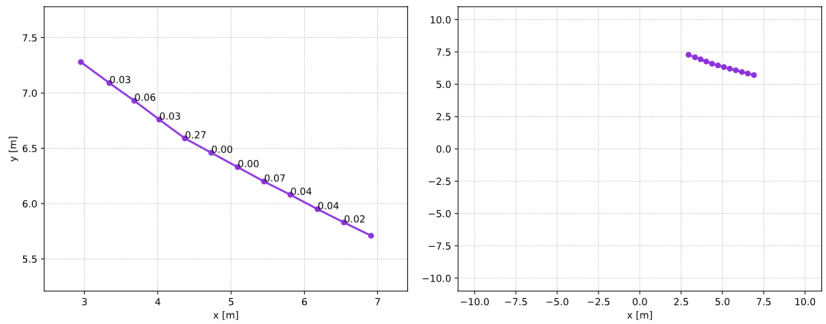


Figure A.8.: Example of a linear trajectory. The values depicted next to the points of the trajectory denote the respective curvature values at these positions.

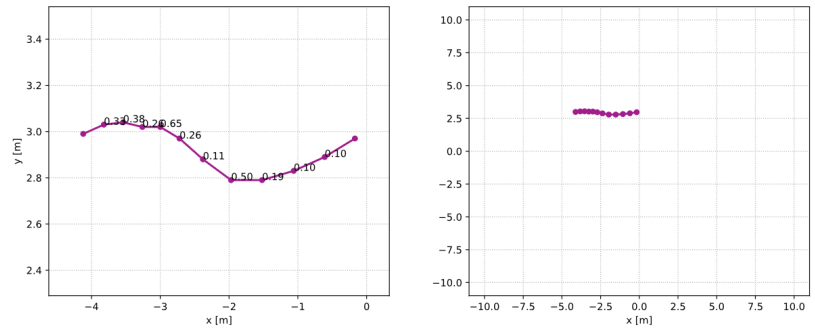


Figure A.9.: Example of a gradually nonlinear trajectory. The values depicted next to the points of the trajectory denote the respective curvature values at these positions.

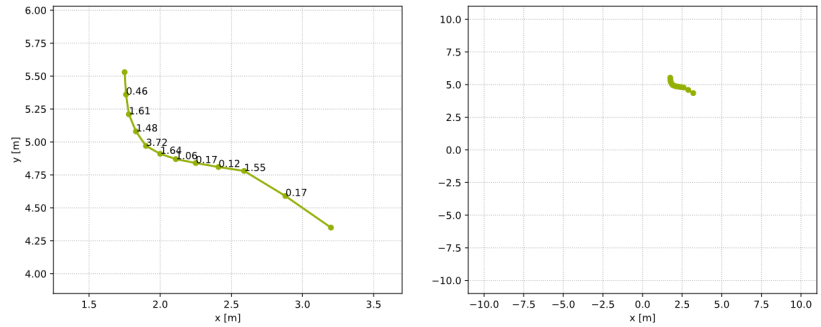
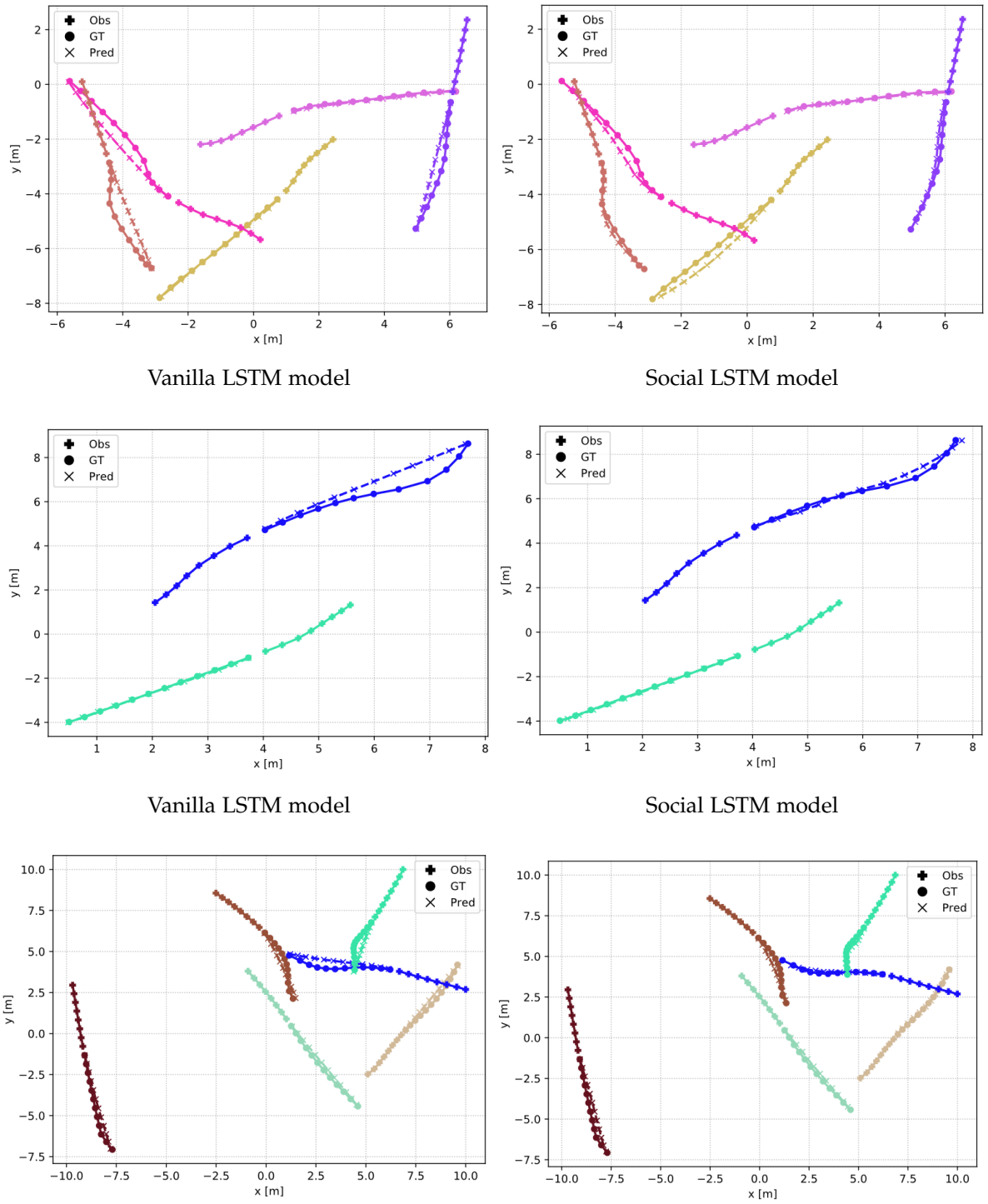


Figure A.10.: Example of a highly nonlinear trajectory. The values depicted next to the points of the trajectory denote the respective curvature values at these positions.



Vanilla LSTM model | Social LSTM model

Vanilla LSTM model | Social LSTM model

Vanilla LSTM model | Social LSTM model

Figure A.10.: Predictions of the Vanilla LSTM model (left) and the Social LSTM model (right) for a dataset with strong and far-ranging social interactions between pedestrians. Both models are provided with information about the destination of each pedestrian in the scene.

List of Figures

List of Tables

Bibliography

[1] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese. "Social LSTM: Human Trajectory Prediction in Crowded Space". In: *CVPR* (2016), pp. 961–971.

[2] J. Amirian, J.-B. Hayet, and J. Pettre. *Social Ways: Learning Multi-Modal Distributions of Pedestrian Trajectories with GANs*. 2019. arXiv: 1904.09507 [cs.CV].

[3] G. Antonini, M. Bierlaire, and M. Weber. "Discrete Choice Models for Pedestrian Walking Behavior". In: *Transportation Research Part B: Methodological* 40 (Sept. 2006), pp. 667–687. DOI: 10.1016/j.trb.2005.09.006.

[4] D. Bahdanau, K. Cho, and Y. Bengio. *Neural Machine Translation by Jointly Learning to Align and Translate*. 2014. arXiv: 1409.0473 [cs.CL].

[5] M. A. Bedau, J. S. McCaskill, N. H. Packard, and S. Rasmussen. "Cellular Automata Model Of Emergent Collective Bi-Directional Pedestrian Dynamics". In: *Artificial Life VII: Proceedings of the Seventh International Conference on Artificial Life*. 2000, pp. 437–445.

[6] A. G. Belyaev. "A Note on Invariant Three-Point Curvature Approximations (Singularity theory and Differential equations)". In: 1111.5 (1999). URL: https://repository.kulib.kyoto-u.ac.jp/dspace/bitstream/2433/63349/1/1111-0.pdf.

[7] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. *InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets*. 2016. arXiv: 1606.03657 [cs.LG].

[8] J. Chorowski, D. Bahdanau, K. Cho, and Y. Bengio. "End-to-end continuous speech recognition using attention-based recurrent nn: First results". English (US). In: *NIPS 2014 Workshop on Deep Learning, December 2014*. 2014.

[9] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling*. 2014. arXiv: 1412.3555 [cs.NE].

[10] N. Fridman, A. Zilka, and G. A. Kaminka. *The impact of cultural differences on crowd dynamics in pedestrian and evacuation domains*. Tech. rep. MAVERICK 2011/01. Bar Ilan University, Computer Science Department, 2011, pp. 18–23.

[11] L. Gao, Z. Guo, H. Zhang, X. Xu, and H. T. Shen. "Video Captioning With Attention-Based LSTM and Semantic Consistency". In: *IEEE Transactions on Multimedia* 19.9 (2017), pp. 2045–2055.

[12] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. *Generative Adversarial Networks*. 2014. arXiv: 1406.2661 [stat.ML].

[13] A. Graves. *Generating Sequences With Recurrent Neural Networks*. 2013. arXiv: 1308.0850 [cs.NE].

[14] A. Graves and N. Jaitly. "Towards End-To-End Speech Recognition with Recurrent Neural Networks". In: *Proceedings of the 31st International Conference on Machine Learning*. Ed. by E. P. Xing and T. Jebara. Vol. 32. Proceedings of Machine Learning Research 2. Bejing, China: PMLR, 2014, pp. 1764–1772. URL: http://proceedings.mlr.press/v32/graves14.html.

[15] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi. "Social GAN: Socially Acceptable Trajectories with Generative Adversarial Networks." In: *CoRR* abs/1803.10892 (2018). URL: http://dblp.uni-trier.de/db/journals/corr/corr1803.html#abs-1803-10892.

[16] N. C. Heglund, G. A. Cavagna, and C. R. Taylor. "Energetics and mechanics of terrestrial locomotion. III. Energy changes of the centre of mass as a function of speed and body size in birds and mammals". In: *Journal of Experimental Biology* 97.1 (1982), pp. 41–56. ISSN: 0022-0949. eprint: https://jeb.biologists.org/content/97/1/41.full.pdf. URL: https://jeb.biologists.org/content/97/1/41.

[17] D. Helbing. *A Fluid Dynamic Model for the Movement of Pedestrians*. 1998. arXiv: cond-mat/9805213 [cond-mat.stat-mech].

[18] D. Helbing. "Physikalische Modellierung des dynamischen Verhaltens von Fußgängern (Physical Modeling of the Dynamic Behavior of Pedestrians)". Diplomarbeit. ETH Zürich, 1990. URL: https://ssrn.com/abstract=2413177%20or%20http://dx.doi.org/10.2139/ssrn.2413177.

[19] D. Helbing and P. Monlar. "Social Force Model for Pedestrian Dynamics". In: *Physical Review* 51.5 (1995).

[20] L. Henderson. "On the fluid mechanics of human crowd motion". In: *Transportation Research* 8.6 (1974), pp. 509–515. ISSN: 0041-1647. DOI: https://doi.org/10.1016/0041-1647(74)90027-6. URL: http://www.sciencedirect.com/science/article/pii/0041164774900276.

[21] S. Hochreiter and J. Schmidhuber. "Long Short-Term Memory". In: *Neural Comput.* 9.8 (1997), pp. 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. URL: https://doi.org/10.1162/neco.1997.9.8.1735.

[22] R. L. Hughes. "A continuum theory for the flow of pedestrians". In: *Transportation Research Part B: Methodological* 36.6 (2002), pp. 507–535. ISSN: 0191-2615. DOI: https://doi.org/10.1016/S0191-2615(01)00015-7. URL: http://www.sciencedirect.com/science/article/pii/S0191261501000157.

[23] R. L. Hughes. "THE FLOW OF HUMAN CROWDS". In: *Annual Review of Fluid Mechanics* 35.1 (2003), pp. 169–182. DOI: 10.1146/annurev.fluid.35.101101.161136. eprint: https://doi.org/10.1146/annurev.fluid.35.101101.161136. URL: https://doi.org/10.1146/annurev.fluid.35.101101.161136.

[24] D. P. Kingma and J. Ba. *Adam: A Method for Stochastic Optimization*. 2014. arXiv: 1412.6980 [cs.LG].

[25] D. P. Kingma and M. Welling. *Auto-Encoding Variational Bayes*. 2013. arXiv: 1312.6114 [stat.ML].

[26] V. Kosaraju, A. Sadeghian, R. Martín-Martín, I. D. Reid, H. Rezatofighi, and S. Savarese. "Social-BiGAT: Multimodal Trajectory Forecasting using Bicycle-GAN and Graph Attention Networks." In: *NeurIPS*. Ed. by H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. B. Fox, and R. Garnett. 2019, pp. 137–146. URL: http://dblp.uni-trier.de/db/conf/nips/nips2019.html#KosarajuSM0RS19.

[27] V. Kosaraju, A. Sadeghian, R. Martín-Martín, I. Reid, S. H. Rezatofighi, and S. Savarese. *Social-BiGAT: Multimodal Trajectory Forecasting using Bicycle-GAN and Graph Attention Networks*. 2019. arXiv: 1907.03395 [cs.CV].

[28] L. Leal-Taixé, G. Pons-Moll, and B. Rosenhahn. "Everybody needs somebody: Modeling social and grouping behavior on a linear programming multiple people tracker". In: *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*. 2011, pp. 120–127.

[29] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. S. Torr, and M. Chandraker. "DESIRE: Distant Future Prediction in Dynamic Scenes with Interacting Agents". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017). DOI: 10.1109/cvpr.2017.233. URL: http://dx.doi.org/10.1109/CVPR.2017.233.

[30] A. Lerner, Y. Chrysanthou, and D. Lischinski. "Crowds by Example". In: *Comput. Graph. Forum* 26 (Sept. 2007), pp. 655–664. DOI: 10.1111/j.1467-8659.2007.01089.x.

[31] J. Li, F. Yang, M. Tomizuka, and C. Choi. *EvolveGraph: Heterogeneous Multi-Agent Multi-Modal Trajectory Prediction with Evolving Interaction Graphs*. 2020. arXiv: 2003.13924 [cs.CV].

[32] M. Lisotto, P. Coscia, and L. Ballan. *Social and Scene-Aware Trajectory Prediction in Crowded Spaces*. 2019. arXiv: 1909.08840 [cs.CV].

[33] M. Luber, J. A. Stork, G. D. Tipaldi, and K. O. Arras. "People tracking with human motion predictions from social forces". In: *2010 IEEE International Conference on Robotics and Automation*. 2010, pp. 464–469.

[34] B. Majecka. "Statistical models of pedestrian behaviour in the Forum". MA thesis. University of Edinburgh, 2009.

[35] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu. *Recurrent Models of Visual Attention*. 2014. arXiv: 1406.6247 [cs.LG].

[36] B. T. Morris and M. M. Trivedi. "Trajectory Learning for Activity Understanding: Unsupervised, Multilevel, and Long-Term Adaptive Approach". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.11 (2011), pp. 2287–2301.

[37] C. Niemitz. "The evolution of the upright posture and gait—a review and a new synthesis". In: *Naturwissenschaften* 97 (2010), pp. 241–263. URL: https://doi.org/10.1007/s00114-009-0637-3.

[38] 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*. 2009, pp. 261–268.

[39] R. Raghavendra, A. Del Bue, M. Cristani, and V. Murino. "Abnormal Crowd Behavior Detection by Social Force Optimization". In: *Human Behavior Understanding*. Ed. by A. A. Salah and B. Lepri. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 134–145.

[40] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese. "Learning Social Etiquette: Human Trajectory Understanding In Crowded Scenes". In: *Computer Vision – ECCV 2016*. Ed. by B. Leibe, J. Matas, N. Sebe, and M. Welling. Cham: Springer International Publishing, 2016, pp. 549–565.

[41] A. Sadeghian, V. Kosaraju, A. Sadeghian, N. Hirose, H. Rezatofighi, and S. Savarese. "SoPhie: An Attentive GAN for Predicting Paths Compliant to Social and Physical Constraints." In: *CVPR*. Computer Vision Foundation / IEEE, 2019, pp. 1349–1358. URL: http://dblp.uni-trier.de/db/conf/cvpr/cvpr2019.html#SadeghianKSHRS19.

[42] A. Sadeghian, V. Kosaraju, A. Sadeghian, N. Hirose, S. H. Rezatofighi, and S. Savarese. *SoPhie: An Attentive GAN for Predicting Paths Compliant to Social and Physical Constraints*. 2018. arXiv: 1806.01482 [cs.CV].

[43] A. Sadeghian, F. Legros, M. Voisin, R. Vesel, A. Alahi, and S. Savarese. *CAR-Net: Clairvoyant Attentive Recurrent Network*. 2017. arXiv: 1711.10061 [cs.CV].

[44] H. Soltau, H. Liao, and H. Sak. "Neural Speech Recognizer: Acoustic-to-Word LSTM Model for Large Vocabulary Speech Recognition". In: *Interspeech 2017* (2017). DOI: 10.21437/interspeech.2017-1566. URL: http://dx.doi.org/10.21437/Interspeech.2017-1566.

[45] I. Sutskever, O. Vinyals, and Q. V. Le. *Sequence to Sequence Learning with Neural Networks*. 2014. arXiv: 1409.3215 [cs.CL].

[46] A. Treuille, S. Cooper, and Z. Popović. "Continuum Crowds". In: *ACM Trans. Graph.* 25.3 (July 2006), pp. 1160–1168. ISSN: 0730-0301. DOI: 10.1145/1141911.1142008. URL: https://doi.org/10.1145/1141911.1142008.

[47] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio. *Graph Attention Networks*. 2017. arXiv: 1710.10903 [stat.ML].

[48] S. Venugopalan, L. A. Hendricks, R. Mooney, and K. Saenko. "Improving LSTM-based Video Description with Linguistic Knowledge Mined from Text". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing* (2016). DOI: 10.18653/v1/d16-1204. URL: http://dx.doi.org/10.18653/v1/D16-1204.

[49] C. Wang, H. Yang, C. Bartz, and C. Meinel. "Image Captioning with Deep Bidirectional LSTMs". In: *Proceedings of the 24th ACM International Conference on Multimedia*. MM '16. Amsterdam, The Netherlands: Association for Computing Machinery, 2016, pp. 988–997. ISBN: 9781450336031. DOI: 10.1145/2964284.2964299. URL: https://doi.org/10.1145/2964284.2964299.

[50] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio. *Show, Attend and Tell: Neural Image Caption Generation with Visual Attention*. 2015. arXiv: 1502.03044 [cs.LG].

[51] K. Yamaguchi, A. C. Berg, L. E. Ortiz, and T. L. Berg. "Who are you with and where are you going?" In: *CVPR 2011*. 2011, pp. 1345–1352.

[52] B. Zhou, X. Wang, and X. Tang. "Understanding Collective Crowd Behaviors:Learning a Mixture Model of Dynamic Pedestrian-Agents". In: June 2012. DOI: 10.1109/CVPR.2012.6248013.

[53] J.-Y. Zhu, R. Zhang, D. Pathak, T. Darrell, A. A. Efros, O. Wang, and E. Shechtman. *Toward Multimodal Image-to-Image Translation*. 2017. arXiv: 1711.11586 [cs.CV].