Human Trajectory Forecasting in Crowds: A Deep Learning Perspective

Parth Kothari, Sven Kreiss, Alexandre Alahi
Visual Intelligence for Transportation (VITA) Lab, EPFL

Abstract.

Since the past few decades, human trajectory forecasting has been a field of active research owing to its numerous real-world applications: evacuation situation analysis, traffic operations, deployment of social robots in crowded environments, to name a few. In this work, we cast the problem of human trajectory forecasting as learning a representation of human social interactions. Early works handcrafted this representation based on domain knowledge. However, social interactions in crowded environments are not only diverse but often subtle. Recently, deep learning methods have outperformed their handcrafted counterparts, as they learned about human-human interactions in a more generic data-driven fashion. In this work, we present an in-depth analysis of existing deep learning based methods for modelling social interactions. Based on our analysis, we propose a simple yet powerful method for effectively capturing these social interactions. To objectively compare the performance of these interaction-based forecasting models, we develop a large scale interaction-centric benchmark *TrajNet++*, a significant yet missing component in the field of human trajectory forecasting. We propose novel performance metrics that evaluate the ability of a model to output socially acceptable trajectories. Experiments on TrajNet++ validate the need for our proposed metrics, and our method outperforms competitive baselines on both real-world and synthetic datasets.

1 Introduction

Humans possess the natural ability to navigate in social environments. In other words, we have understood the social etiquette of human motion from respecting personal space and yielding right-of-way to avoid walking through people belonging to the same group. Our social interactions lead to various complex pattern-formation phenomena in crowds, for instance, the emergence of lanes of pedestrians with uniform walking direction, oscillations of the pedestrian flow at bottlenecks. The ability to model social interactions and thereby forecast crowd dynamics in real world environments is extremely valuable for a wide range of applications: infrastructure design [1, 2, 3], traffic operations [4], crowd abnormality detection systems [5], evacuation situation analysis [6, 7, 8, 9], deployment of autonomous vehicles [10, 11], deployment of social robots in pedestrian-only environments [12] and recently helping in the broad quest of building a digital twin of our built environment. However, modelling social interactions is an extremely challenging task as there exists no fixed set of rules which govern human motion. A task closely related to learning human social interactions is forecasting the

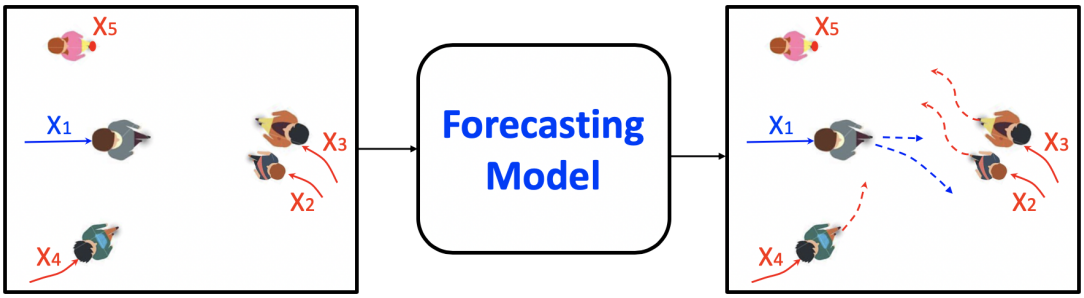


Figure 1: Human trajectory forecasting refers to the task of forecasting the future trajectories of all humans which conform to the social norms, given the past observed scene. It is the presence of social interactions that distinguish human trajectory forecasting from other sequence modelling tasks: the primary pedestrian (X1) deviates from his direction of motion to avoid a collision, by forecasting the trajectory of the child (X2) in front of him. Moreover, one needs to have the ability to measure the performance of the model with respect to predicting socially acceptable outputs: instead of deviating, the model can predict that the primary pedestrian slows down. This prediction can lead to high errors in distance-based metrics, even though it is socially acceptable.

movement of the surrounding people, which conform to common sense unspoken rules. We refer to this task of forecasting the human motion as *human trajectory forecasting.*

Before formally defining human trajectory forecasting, we introduce the notion of *Trajectory* and *Scene*. We define a *Trajectory* as the time-profile of pedestrian motion states. Generally, these states are the position and velocity of a human. However, we can consider more complex states like body pose, to glean more information about a person's movement. We define a *Scene* as a collection of trajectories of multiple humans interacting in a social setting. A scene may also comprise physical objects and non-navigable areas that affect the human trajectories, *e.g.,* walls, doors and elevators. We define human trajectory forecasting as follows:

Given the past trajectories of all humans in a scene, forecast the future trajectories which conform to the social norms.

Human trajectory forecasting is primarily a sequence modelling task. The typical challenges for a sequence modelling task are (1) encoding observation sequence: we need to learn to model the long-term dependency in the past trajectory effectively, (2) multimodality: given the history of a scene, multiple futures (predictions) are plausible. In addition to this, for human trajectory forecasting, there exist two crucial challenges that differentiate it from other sequence prediction tasks such as language modelling, weather forecasting, and stock market forecasting (see Fig 1):

- Presence of social interactions: the trajectory of a person is affected by the motion of the other people in his/her surroundings. Modelling how the observation of one sequence affects the forecast of another sequence is an essential prerequisite for a good human trajectory forecasting model.
- Physically acceptable outputs: a good human trajectory forecasting model should provide physically acceptable outputs, for instance, the model prediction should not undergo collisions. Quantifying the physical feasibility of a model prediction is crucial for safety-critical applications.

In this work, we cast the problem of human trajectory forecasting as learning a representation of human social interactions. In other words, our objective is to encode the observed scene into a representation that captures all information necessary to forecast human motion. For instance, a

representation can be the history of the velocities for each human. Such a representation considers human motion in isolation. However, as mentioned above, an ideal representation has to encode not only the past motion but also the social interactions that a human undergoes with the surrounding humans. We term this representation as **Social Representation.** To ensure that this representation encodes only the social interactions, we assume that there do not exist any physical constraints in our scenes. Moreover, the representation can also be affected by the long-term goal of the human, which cannot always be observed or inferred. To find a representation that captures dominantly the social interactions, our focus in this work is on *short-term* human trajectory forecasting (next 5 seconds).

Early works [13, 14, 15] handcrafted the social representation of the scene based on the domain knowledge of human motion. Social Force [13], one of the first seminal work on human trajectory forecasting, defined attractive forces (towards the goal of a person and towards his/her group) and repulsive forces (away from people not belonging to a person's group and physical obstacles) to forecast human motion. Antonini *et al.* [14] utilized the discrete choice framework for modelling pedestrian dynamics in a crowd by proposing a dynamic and individual-based discretization of space around the pedestrian. Pellegrini *et al.* [15] defined an energy functional to capture interactions between two people based on their distance of closest approach (assuming constant-velocity movement). However, such handcrafted representations cannot capture all the diverse, higher-order, and often subtle interactions of human motion. To overcome this limitation, Alahi *et al.* [16] proposed the first neural network based model 'Social LSTM', paving the way for new deep learning methods for human trajectory forecasting. Neural networks (NNs) are powerful function approximators that can learn useful representations given large amounts of real data, without any prior assumptions.

Following the success of Social LSTM, a variety of NN-based interaction modules have been proposed in literature to model the social interactions. In this work, we explicitly focus on the design of these interaction modules and not the entire forecasting model. The challenge in designing these modules lies in handling a variable number of neighbours and modelling how they collectively influence one's future trajectory. We present a broad umbrella encompassing the existing designs of interaction modules based on the encoding architecture and the process by which the individual information of each neighbour is aggregated. Based on our taxonomy, we propose a simple yet novel module that improves the social interaction due to its ability to preserve the uniqueness of the neighbours and model higher-order interactions in the temporal domain.

To demonstrate the efficacy of a trajectory forecasting model, one needs to have the means to objectively compare with other forecasting baselines on good quality datasets. However, current methods have been evaluated on different subsets of available data without proper sampling of scenes in which social interactions occur. As our final contribution, we introduce TrajNet++, a large scale interaction-centric trajectory forecasting benchmark comprising explicit agent-agent scenarios. Our benchmark provides proper indexing of trajectories by defining a hierarchy of trajectory categorization. In addition, we provide an extensive evaluation system to test the gathered methods for a fair comparison. In our evaluation, we go beyond the standard distance-based metrics and introduce novel metrics that measure the capability of a model to emulate pedestrian behavior in crowds. Finally, we demonstrate the efficacy of our proposed baseline on TrajNet++, in comparison to existing works. We rely on the spirit of crowdsourcing and encourage researchers to submit their models to our benchmark, so the quality of trajectory forecasting models can keep increasing in tackling more challenging scenarios.

To summarize, our main contributions are as follows:

1. We provide an in-depth analysis of existing designs of interaction encoders along with their source code.

2. We propose a simple yet novel method for capturing social interactions, preserving the unique identity of surrounding pedestrians and providing an improved representation with time.

3. We present a large scale interaction-centric trajectory forecasting benchmark with novel evaluation metrics that quantify the *physical feasibility* of a model.

2 Related Work

Finding the ideal representation to encode human social interactions in crowded environments is an extremely challenging task. Social interactions are not only diverse but often subtle. In this work, we consider agent-based or microscopic models of pedestrian crowds, where collective phenomena emerge from the complex interactions between many individuals (self-organizing effects). Current human trajectory forecasting works can be categorized into learning human-human (social) interactions or human-space (physical) interactions or both. Our work is focused on deep learning based models that capture social interactions. In this section, we review the work done for modelling the agent-agent interactions to obtain the social representation.

Among the simplest models for representing human motion are the kinematic models such as constant velocity models and constant acceleration models. Mogelmose *et al.* [17] used a linear motion predictor to infer critical situations near roadside. Classical path prediction algorithms like Kalman filters [18], autoregressive models [19, 20] have also been explored to represent human motion.

With a specific focus on pedestrian path forecasting problem, Helbing and Molnar [13] presented a motion model with attractive forces (towards the goal) and repulsive forces (away from obstacles), called Social Force model, which captures the social and physical interactions. Their seminal work displays competitive results even on modern pedestrian datasets and has been extended for improved trajectory forecasting [21, 22, 23], tracking [24, 25, 26] and activity forecasting [27, 28]. Another prominent model for human motion is Reciprocal Velocity Obstacles (RVO) [29], which guarantees safe and oscillation-free motion, assuming that each agent follows identical collision avoidance reasoning. Social interaction modelling has been approached from different perspectives such as Discrete Choice framework [30], continuum dynamics [31] and Gaussian processes [32, 33]. Scholler *et al.* [34] proposed an effective constant velocity baseline for motion prediction. Robicquet *et al.* [35] defined social sensitivity to characterize human motion into different navigation styles. Alahi *et al.* [36] defined Social Affinity Maps to link broken or unobserved trajectories to forecast pedestrian destinations. Yi *et al.* [37] exploited crowd grouping as a cue to better forecast trajectories. However, all these methods use handcrafted functions based on relative distances and specific rules to model interactions. These functions impose not only strong priors but also have limited capacity when modelling complex interactions. In recent times, methods based on neural networks (NNs) that infer interactions in a data-driven fashion have been shown to outperform the works mentioned above.

Inspired by the application of recurrent neural networks (RNNs) in diverse sequence prediction tasks [38, 39, 40, 41], Alahi et al. [16] proposed Social LSTM, the first NN-based model for human trajectory forecasting. Social LSTM is an LSTM [42] network with a novel social pooling layer to capture social interactions of nearby pedestrians. RNNs incorporating social interactions allow anticipating interactions that can occur in a more distant future. The social pooling module has been extended to incorporate physical space context [43, 44, 45, 46, 47, 48] and various other designs of NN-based interaction module have been proposed [49, 50, 51, 52, 53, 54, 55, 56]. Pfieffer *et al.* [49] proposed an angular pooling grid for efficient computation. Shi *et al.* [50] proposed an elliptical pooling grid placed along the direction of movement of the pedestrian with more focus on the

pedestrians in the front. Bisagno *et al.* [51] proposed to consider only pedestrians not belonging to the same group during social pooling. Gupta *et al.* [52] propose to encode neighbourhood information through the use of a permutation-invariant (symmetric) max-pooling function. Zhang *et al.* [53] proposed to refine the state of the LSTM cell using message passing algorithms. Zhu *et al.* [54] proposed a novel star topology to model interactions. The center hub maintains information of the entire scene which each pedestrian can query. Ivanovic *et al.* [55] proposed to sum-pool the neighbour states and pass it through an LSTM-based encoder to obtain the interaction vector. Liang *et al.* [56] proposed to utilize geometric relations obtained from the spatial distance between pedestrians, to derive the interaction representation. Many works [57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67] propose interaction module designs based on attention mechanisms [68, 69] to identify the neighbours which affect the trajectory of the person of interest. The attention weights are either learned or handcrafted based on domain knowledge (*e.g.*, euclidean distance).

Few works augment the position or velocity of a person with additional social cues to represent human motion and interaction better. Hasan *et al.* [70] uses head direction along with velocity as input to their architecture. While modelling social interactions, the authors only consider the pedestrians in the visual frustum of attention [71]. Sun *et al.* [72] augments body direction with time and date. Ma *et al.* [73] extract visual attributes from the image, and utilize them to predict future trajectories using concept of fictitious play.

Different from the general notion of encoding motion using RNNs, Radwan *et al.* [74] proposed temporal convolutional neural networks (CNNs). Recently, Guiliari *et al.* [75] proposed Transformer architecture for the task of trajectory forecasting, but they do not take social interactions into account. For an extensive survey of all human forecasting methods capturing both social and physical interactions, one can refer to Rudenko *et al.* [76].

3 Method

A global data-driven pipeline for forecasting human motion is illustrated in Fig 2. It comprises of the motion encoding module, the interaction module and the decoder module. On a high level, the motion encoding module is responsible for encoding the past motion of pedestrians. The interaction module learns to capture the social interactions between pedestrians. The motion encoding module and the interaction module are not necessarily mutually exclusive. The output of the interaction module is the social representation of the scene. The social representation is passed to the decoder module to predict a single trajectory or a trajectory distribution depending on the decoder architecture. In this work, we focus on investigating the design choices for the interaction module.

3.1 Problem Statement

Our objective is to forecast the future trajectories of all the pedestrians present in a scene. The network takes as input the trajectories of all the people in a scene denoted by $\mathbf{X} = X_1, X_2, \ldots, X_n$ and our task is to forecast the corresponding future trajectories $\mathbf{Y} = Y_1, Y_2, \ldots, Y_n$. The position of pedestrian i at time-step t is denoted by $\mathbf{x}_i^t = (x_i^t, y_i^t)$. We receive the positions of all pedestrians at time-steps $t = 1, \ldots, T_{obs}$ and want to forecast the future (ground truth) positions $Y_i = (x_i^t, y_i^t)$ from time-steps $t = T_{obs+1}$ to T_{pred}. We denote our predictions using $\hat{\mathbf{Y}}$. The velocity of a pedestrian i at time-step t is denoted by $\mathbf{v}_i^t$. We denote the state of pedestrian i at time-step t by $\mathbf{s}_i^t$. The state can refer to different attributes of the person, *e.g.*, the position as well as velocity of the person ($\mathbf{s}_i^t = [\mathbf{x}_i^t, \mathbf{v}_i^t]$). The problem statement can be extended to take as input more attributes at each time-step, *e.g.*, the body pose, as well as predicting k most-likely future trajectories.

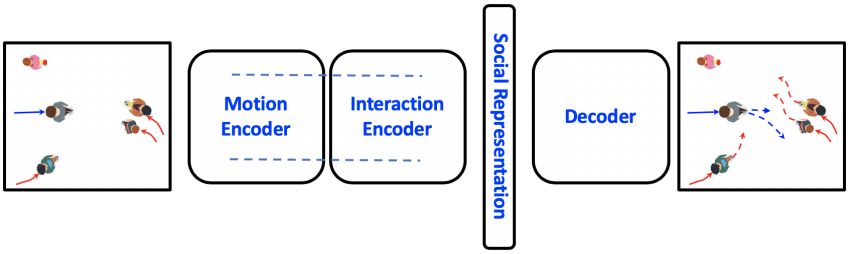


Figure 2: The global data-driven pipeline for human trajectory forecasting. The motion encoding module is responsible for encoding the past motion of pedestrians. The interaction module learns to capture the social interactions between pedestrians. The output of the interaction module is the social representation of the scene. The social representation is passed to the decoder module to forecast a single trajectory or a trajectory distribution.

3.2 Interaction Module

Humans have the capability to navigate with ease in complex, crowded environments by following unspoken social rules. The social interactions arising as a result of these unspoken rules are captured by designing novel interaction modules. We now discuss in detail the different components of data-driven interaction encoders. We present a broad overview of the interaction model designs proposed in the literature. The existing interaction models can be categorized into two categories, based on the input representation:

- Grid based
- Non-Grid based

3.2.1 Grid Based Interaction Models

In grid-based models [16, 49, 45, 46, 47, 51, 43, 48, 44], the interaction module takes as input a local grid constructed around the pedestrian of interest. Each cell within the grid represents a particular spatial position relative to the pedestrian of interest. Each cell contains information about neighbours located in that corresponding position. The information of the neighbours can be provided in various forms:
(1) **Occupancy Pooling** (Fig 3a): Each cell indicates the presence of a neighbour [16, 45]
(2) **Social Pooling** (Fig 3c): Each cell contains the entire past history of the neighbour, represented by, *e.g.*, the LSTM hidden state of the neighbours [16, 45, 47, 51, 43, 48, 44].
Grid-based modules provide the advantage of implicitly modelling the spatial context around the primary pedestrian. However, these methods can suffer from (1) loss of resolution arising from a defined size of each cell, and (2) the inability to model far-away pedestrians due to fixed grid size. These issues can be resolved by increasing the resolution and the grid size, but the solution significantly hampers the computational capability.

Mathematically, for occupancy pooling, at time-step t, we denote the neighbourhood of pedestrian i as O_i^t, which is a $N_o \times N_o$ matrix, where N_o is the size of the grid. The (m, n) element of this grid is

$$O_i^t(m,n) = \sum_{j \in N_i} \mathbb{1}_{mn}[x_j^t - x_i^t, y_j^t - y_i^t], \tag{1}$$

where $\mathbb{1}_{mn}[x, y]$ is an indicator function to check if (x, y) lies in the $(m, n)^{th}$ cell of the grid, and N_i

is the set of neighbors corresponding to person i. We denote this architecture by [**O-Grid**].

For social pooling, given h_i^t denotes the D-dimensional hidden-state of the LSTM of person i at time-step t, the 'social' tensor H_i^t of size $N_o \times N_o \times D$ is constructed as:

$$H_i^t(m, n, :) = \sum_{j \in N_i} \mathbb{1}_{mn}[x_j^t - x_i^t, y_j^t - y_i^t] h_j^{t-1}. \tag{2}$$

We denote this architecture by [**S-Grid**].

The resulting grid is embedded to get the interaction vector p_i^t (Fig 3d):

$$p_i^t = \phi(H_i^t; W_p), \tag{3}$$

where ϕ is an MLP and the weights W_p are learned. We would like to note that the input grid can also be represented in polar coordinates [49, 46].

Despite being more expressive than occupancy pooling, social pooling is known to suffer from high complexity, especially in cases of high-resolution grids. To reduce complexity, we propose **Directional Pooling** (see Fig 3b), wherein each cell comprises of the relative velocity of the neighbour with respect to the primary pedestrian. Let v_{ji}^t denote the relative velocity of person j with respect to person i, *i.e.*, $\mathbf{v}_{ji}^t = v_i^t - v_j^t$. The directional pooling tensor D_i^t is constructed as:

$$D_i^t(m, n, :) = \sum_{j \in N_i} \mathbb{1}_{mn}[x_j^t - x_i^t, y_j^t - y_i^t] \mathbf{v}_{ji}^t. \tag{4}$$

We denote this architecture by [**D-Grid**]. We will demonstrate in the experimental section that directional pooling, in addition to its computational advantages, performs at par with social pooling in controlled synthetic scenarios. Moreover, in real-world settings, directional pooling provides superior performance with respect to the physical acceptability of predicted trajectories.

3.2.2 Non-Grid Based Interaction Models

Non-grid based modules [58, 77, 57, 52, 50, 53, 54, 55, 62, 60], as the name suggests, capture the social interactions in a grid-free manner. The challenge in designing non-grid based models lies in (1) handling a variable number of neighbours and (2) aggregating the state information of multiple neighbours to obtain the interaction vector p_t^i. To achieve this, these models utilize the concepts of social attention [58, 57, 53, 62, 61, 59, 65, 60], or application of a learned symmetric function [52]. Fig 4 illustrates the different neighbouring information aggregation strategies.

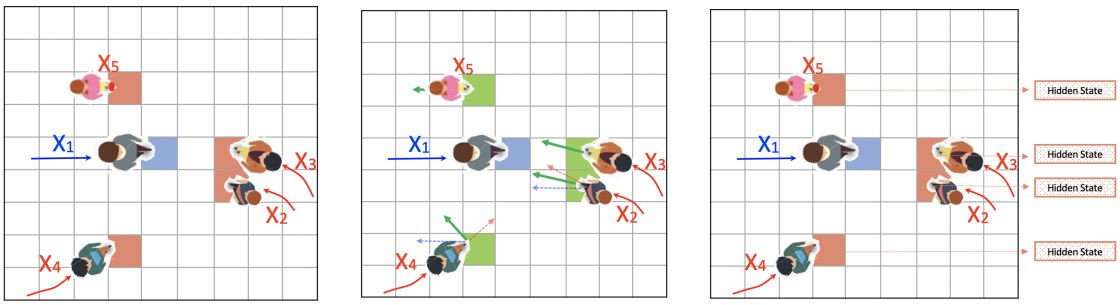
Recent works [58, 57, 53, 62, 59] propose to provide different weights to the neighbouring hidden-states to make the interaction vector p_i^t:

$$H_i^t = \sum_j a_{ij}^t * h_j^t, \tag{5}$$

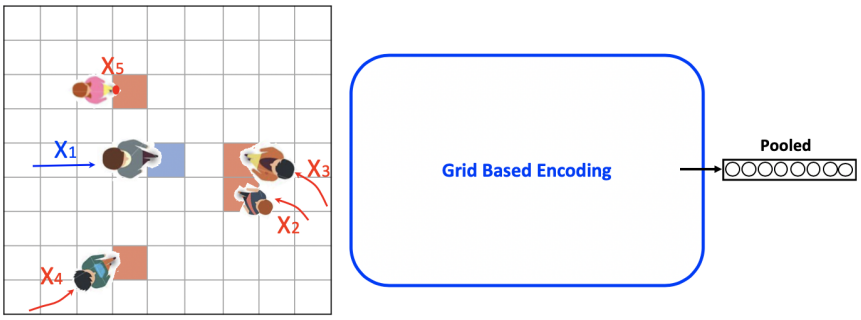
$$p_t^i = \phi(H_i^t; W_p), \tag{6}$$

where h_j^t denotes the hidden-state vector of pedestrian j, a_{ij}^t is the weight indicating the influence of pedestrian j on the trajectory of pedestrian i at time-step t. ϕ is an MLP and the weights W_p are learned. We denote this attention-based design by [**Att-MLP**].

Fernando *et al.*[58] propose *hardwired attention* weights based on the distance of the neighbour from the pedestrian of interest. Amirian *et al.*[62] obtain the attention weights using hidden-state of primary pedestrian and interaction feature vectors of the neighbours learnt from pre-defined geometric features. On top of the attention mechanism, Zhang *et al.* [53] proposes a gating mechanism



(a) Occupancy pooling [16] (b) Directional pooling [Ours] (c) Social pooling [16]



(d) Encoding the grid tensor

Figure 3: Illustration of the grid-based interaction encoding modules. (a) Occupancy Pooling: each cell indicates the presence of a neighbour (b) Our proposed Directional Pooling: each cell contains the relative velocity of the neighbour with respect to the primary pedestrian. (c) Social Pooling: each cell contains the LSTM hidden-state of the neighbour (d) The constructed grid tensors are passed through an MLP-based neural network to obtain the interaction vector

to select the hidden-state features to consider for interaction adaptively. The attention mechanism can be applied multiple times to model higher-order spatial interactions.

Gupta *et al.* [52] proposed to aggregate the interaction information by applying a symmetric Max-Pooling function on the LSTM hidden-states of the neighbouring pedestrians. Non-grid based methods do not contain an implicit notion of the spatial position of neighbours with respect to the primary pedestrian. This problem is usually tackled by additionally concatenating an embedded representation of the relative positions of the surrounding pedestrians like in [52]:

$$r_{ji}^t = \phi_1(x_j^t - x_i^t; W_r), \tag{7}$$

$$h_{emb_j}^t = \phi_2(h_j^t; W_h), \tag{8}$$

$$hr_{ji}^t = \phi_3([r_{ji}^t; h_{emb_j}^t]; W_{rh}), \tag{9}$$

$$p_i^t = MaxPool(hr_{1i}^t, hr_{2i}^t, \ldots, hr_{ni}^t), \tag{10}$$

where ϕ_1, ϕ_2, ϕ_3 are MLP and the embedding weights W_r, W_h, W_{rh} are learned. We denote this architecture by [**MaxPool-MLP**].

The aggregating mechanisms mentioned above, namely attention and max-pooling, merge the

information of the neighbours resulting in the loss of their identity. We propose an additional design to compare against the above-discussed methods, maintaining the uniqueness of each pedestrian: we *concatenate* the embeddings of the relative position of the neighbours.

$$r_{ji}^t = \phi_1(x_j^t - x_i^t; W_r), \tag{11}$$

$$p_i^t = Concat(r_{1i}^t, r_{2i}^t, \ldots, r_{ni}^t), \tag{12}$$

$$\tag{13}$$

where ϕ_1 is an MLP and the embedding weights W_r, W_p are learned. We denote this architecture by [**Concat-MLP**]. The architecture is illustrated in Fig **??**. The issue with our proposed baseline is handling a variable number of pedestrians in a scene, *i.e.*, the concatenated vector is required to have a fixed length. To tackle this, we investigate the performance of this scheme by filtering n neighbours based on a defined criterion, (*e.g.*, euclidean distance). The different aggregation strategies are illustrated visually in Fig 4

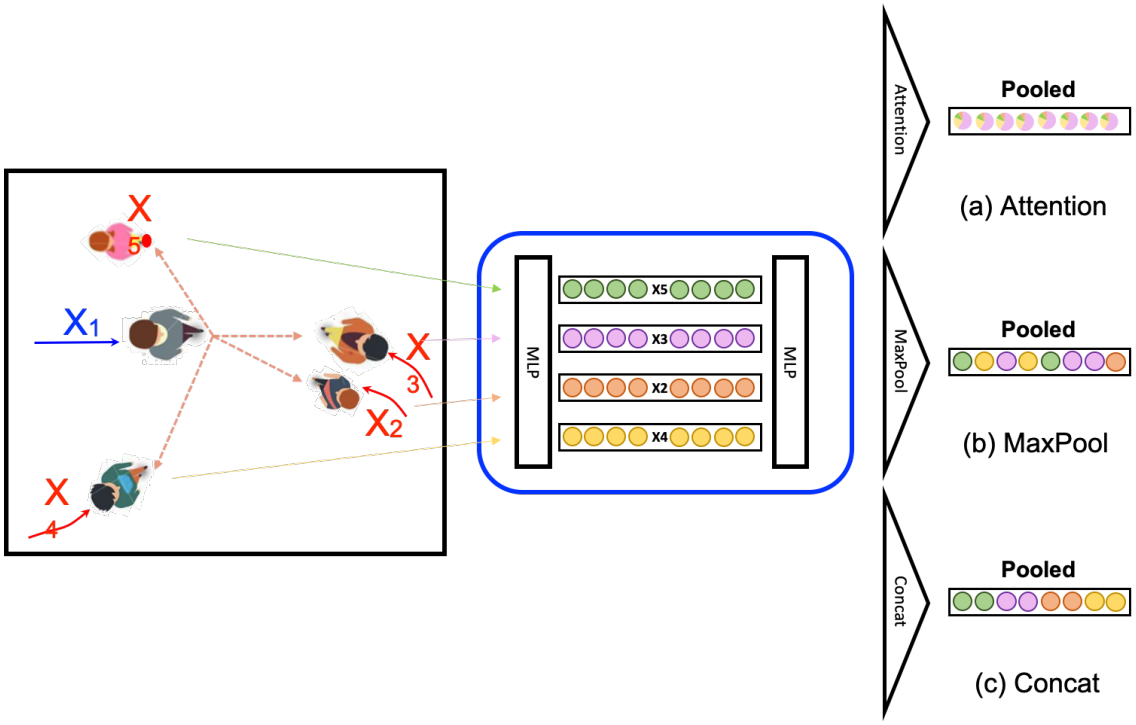


Figure 4: Illustration of the non-grid based encoding modules to obtain the interaction vector (*pooled*). The challenge lies in handling a variable number of neighbours and aggregating their state information to construct the interaction vector (a) Neighbour information is aggregated via attention mechanism (b) Neighbour information is aggregated utilizing a symmetric function (c) Neighbour information is aggregated via concatenation.

In the interaction modules described till now, each neighbouring state has been encoded and merged using an MLP. [55, 77, 64] propose to utilize an RNN-based modules for these tasks as shown in Fig 5. [77, 64] use an RNN architecture to update the neighbouring state information recurrently.

The authors define 'spatial edgeRNNs' to model the dynamics of human-human interactions. Each connection of a primary pedestrian to his neighbour is a different spatial edgeRNN. The relative position of each neighbour is passed to the RNN at each time-step. An attention mechanism is then implemented to weigh the different spatial edges at each time-step as follows:

$$r_{ji}^t = \phi_1(x_j^t - x_i^t; W_r), \tag{14}$$

$$e_{ji}^t = RNN(e_{ji}^{t-1}, r_{ji}^t; W_{RNN}), \tag{15}$$

$$p_i^t = \sum_j a_{ji}^t * e_{ji}^t, \tag{16}$$

where ϕ_1 is an MLP and the embedding weights W_r, W_{RNN} are learned. The weights a_{ji}^t are derived using an attention mechanism. We denote this architecture by [**Att-LSTM**].

Similarly, Ivanovic *et al.* [55] defines an LSTM-based 'edge encoder' connecting the primary pedestrian to the rest of the pedestrians in the scene. At each time-step, the states of the neighbouring pedestrian are summed and passed as input to the 'edge encoder' to handle variable number of pedestrians. In other words, the authors utilize an LSTM to provide a representation of the aggregated vector.

$$e_i^t = [\mathbf{s}_i^t; \sum_{j \in N(i)} \mathbf{s}_j^t], \tag{17}$$

$$p_i^t = LSTM(p_i^{t-1}, e_i^t; W_{EE}), \tag{18}$$

where W_{EE} denote the LSTM weights and $\mathbf{s}_i^t$ signifies the state of pedestrian i are time-step t. We denote this architecture by [**SumPool-LSTM**].

We argue that encoding the aggregated vector using LSTMs offers the advantage of modelling higher-order interactions in the temporal domain. In other words, the interaction module learns how the interaction representations evolve over time. We now propose our non-grid based interaction module called **DirectConcat** combining the strengths of LSTM-based interaction modelling and aggregation through concatenation. Mathematically, our proposed module has the following recurrence:

$$r_{ji}^t = \phi_1([\mathbf{x}_j^t - \mathbf{x}_i^t, \mathbf{v}_j^t - \mathbf{v}_i^t]; W_r), \tag{19}$$

$$e_i^t = Concat(r_{1i}^t, r_{2i}^t, \ldots, r_{ni}^t), \tag{20}$$

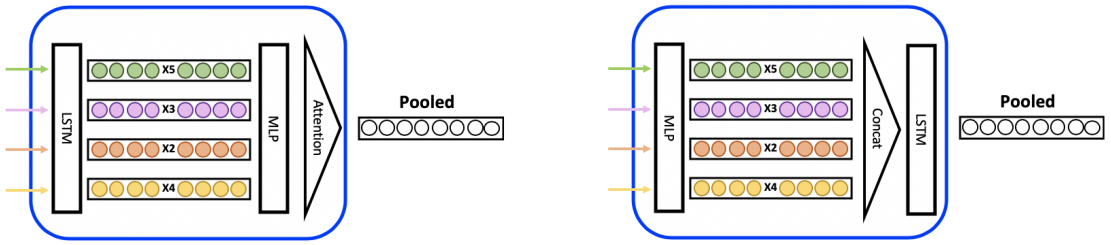
$$h_i^t = LSTM(h_i^{t-1}, e_i^t; W_{EE}), \tag{21}$$

$$p_i^t = \phi_2(h_i^t; W_p), \tag{22}$$

where ϕ_1, ϕ_2 are MLP and the weights W_r, W_p and W_{EE} are learned. In our design, we augment the relative velocity of the neighbours with their relative positions. We will demonstrate in the experimental section that this step greatly boosts the performance metrics. We denote this architecture by [**Concat-LSTM**]. The different LSTM-encoding based interaction modules are illustrated in Fig 5.

3.3 Forecasting Model

We now describe the rest of the components of the forecasting model. To claim that a particular design of the interaction module is superior, it is essential to keep the rest of the forecasting model components constant. Only then we can be sure that it was the interaction module design that boosted performance, and not one of the extra added components. We choose the time-sequence



(a) LSTM encoding of state information

(b) LSTM encoding of concatenated vector

Figure 5: Illustration of LSTM-based architectures for interaction encoding. (a) LSTM-based architectures used for encoding the state information (b) LSTM-based architectures used for encoding the concatenated information

encoder to be an LSTM due to its capability to handle varying input length and capture long-term dependencies. Moreover, most works have LSTMs as their base motion-encoding architecture.

The rest of the architecture we describe now is identical for all the methods described in the previous section. The state of person i at time-step t, $\mathbf{s}_i^t$, is embedded using a single layer MLP to get the state embedding e_i^t. We represent each person's state using his/her velocity, as switching the input representation from absolute coordinates to velocities increases the generalization power of sequence encoder. We obtain the interaction vector p_i^t of person i from the interaction encoder. We concatenate the interaction vector with the velocity embedding and provide the resultant vector as input to the sequence-encoding module. Mathematically, we obtain the following recurrence:

$$e_i^t = \phi(\mathbf{v}_i^t; W_{emb}), \tag{23}$$

$$h_i^t = LSTM(h_i^{t-1}, [e_i^t; p_i^t]; W_{encoder}), \tag{24}$$

where ϕ is the embedding function, $W_{emb}, W_{encoder}$ are the weights to be learned. The weights are shared between all persons in the scene.

The hidden-state of the LSTM at time-step t of pedestrian i is then used to predict the distribution of the velocity at time-step $t+1$. Similar to Graves [78], we output a bivariate Gaussian distribution parametrized by the mean $\mu_i^{t+1} = (\mu_x, \mu_y)_i^{t+1}$, standard deviation $\sigma_i^{t+1} = (\sigma_x, \sigma_y)_i^{t+1}$ and correlation coefficient ρ_i^{t+1}:

$$[\mu_i^t, \sigma_i^t, \rho_i^t] = \phi_{dec}(h_i^{t-1}, W_{dec}), \tag{25}$$

where ϕ_{dec} is modelled using an MLP and W_{dec} is learned.

Training: All the parameters of the forecasting model are learned by minimizing the negative log-likelihood (NLL) loss:

$$\mathcal{L}_i(w) = -\sum_{t=T_{obs}+1}^{T_{pred}} \log(\mathbb{P}(\mathbf{v}_i^t | \mu_i^t, \sigma_i^t, \rho_i^t)). \tag{26}$$

Contrary to the general practice of training the model by minimizing the NLL loss for all the trajectories in the training dataset, we minimize the loss for only the primary pedestrian (defined

in the next section) in each scene of the training dataset. We will demonstrate how this training procedure helps the model better capture social interactions in the experimental section.

Testing: During test time, till time-step T_{obs}, we provide the ground truth position of all the pedestrians as input to the forecasting model. From time T_{obs+1} to T_{pred}, we use the predicted position (derived from the predicted velocity) of each pedestrian as input to the forecasting model and predict the future trajectories of all the pedestrians.

4 TrajNet++: A Trajectory Forecasting Benchmark

In this section, we present *TrajNet++*, our interaction-centric human trajectory forecasting benchmark. To demonstrate the efficacy of a trajectory forecasting model, the standard practice is to evaluate these models against baselines on a standard benchmark. However, current methods have been evaluated on different subsets of available data without proper sampling of scenes in which social interactions occur. In other words, a data-driven method cannot learn to model agent-agent interactions if the benchmark comprises primarily of scenes where the agents are static or move linearly. Therefore, our benchmark comprises largely of scenes where social interactions occur. To this extent, we propose the following trajectory categorization hierarchy.

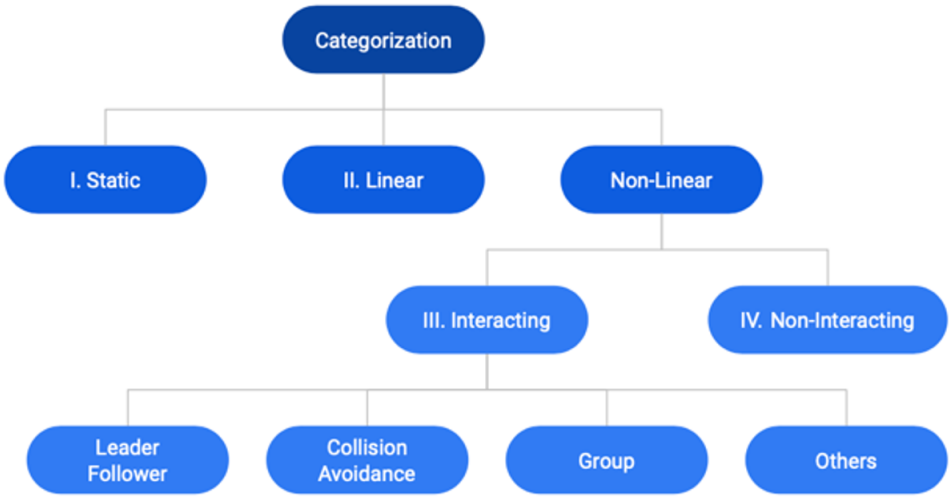


Figure 6: Our proposed hierarchy for Trajectory Categorization. Using our defined trajectory categorization, we construct the *TrajNet++* benchmark by sampling trajectories corresponding largely to 'Type III: Interacting' category.

4.1 Trajectory Categorization

We provide a detailed trajectory categorization (Fig 6). This detailed categorization helps us not only to better sample trajectories for TrajNet++ dataset but also glean insights into the model performance in diverse scenarios, *i.e.*, to verify whether the model captures all the different kinds of interactions.

To aid our categorization, we introduce the notion of a *primary pedestrian* as a reference pedestrian with respect to which we categorize scenes. Each scene has a primary pedestrian whose motion we want to forecast. We refer to the other pedestrians in the scene as *neighbouring pedestrians.*

We explain in detail our proposed hierarchy for trajectory categorization (Fig 6). We also provide example scenarios for the same in Fig 7:

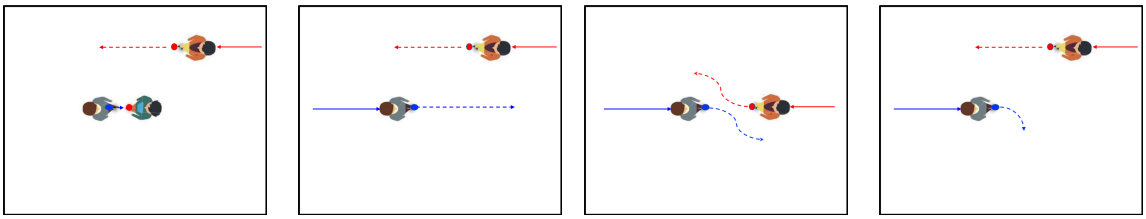
- **Static (Type I)**: If the euclidean displacement of the primary pedestrian in the scene is less than a specific threshold.
- **Linear (Type II)**: If the trajectory of the primary pedestrian can be *correctly forecasted* with the help of an Extended Kalman Filter (EKF). A trajectory is said to be *correctly forecasted* by EKF if the FDE between the ground truth trajectory and forecasted trajectory is less than a specific threshold.

  The rest of the scenes are classified as 'Non-Linear'. We further divide non-linear scenes into Interacting (Type III) and Non-Interacting (Type IV).
- **Interacting (Type III)**: These correspond to scenes where the primary trajectory undergoes social interactions. For a detailed categorization coherent with commonly observed social interactions, we divide interacting trajectories into the following sub-categories (shown in Fig 8).
  - **Leader Follower [LF] (Type IIIa)**: Leader follower phenomenon refers to the tendency to follow pedestrians going in relatively the same direction. The follower tends to regulate his/her speed and direction according to the leader. If the primary pedestrian is a follower, we categorize the scene as Leader Follower.
  - **Collision Avoidance [CA] (Type IIIb)**: Collision avoidance phenomenon refers to the tendency to avoid pedestrians coming from the opposite direction. We categorize the scene as Collision avoidance if the primary pedestrian to be involved in collision avoidance.
  - **Group (Type IIIc)**: The primary pedestrian is said to be a part of a group if he/she maintains a close and roughly constant distance with at least one neighbour on his/her side during prediction.
  - **Other Interactions [Others] (Type IIId)**: Trajectories where the primary pedestrian undergoes social interactions other than LF, CA and Group. We define *social interaction* as follows: We look at an angular region in front of the primary pedestrian. If any neighbouring pedestrian is present in the defined region at any time-instant during prediction, the scene is classified as having the presence of social interactions.
- **Non-Interacting (Type IV)**: If a trajectory of the primary pedestrian is non-linear and undergoes no social interactions during prediction.

Using our defined trajectory categorization, we construct the *TrajNet++* benchmark by sampling trajectories corresponding mainly to the Type III category. Moreover, having many Type I scenes in a dataset can hamper the training of the model and result in misleading evaluation. Therefore, we remove such samples in the construction of our benchmark. A few examples of our categorization in the real world are displayed in Fig 9. In addition to comprising well-sampled trajectories, *TrajNet++* provides an extensive evaluation system to understand model performance better.

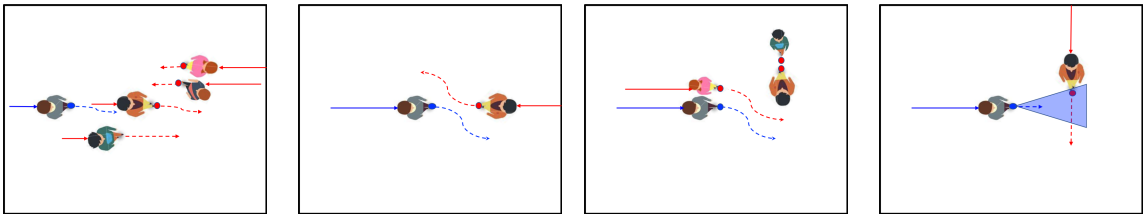
4.2 Evaluation Metrics

Unimodal Evaluation: Unimodal evaluation refers to the evaluation of models that propose a single future mode for a given past observation. The most commonly used metrics of human trajectory



(a) Static (b) Linear (c) Interacting (d) Non-Interacting

Figure 7: Visualization of our high-level defined trajectory categories: (a) Static (b) Linear (c) Interacting (d) Non-Interacting



(a) Leader Follower (b) Collision Avoidance (c) Group (d) Others

Figure 8: Visualization of our Type III interactions commonly occurring in real world crowds: (a) Leader Follower (b) Head-On collision avoidance (c) Group (d) Others

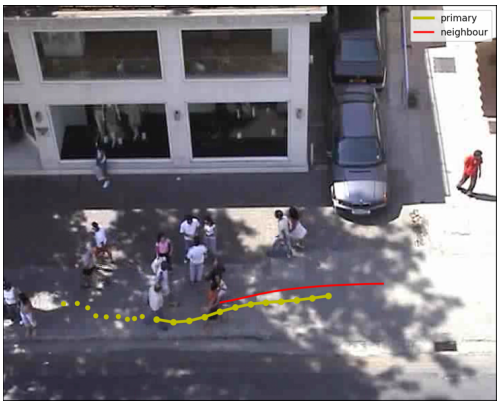
forecasting in the unimodal setting are Average Displacement Error (ADE) and Final Displacement Error (FDE) defined as follows:

1. **Average Displacement Error (ADE)**: Average L_2 distance between ground truth and model prediction overall predicted time steps.
2. **Final Displacement Error (FDE)**: The distance between the predicted final destination and the ground truth final destination at the end of the prediction period T_{pred}.

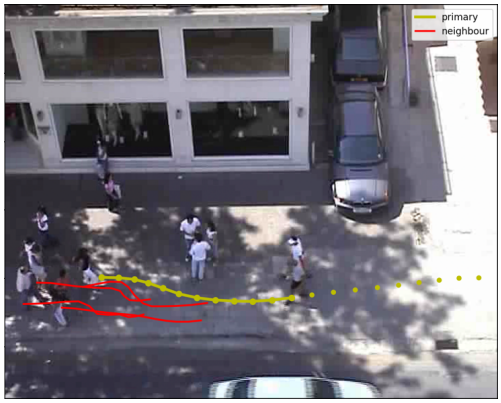
These metrics essentially define different distance measures between the forecasted trajectory and the ground truth trajectory. With respect to our task, one of the most important aspects of human behavior in crowded spaces is collision avoidance. To ensure that models forecast feasible collision-free trajectories, we propose two new collision-based metrics in our framework (see Fig 10):

3. **Collision I - Prediction collision (Col-I)**: This metric calculates the percentage of collision between the primary pedestrian and the neighbors in the *forecasted future* scene. This metric indicates whether the predicted model trajectories collide, *i.e.*, whether the model learns the notion of collision avoidance.
4. **Collision II - Groundtruth collision (Col-II)**: This metric calculates the percentage of collision between the primary pedestrian's prediction and the neighbors in the *groundtruth future* scene.

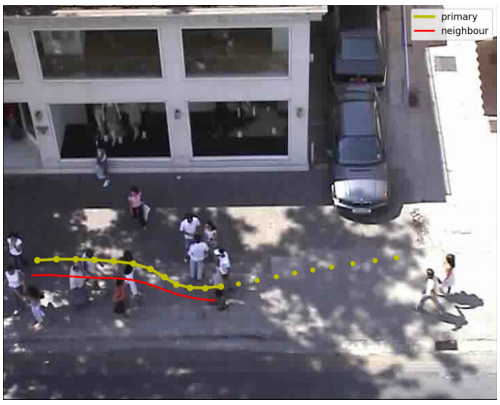
We want to stress further the importance of the collision metrics in the unimodal setup. As mentioned earlier, human motion is multimodal. A model may forecast a physically-feasible future, which is different from the actual ground truth. Such a physically-feasible prediction can result in



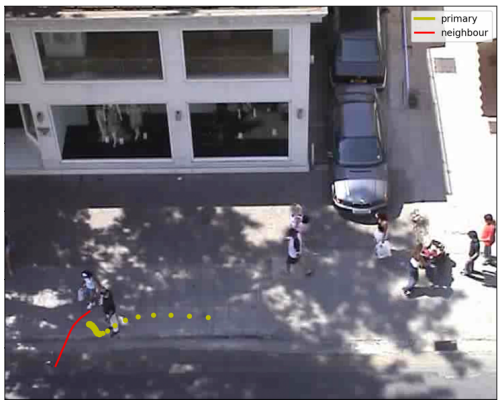
(a) Leader Follower



(b) Collision Avoidance



(c) Group



(d) Others

Figure 9: Visualization of our Type III interactions in real world datasets

a large ADE/FDE, which can be misleading. Our Col-I metric can help overcome this limitation of ADE/FDE metrics and provides a solution to measure 'physical feasibility' of a prediction (aversion to a collision in this case). Col-II metric indicates whether the model understood the intention of the neighbours and predicted the desired trajectory mode indicated by fewer collisions with neighbours in ground truth. We believe our proposed collision metrics are an important step towards capturing the understanding of the model of human social etiquette in crowds.

Multimodal Evaluation: For models performing multimodal forecasting, *i.e.*, outputting a future trajectory distribution, we provide the following metrics to measure their performance:

5. **Top-k ADE**: Given k output predictions for an observed scene, this metric calculate the ADE of the prediction *closest* to the groundtruth trajectory, similar in spirit to Variety Loss [52].

6. **Top-k FDE**: Given k output predictions for an observed scene, this metric calculate the FDE of the prediction *closest* to the groundtruth trajectory, similar in spirit to Variety Loss [52].

For the Top-k metrics, we propose k be small (3 as opposed to 20) as a model outputting

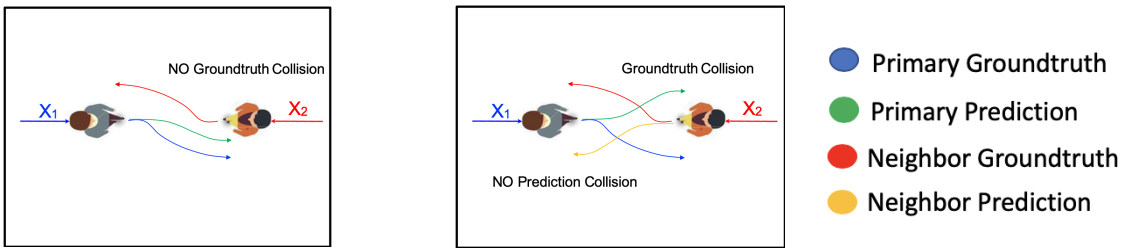


Figure 10: Visualizations of our proposed collision metrics. (a) Col-I: This metric calculates the collision between the model prediction of primary pedestrian and model prediction of neighbours. (b) Col-II: This metric calculates the collision between the model prediction of primary pedestrian and ground truth of neighbours. Our proposed collision metrics help to quantify the ability of the model to understand human social rules in crowds (collision avoidance in this case)

uniformly-spaced predictions, irrespective of the input observation, can result in a much lower Top-20 ADE/FDE.

7. **Average NLL**: This metric was proposed by Boris *et. al.* [55]. At each prediction step, the authors utilize a Kernel Density Estimate (KDE) [79]. From these estimates, the log-likelihood of ground truth trajectory is computed at each time step and is subsequently averaged over the prediction horizon. This metric provides a good indication of the probability of the ground truth trajectory in the model prediction distribution.

4.3 Datasets

We now describe the datasets used in the TrajNet++ benchmark. Since the focus of this work is to tackle agent-agent interactions in crowded settings, we explicitly select datasets where scene constraints do not play a significant role in determining the future trajectory. For each real world dataset, we utilize only the information regarding the pedestrian locations from the respective annotations files, *i.e.*, spatial coordinates of each pedestrian at each time frame. Furthermore, we provide no information regarding the destination of each pedestrian or structure of the scene. Our goal is to forecast only the 2D spatial coordinates for each pedestrian.

4.3.1 TrajNet++ Real Datasets

- **ETH:** ETH dataset provides for two locations: Univ and Hotel, where pedestrian trajectories are observed. This dataset contains a total of approximately 750 pedestrians exhibiting complex interactions (Pellegrini *et. al.* [25]). The dataset is one of the widely used benchmarks for pedestrian trajectory forecasting. It captures diverse real-world social interactions like leader follower, collision avoidance, and group forming and dispersing.
- **UCY:** UCY dataset consists of three scenes: Zara01, Zara02 and Uni, with a total of approximately 780 pedestrians (Lerner *et. al.* [2]). This dataset, in addition to the ETH dataset, is widely used as benchmarks for pedestrian trajectory forecasting, offering a wide range of non-linear trajectories arising out of social interactions.
- **WildTrack:** This is a recently proposed benchmark [80] for pedestrian detection and tracking captured in front of ETH Zurich. Since the dataset comprises of diverse crowd interactions in the wild, we utilize it for our task of trajectory forecasting.

- **L-CAS:** This is a recently proposed benchmark for pedestrian trajectory forecasting (Sun *et. al.* [72]). The dataset, comprising over 900 pedestrian tracks, comprises diverse social interactions that are captured within indoor environments. Some of the challenges scenarios in this dataset include people pushing trolleys and running children.

- **CFF:** This is a large-scale dataset of 42 million trajectories extracted from real-world train stations [36]. It is one of the biggest datasets that capture agent-agent interactions in crowded settings during peak travel times. Due to the high density of people, we observe higher instances of social interactions like leader-follower in this dataset.

4.3.2 TrajNet++ Synthetic Dataset

Interaction-centric synthetic datasets can provide the necessary controlled environment to compare the performances of different model components. We provide synthetic data in TrajNet++ to evaluate the performance of a model under controlled interaction scenarios.

Simulator Selection: It is a necessary condition that the interactions in the synthetic dataset are similar to those in the real world. Empirically, we find that in comparison to Social Force [13], ORCA [29] provides a better similarity to real world human motion with respect to collision avoidance. We choose ORCA parameters, which demonstrate a reaction distance and reaction curvature similar to real data during collision avoidance (Fig 11).

Dataset Generation: Given the ORCA parameters, we generated the synthetic dataset using the following procedure: n pedestrians were initialized at random on a circle of radius r keeping a certain minimum distance d_min between their initial positions. The goal of each pedestrian was defined to be the point diametrically opposite to the initial position on the circle. For the TrajNet++ synthetic dataset: We ran different simulations with n chosen randomly from the range [4, 7) on a circle of radius $r = 10$ meters and $d\_min = 2$ meters.

Given the generated trajectories, we selected only those scenes which belonged to the Type III: 'Interacting' category. The ORCA simulator demonstrates sensitive dependence on initial conditions. This can be attributed to the fact that all the agents are expected to collide near the same point (at the origin), so slight perturbations can greatly affect the future trajectory of all agents. Sensitivity to initial conditions, also known as the *Butterfly Effect*, is a well-studied phenomenon of *Chaos theory*[81]. To identify such sensitive initial conditions, the practice which is often followed is to perturb the initial conditions with arbitrary small noise and observe the effect. Along similar lines, we propose an additional step to filter out such 'sensitive' scenes: in each scene, we perturb all trajectories at the point of observation with a small uniform noise ($noise \in U[-noise\_thresh, noise\_thresh]$), and forecast the future trajectories using ORCA. We perform this procedure k times. If any of the k ORCA predictions have a significant ADE compared to the ground truth, we filter out such scenes. Fig 12 visualizes the sample outputs of our filtering process (with $noise\_thresh = 0.01$, $k = 20$, $n = 5$). We passed the selected scenes through a final additional filter that identifies sharp unrealistic turns in trajectories. Fig 13 illustrates a few sample scenes in our TrajNet++ synthetic dataset.

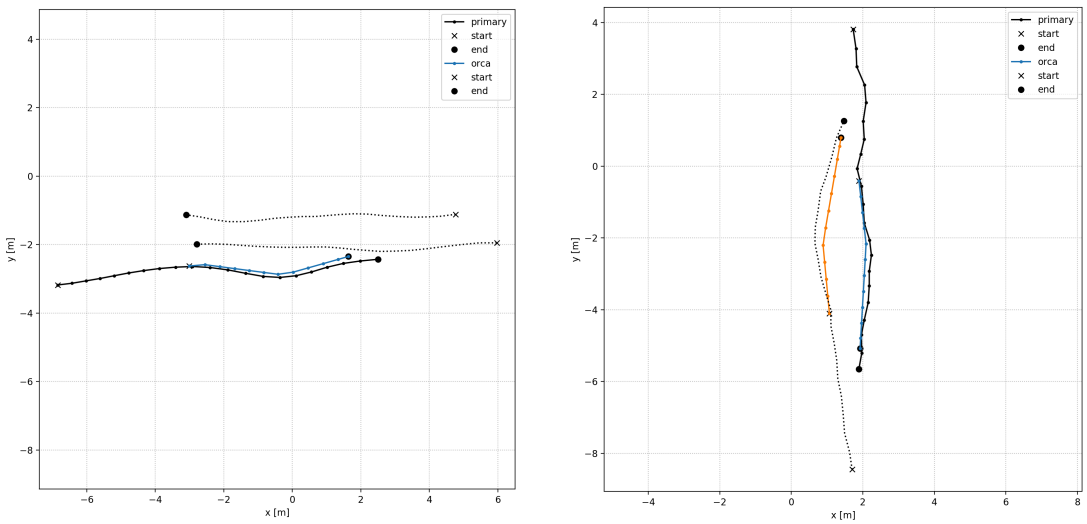


Figure 11: Our calibrated ORCA parameters show similar reaction curvature (in blue) as the social interactions in real world datasets (in black). Solid line denotes the primary pedestrian. Dotted lines denote the neighbours.

5 Experiments

In this section, we perform extensive experimentation on both TrajNet++ synthetic and real-world datasets to understand the efficacy of interaction module designs for human trajectory forecasting. Moreover, we demonstrate how our proposed metrics help to provide a complete picture of model performance. Our proposed simple yet powerful method outperforms competitive baselines on both real-world and synthetic datasets in terms of forecasting physically-acceptable trajectories.

5.1 Implementation Details

The velocity of each pedestrian is embedded into a 64-dimensional vector. The dimension of the interaction vector is fixed to 256. The dimension of the goal direction vector is fixed to 64. For grid-based interaction encoding, we construct a grid of size 16×16 with a resolution of 0.6 meters. The dimension of the hidden state of both the encoder LSTM and decoder LSTM is 128. As mentioned earlier, each pedestrian has his/her own encoder and decoder. The batch size is fixed to 8. We train using ADAM optimizer [82] with a learning rate of 1e-3. We perform interaction encoding at every time-step.

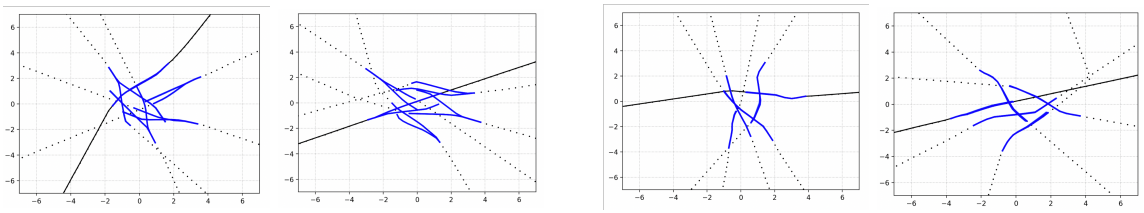
Our proposed trajectory categorization allows one to train the model focusing on the non-linear interacting trajectories. Since each scene is categorized with respect to the primary pedestrian, during training, the loss is calculated only with respect to the prediction of the primary trajectory.

Name	Total	I	II	III	LF	CA	Grp	Oth	IV
Synthetic	54513	0	0	54513	495	7183	0	46853	0
BIWI Hotel.	238	22	91	109	24	29	41	39	16
Zara01.	1017	4	184	542	109	160	231	134	287
Zara03.	960	16	152	634	108	222	232	200	158
Stud01.	5719	94	605	4772	712	2030	1862	1364	248
Stud03.	4302	46	350	3598	537	1508	1469	1118	308
WildTr.	1098	115	43	668	43	75	145	422	272
L-CAS	874	180	87	310	10	85	12	210	297
CFF06	20972	22	4267	15384	5194	8239	267	4841	1299
CFF07	21145	16	4251	15635	5352	8361	252	4991	1243
CFF08	19840	13	3950	14619	4805	7521	216	4881	1258
CFF09	10548	10	2579	6717	1733	3010	203	2742	1242
CFF12	20962	11	4242	15445	5309	8294	268	4890	1264
CFF13	19792	17	3746	14679	4768	7519	263	4898	1350
CFF14	20509	12	4041	15135	5099	7893	274	4927	1321
CFF15	19866	15	3824	14741	4815	7563	277	4984	1286
CFF16	10044	5	2523	6258	1626	2689	249	2566	1258
CFF17	9250	8	2458	5694	1508	2694	198	2227	1090
CFF18	19437	13	4067	14042	4744	7211	248	4669	1315
Total	**250k**								

Table 1: TrajNet++: Statistics of the Training Split.

Name	N	I	II	III	LF	CA	Grp	Oth	IV
Synthetic	3842	0	0	3842	73	632	0	3142	0
BIWI ETH.	1139	11	227	640	189	153	172	244	261
UNI.	244	0	50	100	7	11	38	49	94
Zara02.	1881	80	496	998	192	355	452	270	307
Total	**7106**								

Table 2: TrajNet++: Statistics of the Testing Split.



(a) Sensitive Scenes

(b) Insensitive Scenes

Figure 12: Illustration of our filtering procedure to generate Trajnet++ Synthetic dataset. Given a ground-truth scene (in black) generated by ORCA, we perturb the positions of agents and forecast the future with ORCA, iteratively, to obtain a distribution (in blue). This procedure helps us identify the *sensitive scenes* and consequently remove them.

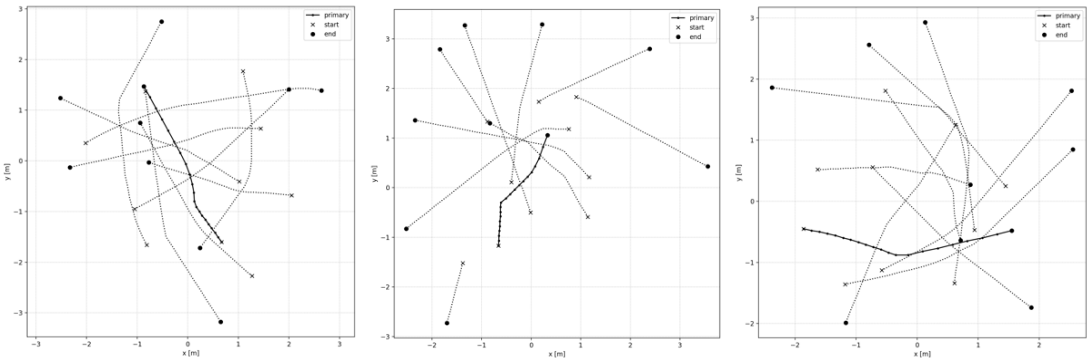


Figure 13: Illustration of our synthetically generated samples using the calibrated ORCA parameters.

5.2 Interaction Models

We consider the following grid based baselines:

- Vanilla [**Van**]: No interaction encoding takes place.
- Occupancy Pooling [**O-Grid**] [16, 45]: Grid based interaction model where each cell indicates the presence of the neighbour
- Social Pooling [**S-Grid**] [16, 45, 47, 51]: Grid based interaction model where each cell comprises of the hidden-state of the neighbour
- Directional Pooling [**D-Grid**] [Ours]: Grid based interaction model where each cell comprises of the relative velocity of the neighbour.

The grid is then passed through a two layer MLP to get the interaction vector. We experimented with various feedforward architectures and a two layer MLP performs the best for encoding the grid.

We consider the following are the non-grid based baselines:

- Positional Concatenation MLP [**O-Concat-MLP**]: The relative position of each neighbour is embedded and *concatenated* with each other. The resulting vector is then passed through an MLP to get the interaction vector. We consider the top-n neighbours based on euclidean distance (n being a hyperparameter).

- Directional Concatenation MLP [**D-Concat-MLP**]: The relative position and relative velocity of each neighbour are embedded and *concatenated* with each other. The resulting vector is then passed through an MLP to get the interaction vector.
- Directional MaxPool MLP [**D-MaxPool-MLP**]: The relative position and relative velocity of each neighbour are embedded and *max-pooled*. The resulting vector is then passed through an MLP to get the interaction vector.
- Directional Attention MLP [**D-Attn-MLP**]: The relative position and relative velocity of each neighbour are embedded and passed through a *self-attention block*. The resulting vector is then passed through an MLP to get the interaction vector.
- Hidden State MaxPool MLP [**H-MaxPool-MLP**] [52]: The relative position and hidden-state of each neighbour are embedded and *max-pooled*. The resulting vector is then passed through an MLP to get the interaction vector.
- Social Attention MLP [**H-Att-MLP**] (similar to [61, 58, 57, 53, 62]): The relative position, relative velocity and hidden-state of each neighbour are embedded. The embeddings are passed through a *self-attention block* [68]. The resulting vector is then passed through an MLP to get the interaction vector.
- Sum Pool LSTM [**D-SumPool-LSTM**] [55]: The *absolute position and velocity* of each neighbour is *summed* and concatenated to that of primary pedestrian. The resulting vector is then passed through an LSTM to get the interaction vector.
- Social Attention LSTM [**O-Att-LSTM**] [77]: The relative position of each neighbour are encoded through LSTMs. The hidden-states of LSTMs are passed through a *self-attention block* to get the interaction vector.
- Directional Concatenation LSTM [**D-Concat-LSTM**] (Ours): The *relative position and relative velocity* of each neighbour are embedded and concatenated. The resulting vector is then passed through an LSTM to get the interaction vector.

Please note that for an objective comparison between interaction modules, we fix the base sequence encoder architecture to be an LSTM. Data augmentation is another technique that can help increase accuracy, which can get wrongly attributed to the interaction encoder. We use rotation augmentation as the data augmentation technique to regularize all the models.

5.3 Synthetic Experiments

Synthetic datasets are the ideal testbeds to validate model performances in noise-free controlled scenarios explicitly. The synthetic dataset is generated using the procedure described in Section 4.3.2. For the synthetic dataset, since ORCA has access to the goals to the pedestrian, we embed the goal-direction and concatenate it to the velocity embedding (see Eq 23). We utilize synthetic datasets to validate the efficacy of various interaction modules in a controlled setup.

Unimodal Evaluation: Table 3 quantifies the performance of the different designs of interaction modules published in the literature on TrajNet++ synthetic dataset. Among the grid-based models, our proposed **D-Grid** outperforms **O-Grid**, especially in terms of Col-I, *i.e.*, **D-Grid** learns better to avoid collisions. It is interesting to note that even though the motion encoder (LSTM) has the potential to infer the relative velocity of neighbours over time, there is significant difference in performance when we provide relative velocity of the neighbours as input to the pooling grid.

Model (Acronym)	Merge	Enc.	ADE	FDE	Col-I	Col-II
Grid based methods						
Vanilla	–	–	0.32	0.62	19.2	7.1
O-LSTM [16] (O-Grid)	Grid	MLP	0.27	0.53	10.1	5.0
S-LSTM [16] (S-Grid)	Grid	MLP	**0.24**	**0.50**	**2.0**	**4.4**
D-LSTM (**Ours**) (D-Grid)	Grid	MLP	0.25	**0.50**	2.4	4.8
Non-Grid based methods						
S-GAN [52] (H-MaxPool-MLP)	MaxPool	MLP	0.27	0.52	6.8	5.2
S-BiGAT [61] (H-Att-MLP)	Attention	MLP	0.25	0.50	2.5	5.8
DirectConcat-MLP (**Ours**) (D-Concat-MLP)	Concat	MLP	0.25	0.50	1.3	5.6
Trajectron [55] (D-SumPool-LSTM)	SumPool	LSTM	0.29	0.57	14.0	6.5
Social Attention [77] (O-Att-LSTM)	Attention	LSTM	**0.24**	**0.48**	1.0	**5.1**
DirectConcat (**Ours**) (D-Concat-LSTM)	Concat	LSTM	**0.24**	**0.48**	**0.7**	5.2

Table 3: Baseline models compared according to their interaction encoder designs when forecasting 12 future time-steps, given the previous 9 time-steps on TrajNet++ synthetic dataset. The interaction model design is categorized with respect to neighbour information aggregation (**Merge**) strategy and type of the encoder architecture (**Enc.**). Errors reported are ADE / FDE in meters, Col I / Col II in % as defined in Section 4.2

Further, **D-Grid** performs at par with **S-Grid**, along with being computationally less expensive, thereby rendering it more suitable for real-world deployment tasks.

Among the non-grid based models, we focus on the information aggregation strategies for MLP-based encoders. It is evident that our baseline **D-Concat-MLP** of *concatenating* the neighbourhood information performs at par, if not better than weighting-based and max-pooling-based alternatives. This performance can be attributed to the fact that the interaction vector obtained using **D-Concat-MLP** preserves the identity of the surrounding neighbours.

Among the non-grid LSTM-based designs, the drop in performance of **D-SumPool-LSTM** module [55] can be attributed to (1) sum pooling which loses the individual identity of the neighbours and (2) encoding of absolute neighbour coordinates instead of relative coordinates: relational coordinates of agents to the target agent are easier to train than exact coordinates of agents. On the other hand, the relatively higher Col-I metric for **O-Att-LSTM** [77] can be attributed to its design as that it does not account for the relative velocity between agents. Finally, we notice that encoding the interaction information using LSTM, **D-Concat-LSTM**, improves performance over its MLP-based counterpart **D-Concat-MLP**. MLP encoders, due to their non-recurrent nature, have no information regarding the representation at the previous step. We argue that LSTM can capture the evolution of interaction and therefore provide a better neighbourhood representation as the scene evolves. Moreover, having a separate LSTM for encoding interactions can reduce the load on the sequence-encoding LSTM that monitors past motion as well.

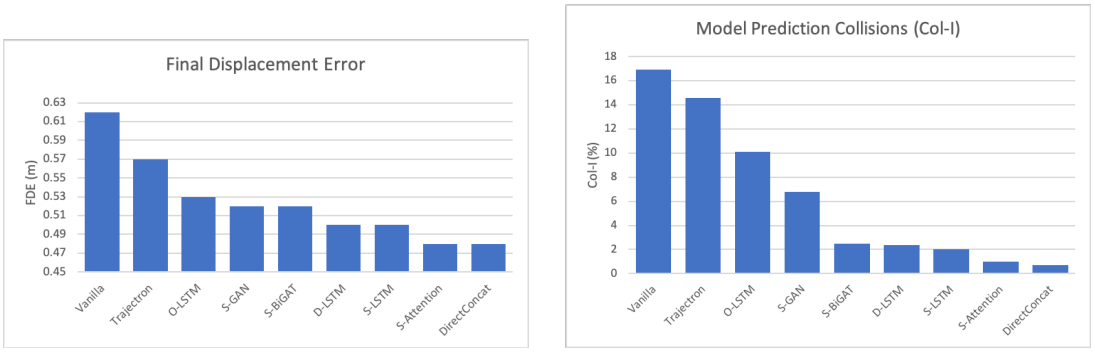


Figure 14: Graphical illustration of competitive baselines on TrajNet++ Synthetic Dataset.

5.4 Real World Experiments

We now discuss the performances of forecasting models on TrajNet++ real world data. With the help of our defined trajectory categorization, we construct the *TrajNet++* real-world benchmark by sampling trajectories corresponding mainly to 'Interacting' category. Moreover, many real-world trajectories are static (people in groups standing and talking to each other). Having many static scenes in a dataset can provide misleading results during evaluation. Therefore, we remove such samples from our benchmark. Having gained insights on the performance of different modules on controlled synthetic data, we explore the question, 'Do these findings generalize to the real world datasets comprising much more diverse interactions?'

Additional Baselines: We compare the data-driven baselines with the classical trajectory forecasting models, namely, Extended Kalman Filter (EKF), Social Force [13], and ORCA [29]. Both Social Force and ORCA models forecast the future trajectory based on the assumption that each pedestrian has an intended direction of motion and a preferred velocity, as a result of his/her intended goal. However, estimating this goal from the observed trajectory required us to know the person's intention, which we cannot access. Since we focus on short-term human trajectory forecasting, we interpolate the observed trajectory to identify the *virtual goals* for each agent.

Unimodal Evaluation: Table 4 provides an extensive evaluation of existing baselines on the Type III 'Interacting' trajectories of the TrajNet++ real dataset. The first part of the table compares the classical methods. The high error of EKF can be attributed to the fact that the filter does not model social interactions. The classical methods of **Social Force** and **ORCA** are calibrated to fit the TrajNet++ training data by minimizing ADE/FDE metrics, along with the constraint that collisions should be avoided [2].

The second part of Table 4 compares the performance of the various NN-based interaction encoder designs. The interaction-based NN models outperform the handcrafted models in terms of the distance-based metrics. Our collision metrics help to differentiate the NN-based model performance in terms of the physical acceptability of predictions. In contrast to synthetic experiments, our proposed **D-Grid** performs superior to **S-Grid** in terms of avoiding collisions in the real world. Furthermore, our proposed baseline **D-Concat-LSTM**, built from simple principles, performs at par, if not better, than the existing non-grid counterparts.

Multimodal Evaluation: TrajNet++ synthetic dataset was generated using ORCA, which simulates the scene following a deterministic unimodal policy. However, human motion in real world

[2]the parameter corresponding to minimum distance between neighbouring agents was fixed to be *greater than* our defined threshold of collision

Model (Acronym)	Merge	Enc.	ADE	FDE	Col-I	Col-II	NLL
Hand-crafted methods							
Kalman Filter	–	–	0.87	1.69	16.20	22.1	–
Social Force	–	–	0.89	1.53	**0.0**	**13.1**	–
ORCA	–	–	**0.68**	**1.40**	**0.0**	15.0	–
Top submitted methods*[1]							
AMENet [83]	–	–	0.62	1.30	14.1	16.90	–
AIN [84]	–	–	0.62	1.24	10.7	17.10	–
Grid based methods							
Vanilla	–	–	0.61	1.31	14.5	16.1	12.29
O-LSTM [16] (O-Grid)	Grid	MLP	0.56	1.21	11.3	15.6	11.43
S-LSTM [16] (S-Grid).	Grid	MLP	**0.55**	**1.19**	7.8	15.8	**10.01**
D-LSTM (**Ours**) (D-Grid)	Grid	MLP	0.57	1.25	**7.3**	**14.8**	11.22
Non-Grid based methods							
S-GAN [52] (H-MaxPool-MLP)	MaxPool	MLP	0.58	1.26	14.1	16.0	11.93
S-BiGAT [61] (Att-MLP)	Attention	MLP	0.60	1.29	8.3	16.4	**9.22**
Trajectron [55] (SumPool-LSTM)	SumPool	LSTM	0.58	1.25	15.6	16.4	12.70
Social Attention [77] (Att-LSTM)	Attention	LSTM	**0.55**	**1.19**	9.8	16.1	10.65
DirectConcat (**Ours**) (D-Concat-LSTM)	Concat	LSTM	0.57	1.24	**7.4**	**16.0**	9.78

Table 4: Baseline models compared according to their interaction encoder designs (see acronyms) when forecasting 12 future time-steps, given the previous 9 time-steps on TrajNet++ real world dataset. The model design is categorized with respect to the neighbour information mixing (**Merge**) strategy and the type of encoder architecture (**Enc.**). Errors reported are ADE / FDE in meters, Col I / Col II in %, NLL in units as defined in Sec 4.2

Model	Dataset	Rel. Pos.	Rel. Vel.	Merge	ADE	FDE	Col-I
O-Grid	Synthetic	✓	–	Grid	0.27	0.53	10.1
D-Grid	Synthetic	✓	✓	Grid	**0.25**	**0.50**	**2.4**
O-NN	Synthetic	✓	–	concat	0.28	0.53	10.4
D-NN	Synthetic	✓	✓	concat	**0.25**	**0.50**	**1.3**
O-Grid	Real	✓	–	Grid	**0.56**	**1.21**	11.3
D-Grid	Real	✓	✓	Grid	0.57	1.25	**7.3**
O-NN	Real	✓	–	concat	0.61	1.29	17.9
D-NN	Real	✓	✓	concat	**0.59**	**1.26**	**8.1**

Table 5: Providing relative velocity to the interaction modules leads to a significant boost in both synthetic and real-world settings.

is multimodal. Therefore, as described in Sec 4.2, our *TrajNet++* framework provides multimodal evaluation metrics in addition to unimodal metrics. For a complete evaluation, we report the performance of various methods trained in multimodal settings using the variety loss defined in [52], on TrajNet++ real dataset. In Table 4, we report the NLL metric that provides an estimate of the probability of the ground truth trajectory in the model prediction distribution. Among the grid-based models, **S-Grid** performs the best while among the non-grid based models, **Att-MLP** performs superior. Exploring techniques to output accurate yet diverse multimodal distributions is an avenue for future research.

To summarize, despite claims in literature that specific interaction modules better model interactions, we observe that under *identical* conditions, all modules perform similar in terms of the distance-based ADE and FDE metrics. There certainly exists room for improvement, and we hope that our benchmark provides the necessary resources to advance the field of trajectory forecasting.

5.5 Ablation Studies

While benchmarking the interaction modules on both synthetic and real datasets, we empirically observed important design choices and training strategies. We highlight them in this subsection through a series of ablation studies. Moreover, we open-source our code for reproducibility. We hope that such practices will help to accelerate the development of interaction modules in future research.

1. **Col-I is an essential evaluation metric:** Table 3 and Table 4 emphasize the importance of our proposed Col-I metric, *i.e.*, the percentage of collision of primary pedestrian with neighbors in the *forecasted* scene. This metric indicates the ability of the model to learn the social etiquette of collision avoidance. In safety-critical scenarios, it is more important for a model to prevent collisions in comparison to minimizing ADE/FDE. We hope that in future, researchers will incorporate this metric while reporting their model performances on trajectory forecasting datasets.

2. **Embedding relative velocity provides a boost:** Table 5 illustrates the improvement in performance on providing the relative velocity to the interaction modules. A key to the success of the interaction modules is to have informed input features. Having the relative velocity embedding significantly improves the performance of both grid and non-grid based models, especially in learning to avoid collisions. Empirically, one can argue that it is easier to provide the relative velocity to the interaction model compared to relying on the sequence encoder to infer the relative velocity through time.

3. **Concatenation of embeddings are simple yet powerful baselines:** Table 6 illustrates

Model	Dataset	Rel. Pos.	Rel. Vel.	Merge	ADE	FDE	Col-I
D-MLP	Synthetic	✓	✓	max-pool	0.28	0.55	14.3
D-Attn	Synthetic	✓	✓	attn	0.27	0.52	8.1
D-NN	Synthetic	✓	✓	concat	**0.25**	**0.50**	**1.3**
D-MLP	Real	✓	✓	max-pool	0.59	1.24	13.6
D-Attn	Real	✓	✓	attn	**0.56**	**1.23**	**7.6**
D-NN	Real	✓	✓	concat	0.59	1.26	8.1

Table 6: Concatenation of embeddings are simple yet powerful baselines for comparing different neighbour information aggregation strategies. For concatenation, we consider the top-4 neighbours based on euclidean distance

Strategy	Dataset	ADE	FDE	Col-I
Standard Training [52, 77, 55]	Synthetic	0.25	0.48	11.9
Proposed Training	Synthetic	**0.24**	**0.48**	**2.5**
Standard Training [52, 77, 55]	Real	0.59	1.27	14.8
Proposed Training	Real	**0.56**	**1.21**	**11.3**

Table 7: Our proposed training objective that penalizes only the prediction of the primary pedestrian, instead of penalizing all the pedestrians in the scene, provides superior performance with respect to helping the model learn to avoid collisions

an ablation study where the embeddings (relative position and relative velocity) of neighbours provided to the interaction module are identical; only the information aggregation strategy differs. We observe that concatenation performs superior to attention weighting and max-pooling, in the synthetic datasets. We argue that the concatenated embedding preserves the unique identity of neighbour states. In the real-world scenario, the performance is at par with the attention-based scheme. We believe that the concatenation baseline should be a standard baseline to compare to, when designing better information aggregating modules.

4. **A Different Training Objective:** We employ a different training objective in comparison to the standard practice to train the forecasting model. As mentioned earlier, we penalize only the primary pedestrian during training. Moreover, during training, we provide the ground truth of neighbouring trajectories during the prediction period and forecast only the primary trajectory. This training scheme is different from the common practice where the neighbouring trajectories are also predicted during training [52, 77, 55]. Table 7 illustrates the effectiveness of our training objective in helping the model to learn collision avoidance better. During test time, we do *not* provide the ground truth neighbour trajectories.

6 Conclusions

In this work, we tackled the challenge of modelling social interactions between pedestrians in crowds. While modelling social interactions is a central issue in human trajectory forecasting, the literature lacks a definitive comparison between the many existing interaction models on identical grounds. We presented an in-depth analysis of the design of interaction modules proposed in the literature and developed a simple yet powerful method DirectConcat, which serves two advantages: (1) it retains

the uniqueness of neighbouring pedestrians, and (2) the recurrent modelling of interactions helps to better model interactions.

A significant yet missing component in this field is an objective and informative evaluation of these interaction-based methods. To solve this issue, we propose *TrajNet++*: (1) TrajNet++ is interaction-centric as it largely comprises scenes where interactions take place thanks to our defined trajectory categorization, both in the real world and synthetic settings, (2) TrajNet++ provides an extensive evaluation system that includes novel collision-based metrics that can help measure the *physical feasibility* of model predictions. The superior quality of TrajNet++ is highlighted by the improved performance of interaction-based models on real world datasets on all metrics (4 of the top 5 methods on TrajNet [85], an earlier benchmark, do not model social interactions). Further, we demonstrated how our collision-based metrics provide a more concrete picture regarding the model performance.

DirectConcat, our method built from simple principles, outperforms competitive baselines on TrajNet++ synthetic dataset by benchmarking against several popular interaction module designs in the field. On the real dataset, there is no clear winner amongst all the designs, when compared on equal grounds. There is room for improvement, and we hope that our benchmark facilitates researchers to objectively and easily compare their methods against existing works so that the quality of trajectory forecasting models can keep increasing, allowing us to tackle more challenging scenarios.

References

[1] Bin Jiang. Simped: simulating pedestrian flows in a virtual urban environment. 1999.

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

[3] Stephen Bitgood. An analysis of visitor circulation: Movement patterns and the general value principle. *Curator: The Museum Journal*, 49:463–475, 2006.

[4] Andreas Horni, Kai Nagel, and Kay W. Axhausen. The multi-agent transport simulation matsim. 2016.

[5] Ramin Mehran, Alexis Oyama, and M. A. Hadi Shah. Abnormal crowd behavior detection using social force model. In *CVPR*, 2009.

[6] Dirk Helbing, Illés J. Farkas, Peter Molnar, and Tamás Vicsek. Simulation of pedestrian crowds in normal and evacuation situations. 2002.

[7] Dirk Helbing, Illés J. Farkas, and Tamás Vicsek. Simulating dynamical features of escape panic. *Nature*, 407:487–490, 2000.

[8] Xiaoping Zheng, Tingkuan Zhong, and Mengting Liu. Modeling crowd evacuation of a building based on seven methodological approaches. 2009.

[9] Mehdi Moussaïd, Dirk Helbing, and Guy Theraulaz. How simple rules determine pedestrian behavior and crowd disasters. *Proceedings of the National Academy of Sciences*, 108:6884 – 6888, 2011.

[10] https://storage.googleapis.com/sdc-prod/v1/safety-report/safety%20report%202018.pdf.

[11] https://uber.app.box.com/v/uberatgsafetyreport.

[12] Changan Chen, Yuejiang Liu, Sven Kreiss, and Alexandre Alahi. Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning. *2019 International Conference on Robotics and Automation (ICRA)*, pages 6015–6022, 2019.

[13] Dirk Helbing and Peter Molnar. Social force model for pedestrian dynamics. *Physical Review E*, 51, 05 1998.

[14] Gianluca Antonini and Michel Bierlaire. Discrete choice models for pedestrian walking behavior. 2006.

[15] Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. *2009 IEEE 12th International Conference on Computer Vision*, pages 261–268, 2009.

[16] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 961–971, 2016.

[17] Andreas Møgelmose, Mohan Manubhai Trivedi, and Thomas B. Moeslund. Trajectory analysis and prediction for improved pedestrian safety: Integrated framework and evaluations. *2015 IEEE Intelligent Vehicles Symposium (IV)*, pages 330–335, 2015.

[18] Ashraf Elnagar. Prediction of moving objects in dynamic environments using kalman filters. *Proceedings 2001 IEEE International Symposium on Computational Intelligence in Robotics and Automation (Cat. No.01EX515)*, pages 414–419, 2001.

[19] Ashraf Elnagar and K. Gupta. Motion prediction of moving objects based on autoregressive model. *IEEE Trans. Systems, Man, and Cybernetics, Part A*, 28:803–810, 1998.

[20] Yizheng Cai, Nando de Freitas, and James J. Little. Robust visual tracking for multiple targets. In *ECCV*, 2006.

[21] Jos Elfring, René van de Molengraft, and Maarten Steinbuch. Learning intentions for improved human motion prediction. *Robotics and Autonomous Systems*, 62:591–602, 2013.

[22] Andrey Rudenko, Luigi Palmieri, Achim J. Lilienthal, and Kai Oliver Arras. Human motion prediction under social grouping constraints. *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3358–3364, 2018.

[23] Andrey Rudenko, Luigi Palmieri, and Kai Oliver Arras. Joint long-term prediction of human motion using a planning-based social force approach. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–7, 2018.

[24] Matthias Luber, Johannes Andreas Stork, Gian Diego Tipaldi, and Kai Oliver Arras. People tracking with human motion predictions from social forces. *2010 IEEE International Conference on Robotics and Automation*, pages 464–469, 2010.

[25] Stefano Pellegrini, Andreas Ess, and Luc Van Gool. Improving data association by joint modeling of pedestrian trajectories and groupings. In *ECCV*, 2010.

[26] Laura Leal-Taixé, Michele Fenzi, Alina Kuznetsova, Bodo Rosenhahn, and Silvio Savarese. Learning an image-based motion context for multiple people tracking. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3542–3549, 2014.

[27] Wongun Choi and Silvio Savarese. A unified framework for multi-target tracking and collective activity recognition. In *ECCV*, 2012.

[28] Wongun Choi and Silvio Savarese. Understanding collective activitiesof people from videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36:1242–1257, 2014.

[29] Jur P. van den Berg, Ming C. Lin, and Dinesh Manocha. Reciprocal velocity obstacles for real-time multi-agent navigation. *2008 IEEE International Conference on Robotics and Automation*, pages 1928–1935, 2008.

[30] Gianluca Antonini, Michel Bierlaire, and Mats Weber. Discrete choice models for pedestrian walking behavior. *Transportation Research Part B: Methodological*, 40:667–687, 09 2006.

[31] Adrien Treuille, Seth Cooper, and Zoran Popovi263. Continuum crowds. In *SIGGRAPH '06*, 2006.

[32] Christopher Tay Meng Keat and Christian Laugier. Modelling smooth paths using gaussian processes. In *FSR*, 2007.

[33] Kihwan Kim, Dongryeol Lee, and Irfan A. Essa. Gaussian process regression flow for analysis of motion trajectories. *2011 International Conference on Computer Vision*, pages 1164–1171, 2011.

[34] Christoph Schöller, Vincent Aravantinos, Florian Lay, and Alois Knoll. What the constant velocity model can teach us about pedestrian motion prediction. *IEEE Robotics and Automation Letters*, 5:1696–1703, 2020.

[35] Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Learning social etiquette: Human trajectory understanding in crowded scenes. In *ECCV*, 2016.

[36] Alexandre Alahi, Vignesh Ramanathan, and Li Fei-Fei. Socially-aware large-scale crowd forecasting. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2211–2218, 2014.

[37] Shuai Yi, Hongsheng Li, and Xiaogang Wang. Understanding pedestrian behaviors from stationary crowd groups. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3488–3496, 2015.

[38] Alex Graves, Abdel rahman Mohamed, and Geoffrey E. Hinton. Speech recognition with deep recurrent neural networks. *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6645–6649, 2013.

[39] Fandong Meng. Neural machine translation by jointly learning to align and translate. 2014.

[40] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3156–3164, 2015.

[41] Chunshui Cao, Xianming Liu, Yi Yang, Yinan Yu, Jiang Wang, Zilei Wang, Yongzhen Huang, Liang Wang, Chang Huang, Wei Xu, Deva Ramanan, and Thomas S. Huang. Look and think twice: Capturing top-down visual attention with feedback convolutional neural networks. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2956–2964, 2015.

[42] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9:1735–1780, 1997.

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

[44] Namhoon Lee, Wongun Choi, Paul Vernaza, Christopher Bongsoo Choy, Philip H. S. Torr, and Manmohan Krishna Chandraker. Desire: Distant future prediction in dynamic scenes with interacting agents. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2165–2174, 2017.

[45] Federico Bartoli, Giuseppe Lisanti, Lamberto Ballan, and Alberto Del Bimbo. Context-aware trajectory prediction. *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 1941–1946, 2018.

[46] Hao Xue, Du Q. Huynh, and Mark Reynolds. Ss-lstm: A hierarchical lstm model for pedestrian trajectory prediction. *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1186–1194, 2018.

[47] Tianyang Zhao, Yifei Xu, Mathew Monfort, Wongun Choi, Chris Baker, Yibiao Zhao, Yizhou Wang, and Ying Nian Wu. Multi-agent tensor fusion for contextual trajectory prediction. In *CVPR*, 2019.

[48] Matteo Lisotto, Pasquale Coscia, and Lamberto Ballan. Social and scene-aware trajectory prediction in crowded spaces. *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 2567–2574, 2019.

[49] Mark Pfeiffer, Giuseppe Paolo, Hannes Sommer, Juan I. Nieto, Roland Siegwart, and Cesar Cadena. A data-driven model for interaction-aware pedestrian motion prediction in object cluttered environments. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–8, 2017.

[50] Xiaodan Shi, Xiaowei Shao, Zhiling Guo, Guangming Wu, Haoran Zhang, and Ryosuke Shibasaki. Pedestrian trajectory prediction in extremely crowded scenarios. In *Sensors*, 2019.

[51] Niccoló Bisagno, B. O. Zhang, and Nicola Conci. Group lstm: Group trajectory prediction in crowded scenarios. In *ECCV Workshops*, 2018.

[52] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2255–2264, 2018.

[53] Pu Zhang, Wanli Ouyang, Pengfei Zhang, Jianru Xue, and Nanning Zheng. Sr-lstm: State refinement for lstm towards pedestrian trajectory prediction. *ArXiv*, abs/1903.02793, 2019.

[54] Yanliang Zhu, Deheng Qian, Dongchun Ren, and Huaxia Xia. Starnet: Pedestrian trajectory prediction using deep neural network in star topology. *ArXiv*, abs/1906.01797, 2019.

[55] Boris Ivanovic and Marco Pavone. The trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs. 2018.

[56] Junwei Liang, Lu Jiang, Juan Carlos Niebles, Alexander G. Hauptmann, and Li Fei-Fei. Peeking into the future: Predicting future person activities and locations in videos. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5718–5727, 2019.

[57] Yanyu Xu, Zhixin Piao, and Shenghua Gao. Encoding crowd interaction with deep neural network for pedestrian trajectory prediction. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5275–5284, 2018.

[58] Tharindu Fernando, Simon Denman, Sridha Sridharan, and Clinton Fookes. Soft + hardwired attention: An lstm framework for human trajectory prediction and abnormal event detection. *Neural networks : the official journal of the International Neural Network Society*, 108:466–478, 2018.

[59] Jiachen Li, Hengbo Ma, Zhihao Zhang, and Masayoshi Tomizuka. Social-wagdat: Interaction-aware trajectory prediction via wasserstein graph double-attention network. *ArXiv*, abs/2002.06241, 2020.

[60] Amir Sadeghian, Vineet Kosaraju, Ali Sadeghian, Noriaki Hirose, and Silvio Savarese. Sophie: An attentive gan for predicting paths compliant to social and physical constraints. *CoRR*, abs/1806.01482, 2018.

[61] Vineet Kosaraju, Amir Sadeghian, Roberto Martín-Martín, Ian D. Reid, Seyed Hamid Rezatofighi, and Silvio Savarese. Social-bigat: Multimodal trajectory forecasting using bicycle-gan and graph attention networks. *ArXiv*, abs/1907.03395, 2019.

[62] Javad Amirian, Jean-Bernard Hayet, and Julien Pettré. Social ways: Learning multi-modal distributions of pedestrian trajectories with gans. *ArXiv*, abs/1904.09507, 2019.

[63] Tharindu Fernando, Simon Denman, Sridha Sridharan, and Clinton Fookes. Gd-gan: Generative adversarial networks for trajectory prediction and group detection in crowds. *ArXiv*, abs/1812.07667, 2018.

[64] Sirin Haddad, Meiqing Wu, He Wei, and Siew Kei Lam. Situation-aware pedestrian trajectory prediction with spatio-temporal attention model. *ArXiv*, abs/1902.05437, 2019.

[65] Yingfan Huang, Huikun Bi, Zhaoxin Li, Tianlu Mao, and Zhaoqi Wang. Stgat: Modeling spatial-temporal interactions for human trajectory prediction. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6271–6280, 2019.

[66] Abduallah A. Mohamed, Kun Qian, Mohamed Elhoseiny, and Christian G. Claudel. Social-stgcnn: A social spatio-temporal graph convolutional neural network for human trajectory prediction. *ArXiv*, abs/2002.11927, 2020.

[67] Jiachen Li, Fan Yang, Masayoshi Tomizuka, and Chiho Choi. Evolvegraph: Heterogeneous multi-agent multi-modal trajectory prediction with evolving interaction graphs. *ArXiv*, abs/2003.13924, 2020.

[68] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017.

[69] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014.

[70] Irtiza Hasan, Francesco Setti, Theodore Tsesmelis, Alessio Del Bue, Fabio Galasso, and Marco Cristani. Mx-lstm: Mixing tracklets and vislets to jointly forecast trajectories and head poses. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6067–6076, 2018.

[71] Irtiza Hasan, Francesco Setti, Theodore Tsesmelis, Alessio Del Bue, Marco Cristani, and Fabio Galasso. "seeing is believing": Pedestrian trajectory forecasting using visual frustum of attention. *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1178–1185, 2018.

[72] Li Sun, Zhi Yan, Sergi Molina Mellado, Marc Hanheide, and Tom Duckett. 3dof pedestrian trajectory prediction learned from long-term autonomous mobile robot deployment data. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–7, 2017.

[73] Wei-Chiu Ma, De-An Huang, Namhoon Lee, and Kris M. Kitani. Forecasting interactive dynamics of pedestrians with fictitious play. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4636–4644, 2017.

[74] Noha Radwan, Abhinav Valada, and Wolfram Burgard. Multimodal interaction-aware motion prediction for autonomous street crossing. *ArXiv*, abs/1808.06887, 2018.

[75] Francesco Giuliari, Irtiza Hasan, Marco Cristani, and Fabio Galasso. Transformer networks for trajectory forecasting. *ArXiv*, abs/2003.08111, 2020.

[76] Andrey Rudenko, Luigi Palmieri, Serge Herman, Kris M. Kitani, Dariu M. Gavrila, and Kai Oliver Arras. Human motion trajectory prediction: A survey. *ArXiv*, abs/1905.06113, 2019.

[77] Anirudh Vemula, Katharina Muelling, and Jean Oh. Social attention: Modeling attention in human crowds. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–7, 2017.

[78] Alex Graves. Generating sequences with recurrent neural networks. *CoRR*, abs/1308.0850, 2013.

[79] Emanuel Parzen. On estimation of a probability density function and mode. 1962.

[80] Tatjana Chavdarova, Pierre Baqué, Stéphane Bouquet, Andrii Maksai, Cijo Jose, Timur M. Bagautdinov, Louis Lettry, Pascal Fua, Luc Van Gool, and Franois Fleuret. Wildtrack: A multi-camera hd dataset for dense unscripted pedestrian detection. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5030–5039, 2018.

[81] N Edward. Does the flap of a butterfly's wings in brazil set off a tornado in texas. 1972.

[82] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

[83] Hao Cheng, Wentong Liao, Michael Ying Yang, Bodo Rosenhahn, and Monika Sester. Amenet: Attentive maps encoder network for trajectory prediction. 2020.

[84] Yanliang Zhu, Dongchun Ren, Mingyu Fan, Deheng Qian, Xin Li, and Huaxia Xia. Robust trajectory forecasting for multiple intelligent agents in dynamic scene. *ArXiv*, abs/2005.13133, 2020.

[85] Amir Sadeghian, Vineet Kosaraju, Agrim Gupta, Silvio Savarese, and Alexandre Alahi. Trajnet: Towards a benchmark for human trajectory prediction. *arXiv preprint*, 2018.