Human Trajectory Forecasting in Crowds: A Deep Learning Perspective

Parth Kothari, Sven Kreiss, and Alexandre Alahi

***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, deployment of intelligent transport systems, traffic operations, 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 learn 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. We propose two domain-knowledge inspired data-driven methods to effectively capture 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.**

***Index Terms*— Pedestrians, trajectory forecasting, deep learning, social interactions.**

I. Introduction

HUMANS possess the natural ability to navigate in social environments. In other words, we have understood the social etiquette of human motion like respecting personal space, yielding right-of-way, 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]–[3], traffic operations [4], crowd abnormality detection systems [5], evacuation situation analysis [6]–[10], deployment of intelligent transport systems [11]–[14] 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 movement of the surrounding people, which conform to common social norms. We refer to this task of forecasting the human motion as *human trajectory forecasting.*

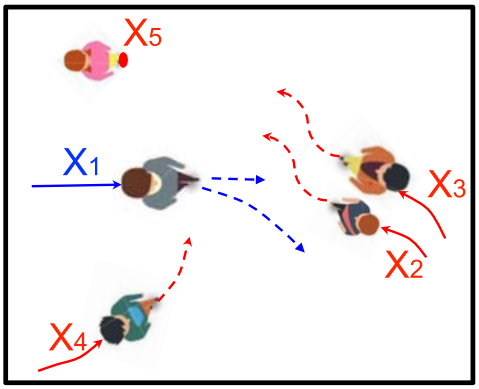


Fig. 1. Human trajectory forecasting is the task of forecasting the future trajectories (dashed) of all humans which conform to the social norms, given the past observed scene (solid). The presence of social interactions distinguish human trajectory forecasting from other sequence modelling tasks: the primary pedestrian (X1) deviates from his direction of motion to avoid a collision, by forecasting the trajectory of the child (X2).

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. Wherever necessary, we refer to a particular pedestrian of interest in the scene as the *Primary pedestrian.* 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

Manuscript received July 24, 2020; revised January 6, 2021 and March 2, 2021; accepted March 11, 2021. This work was supported in part by Honda Research and Development Company, Ltd., EPFL Open Science Fund and in part by the Swiss NSF Spark Fund under Grant 190677. The Associate Editor for this article was A. Y. Lam. *(Corresponding author: Parth Kothari.)*

The authors are with Ecole Polytechnique Federale de Lausanne (EPFL), 1015 Lausanne, Switzerland (e-mail: parth.kothari@epfl.ch).

This article has supplementary downloadable material available at https://doi.org/10.1109/TITS.2021.3069362, provided by the authors.

Digital Object Identifier 10.1109/TITS.2021.3069362

to model the long-term dependencies 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 requirement 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.

Our objective is to encode the observed scene into a representation that captures all information necessary to forecast human motion. To focus on learning the social interactions that affect human motion, we assume that there do not exist any physical constraints in our scenes. The future trajectory of a human can also be affected by his/her long-term goal, which cannot always be observed or inferred. We therefore focus on *short-term* human trajectory forecasting (next 5 secs).

Following the success of Social LSTM [15], a variety of neural networks (NN) based modules that model social interactions have been proposed in literature. In this work, we explicitly focus on the design of these interaction modules and not the entire forecasting model. The challenge in designing these interaction modules lies in handling a variable number of neighbours and modelling how they collectively influence one's future trajectory. We present a high-level pipeline encompassing most of the existing designs of interaction modules. Based on our taxonomy, we propose two novel modules which incorporate domain knowledge into the NN-based pipeline. As a consequence, these modules are better equipped to learn social etiquettes like collision avoidance and leader-follower. A long-standing question in NN-based trajectory forecasting models is to explore techniques that help to explain the model decisions. In this work, we propose to utilize Layer-wise Relevance Propagation (LRP) [16] to explain the decisions of our trajectory forecasting models. To the best of our knowledge, this is the first work that applies LRP, in a *regression* setting, to infer inter-sequence (neighbours) effects on the model output.

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 a 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. We demonstrate the efficacy of our proposed methods on TrajNet++, in comparison to various interaction encoder designs. Furthermore, we illustrate how the decisions of our proposed model architecture can be explained using LRP in real-world scenarios.

To summarize, our main contributions are as follows:

1) We provide an in-depth analysis of existing designs of NN-based interaction encoders along with their source code. We explain the decision-making of trajectory forecasting models by extending layer-wise relevance propagation to the regression setting of trajectory forecasting.
2) We propose two NN-based novel methods driven by domain knowledge for capturing social interactions.
3) We present TrajNet++, a large scale interaction-centric trajectory forecasting benchmark with novel evaluation metrics that quantify the *physical feasibility* of a model.

II. 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 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 human-human interactions to obtain the social representation.

With a specific focus on pedestrian path forecasting problem, Helbing and Molnar [17] presented a force-based motion model with 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), 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 [18]–[21] and activity forecasting [22], [23]. Burstedde *et al.* [24] utilize the cellular automaton model, another type of microscopic model, for predicted pedestrian motion. In their model, the environment is divided into uniformly distributed grids and each pedestrian has a matrix of preference to determine the transition to neighbouring cells. The matrix of preference is determined by the pedestrian's own intent along with the locations of surrounding agents. Similar to social force, the cellular automaton model has been extended over the years for improved trajectory forecasting [25]. Another prominent model for simulating human motion is Reciprocal Velocity Obstacles (RVO) [26], which guarantees safe and oscillation-free motion, assuming that each agent follows

identical collision avoidance reasoning. Social interaction modelling has been further approached from different modelling perspectives such as Discrete Choice framework [27], continuum dynamics [28] and Gaussian processes [29]–[31]. Robicquet *et al.* [32] defined social sensitivity to characterize human motion into different navigation styles. Alahi *et al.* [33], [34] defined Social Affinity Maps to link broken or unobserved trajectories to forecast pedestrian destinations. Yi *et al.* [35] 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 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 [36]–[39], Alahi *et al.* [15] proposed Social LSTM, the first NN-based model for human trajectory forecasting. Social LSTM is an Long-Short Term Memory network (LSTM) [40] 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 [41]–[47] and various other designs of NN-based interaction module have been proposed [48]–[61]. Pfieffer *et al.* [48] 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. While modelling social interactions, Hasan *et al.* [59], [60] based on domain knowledge, only consider the pedestrians in the visual frustum of attention [62]. 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] and Salzmann *et al.* [61] 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. [57], [58] propose to input the relative position and relative velocity of k nearest neighbours directly to a Multi-Layer Perceptron (MLP) to obtain the interaction vector. Many works [63]–[77] propose interaction module designs based on attention mechanisms [78], [79] 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). For an extensive survey of all human forecasting methods capturing both social and physical interactions, one can refer to Rudenko *et al.* [80].

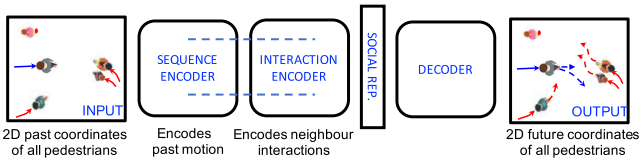


Fig. 2. A data-driven pipeline for human trajectory forecasting. We focus on the design choices for the interaction module.

III. 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 and velocity of pedestrian i at time-step t is denoted by $\mathbf{x}_i^t = (x_i^t, y_i^t)$ and $\mathbf{v}_i^t$ respectively. We receive the positions of all pedestrians at time-steps $t = 1, \ldots, T_{obs}$ and want to forecast the future positions from time-steps $t = T_{obs+1}$ to T_{pred}. We denote our predictions using $\hat{\boldsymbol{Y}}$.

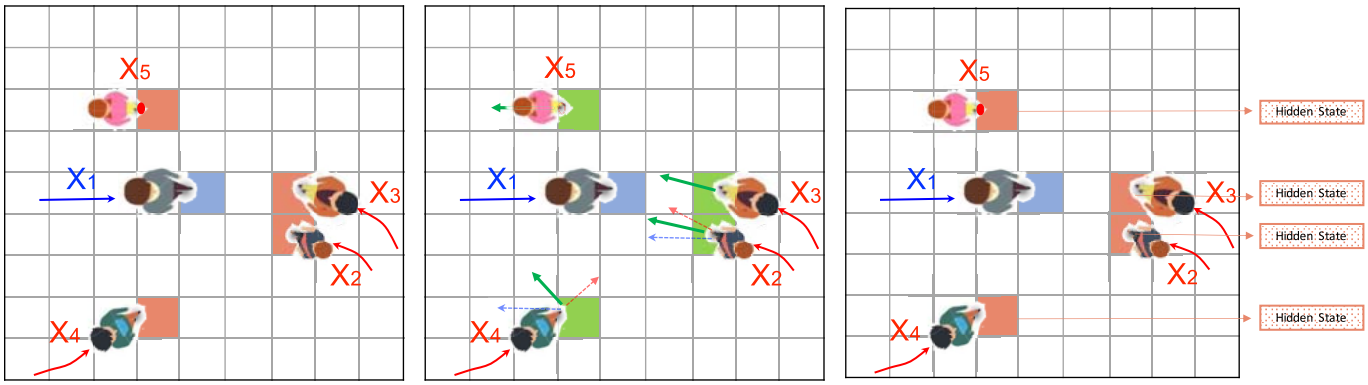
At time-step t, we denote the state of pedestrian i by $\mathbf{s}_i^t$. The state can refer to different attributes of the person, *e.g.*, the position concatenated with velocity ($\mathbf{s}_i^t = [\mathbf{x}_i^t, \mathbf{v}_i^t]$). The problem statement can be extended to take as input more attributes at each time-step, *e.g.*, the body pose, as well as predicting k most-likely future trajectories.

IV. 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. Since the objective of our work is to model human social interactions, we focus on investigating the design choices for the interaction module.

A. Interaction Module

Humans have the capability to navigate with ease in complex, crowded environments by following unspoken social rules, which result in social interactions. In recent years, these social interactions are captured effectively by designing novel interaction modules. In this section, we broadly categorize the different data-driven interaction encoders studied in literature, based on their underlying components. We show how most of these designs fall within our categorization. Following this, in the experimental section we empirically analyze the



(a) Occupancy pooling [15] (b) Directional pooling [Ours] (c) Social pooling [15]

Fig. 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. The constructed grid tensors are passed through an MLP-based neural network to obtain the interaction vector.

effectiveness of each of these components and provide recommendations for designing improved interaction modules. The existing designs can be broadly categorized into (1) **Grid based** and (2) **Non-Grid based**. We now discuss in detail the different components of these interaction encoders.

1) Grid Based Interaction Models: In grid-based models, the interaction module takes as input a local grid constructed around the pedestrian of interest, the primary pedestrian. Each cell within the grid represents a particular spatial position relative to the primary pedestrian. The design of grid-based models largely differ based on neighbour input state representation.

Neighbour Input State: Consider an $N_o \times N_o$ grid around the primary pedestrian, where each cell contains information about neighbours located in that corresponding position. Existing designs provides the information of the neighbours in two main forms: (a) *Occupancy Pooling* [15], [44] where each cell in the grid indicates the presence of a neighbour (see Fig 3a) (b) *Social Pooling* [15], [42]–[44], [46], [47], [51] where each cell contains the entire past history of the neighbour, represented by, *e.g.*, the LSTM hidden state of the neighbours (see Fig 3c). The obtained grid is flattened and subsequently embedded using an MLP to get the interaction vector p_i^t.

Directional pooling: In this work, based on our domain knowledge, we propose to take as input the relative velocity of each neighbour in the corresponding grid cell. When humans navigate in crowded environments, in addition to relative positions of the neighbours, they naturally tend to focus on the neighbours' relative velocities. For the same positional configuration, the relative velocities of neighbours lead to the concepts of leader-follower and collision avoidance *i.e.,* one exhibits leader-follower and accelerates when the neighbour is in front and walking along the same direction, while the same positional configuration leads to deceleration when the neighbour moves in the opposite direction. Having access to relative velocities can therefore significantly reduce model prediction collisions.

Furthermore, due to the complex nature of real-world movements combined with the possibility of noisy measurements, the current design of social pooling [15] can sometimes fail to learn the important notion of preventing collisions. One reason lies in the fact that the models are trained to minimize the displacement errors [15], [67] and not collisions. The models are expected to learn the notion of collision avoidance implicitly. By focusing explicitly on relative velocity configurations, we can obtain more domain-knowledge driven control over the design of the interaction encoder. When the model explicitly focuses only on relative velocity configuration (rather than abstract hidden-state configurations), which is sufficient to learn concepts of leader-follower and collision avoidance, the resulting simple design has the potential to output safer predictions. Furthermore, our proposed directional pooling is computationally faster to deploy in real-time scenarios due to the reduced size of input ($N \times N \times 2$ in comparison to $N \times N \times H_{dim}$ where H_{dim} is the hidden-state dimension).

One might additionally argue to only consider the neighbours in front of the primary pedestrian as proposed in [62]. We will demonstrate in the experimental section that directional pooling implicitly learns this notion of only focusing on the neighbours in the field-of-view of the primary pedestrian.

2) Non-Grid Based Interaction Models: Non-grid based modules, 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. As illustrated in Fig 4, the design choices of these modules can be categorized based on four factors: (a) neighbour input state, (b) input state embedding, (c) neighbour information aggregation strategy, and (d) aggregated vector embedding.

a) Neighbour input state: Non-grid based methods do not contain an implicit notion of the spatial position of neighbours with respect to the primary pedestrian, unlike the grid-based counterparts. Hence, almost all the existing designs in literature take as input the relative spatial position of the neighbours. Another popular input choice is the hidden-state of the neighbouring pedestrian [52], [67] as the hidden-state has the ability to encode information regarding the motion history of the corresponding pedestrian. Amirian *et al.* [68] models the

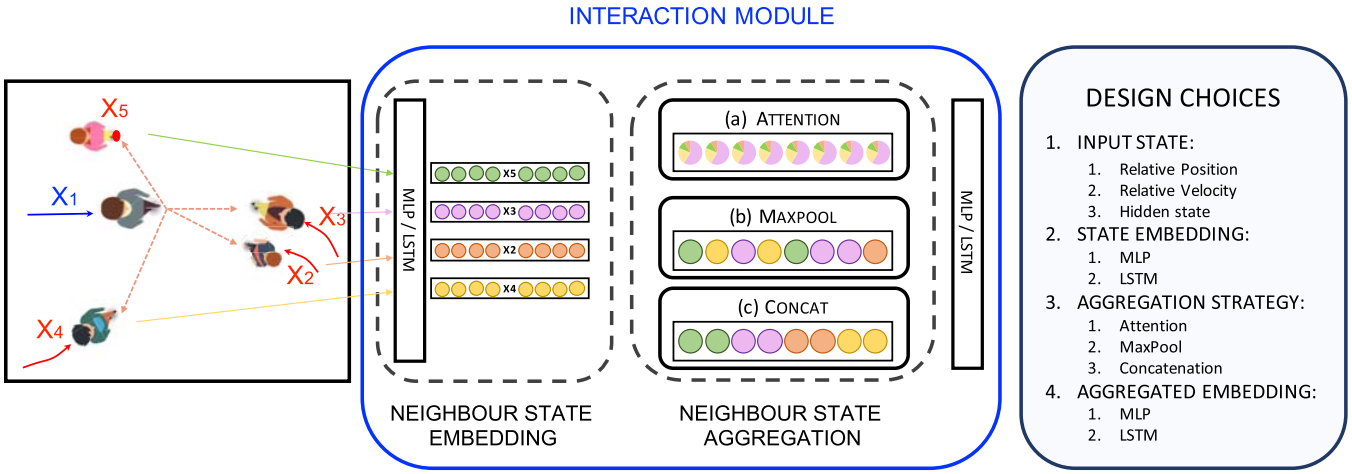


Fig. 4. Illustration of the non-grid based encoding modules to obtain the interaction vector. The challenge lies in handling a variable number of neighbours and aggregating their state information to construct the interaction vector (a) Neighbour information is aggregated via attention mechanism (b)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

neighbour states using interaction-centric geometric features like bearing angle between agents and distance of closest approach [81]. Ivanovic *et al.* [55] takes as input the velocity of neighbours. In this work, we argue that inputting *relative* velocity of neighbours is an important factor for reducing collisions in model predictions.

b) Input state embedding: The input states of the neighbours are usually embedded using an MLP. However, recent works [70], [82] based on graph neural network [83] designs, embed the input states using an LSTM. Each connection of the primary pedestrian to his neighbour is modelled using a different LSTM. The LSTM helps to capture the evolution of the neighbour states, unlike the first-order MLP.

c) Aggregation strategy: One of the most important challenges of non-grid based models is to find the ideal strategy to aggregate the information of all the neighbours. Gupta *et al.* [52] proposed to aggregate the interaction information by applying a symmetric max-pool function on the obtained neighbour state embeddings. Ivanovic *et al.* [55] and Hasan *et al.* [59] utilized the symmetric sum-pooling function.

A large body of works utilize the attention mechanism [78], [79] to determine the weights of different neighbours in predicting the future trajectory. These weights can be either hand-crafted [64] or learnt in a data-driven manner [66]–[68]. The attention mechanism can be applied multiple times to model higher-order spatial interactions [67].

A simple baseline for aggregating neighbour information is to concatenate the neighbour embeddings. To tackle the issue of handling variable number of neighbours, we investigate the performance of the concatenation scheme by selecting the top-k neighbours based on a defined criterion, (*e.g.*, euclidean distance). Despite the simplicity, we demonstrate that the concatenation strategy performs at par with its sophisticated counterparts.

d) Aggregated vector embedding: The aggregated neighbour vector is passed through an MLP, with the exception of Ivanovic *et al.* [55] that pass the sum-pooled vector through an LSTM, to obtain the interaction vector p_t^i. 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.

For brevity, the interaction modules are denoted using acronyms based on their designs. The acronyms are of the form **P-Q-R-S** where **P** denotes the input to the module, **Q** denotes the state embedding module, **R** denotes the information aggregation mechanism and **S** denotes aggregated vector embedding module. Table I illustrates how our categorization encompasses the popular designs on NN-based interaction modules in literature.

DirectConcat: Equivalent to our proposed `D-Grid`, we now describe its non-grid counterpart *DirectConcat*. Grid-based models, based on their design, implicitly consider only those neighbours that are within the grid constructed around the primary pedestrian. We argue that modelling interactions of all pedestrians (even those far away) can lead to the model learning spurious correlations. Thus, we propose to consider only the top-k neighbours closest to the primary pedestrian. We will demonstrate in the experimental section that if k is set to a large value, *i.e.* if the model considers all pedestrians in the scene, the model deteriorates in its ability to learn collision avoidance.

Similar to aggregating the obtained directional grid by flattening the obtained grid, in *DirectConcat* we propose to concatenate the relative-velocity and relative-position embeddings of top-k neighbours. This preserves the unique identity of the neighbours as compared to mixing the different embeddings like in max-pooling [52] or sum-pooling [55]. Finally, we pass the aggregated vector through an LSTM as compared to an MLP. This design choice helps to model higher-order spatio-temporal interactions better and is more robust to noise in the real-world measurements. We demonstrate in the experimental section that indeed the LSTM embedding helps to improve the collision metric. By design, *DirectConcat* falls under the `D-MLP-ConC-LSTM` architecture of our categorization. We will use the terms *DirectConcat* and `D-MLP-ConC-LSTM` interchangeably.

TABLE I

MODEL ACRONYMS: ACRONYMS FOR THE VARIOUS DESIGNS OF INTERACTION MODULES. WE OBSERVE THAT MOST OF THE EXISTING INTERACTION ENCODER DESIGNS FALL UNDER OUR DEFINED CATEGORIZATION

Acronym (P-Q-R-S)	Input (P)	Embed-I (Q)	Aggreg. (R)	Embed-II (S)	References
O-Grid	Position	None	Grid	MLP	**O-LSTM** [15], [44], [48]
S-Grid	H-State	None	Grid	MLP	**S-LSTM** [15], [44], [46], [51], [42], [47], [43], [59]
D-Grid	Velocity	None	Grid	MLP	**Directional Pooling** [**Ours**]
D-MLP-Attn-MLP	Velocity	MLP	Attn	MLP	[50]
S-MLP-Attn-MLP	H-State	MLP	Attn	MLP	**S-BiGAT** [67], [68], [64], [66], [53], [63], [65], [71], [75]
S-MLP-MaxP-MLP	H-State	MLP	MaxPool	MLP	**S-GAN** [52]
D-MLP-ConC-MLP	Velocity	MLP	Concat	MLP	[57], [58]
D-MLP-SumP-LSTM	Velocity	MLP	SumPool	LSTM	**Trajectron** [55] [1]
O-LSTM-Att-MLP	Position	LSTM	Attn	MLP	**S-Attn** [82], [70]
D-MLP-ConC-LSTM	Velocity	MLP	Concat	LSTM	**DirectConcat** [**Ours**]

B. 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 encoder to be an LSTM due to its capability to handle varying input length and capture long-term dependencies. Moreover, most works utilize 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 subsection. 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:

$$e_i^t = \phi(\mathbf{v}_i^t; W_{emb}), \tag{1}$$

$$h_i^t = LSTM(h_i^{t-1}, [e_i^t; p_i^t]; W_{encoder}), \tag{2}$$

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 [84], 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}:

$$[\mu_i^t, \sigma_i^t, \rho_i^t] = \phi_{dec}(h_i^{t-1}, W_{dec}), \tag{3}$$

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:

$$\mathcal{L}_i(w) = -\sum_{t=T_{obs}+1}^{T_{pred}} \log(\mathbb{P}(\mathbf{v}_i^t | \mu_i^t, \sigma_i^t, \rho_i^t)). \tag{4}$$

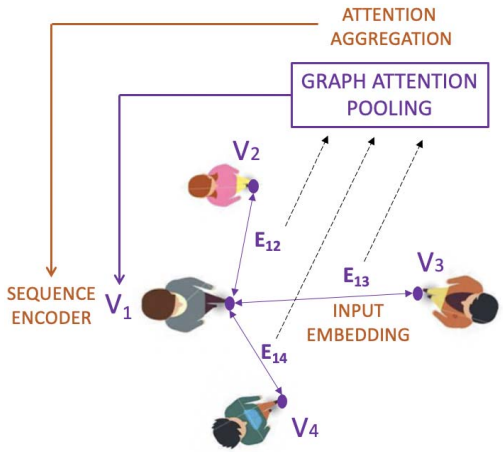
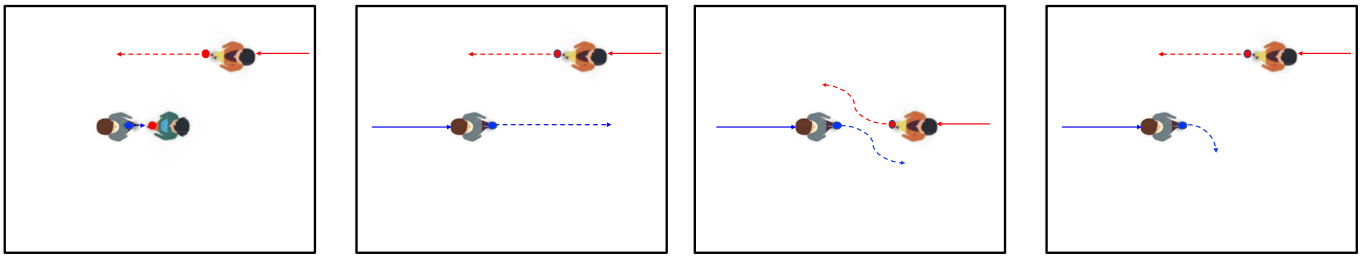


Fig. 5. Illustration of Graph neural networks (purple) as a special case of our data-driven pipeline (brown). Each vertex V_i is modelled using a sequence encoder, the neighbour edges E_{ij} correspond to our input embeddings which are aggregated via attention mechanism. The resulting interaction vector is provided as input to the sequence encoder (vertex V_i).

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 in each scene of the training dataset. We will demonstrate how this training procedure helps the model to 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.

1) Equivalence to Graph Neural Networks: Recently, graph neural networks (GNNs) have become popular for forecasting human motion. In the GNN setup, each pedestrian is represented as a node/vertex V_i and two interacting pedestrians are connected via an edge E_{ij}. V_i models the sequence representation of the associated pedestrian and edge E_{ij} updates according to the interactions between the associated pedestrians. We show an equivalence between dynamic-interaction-based GNNs and our proposed LSTM-based pipeline with `S-X-Attn-MLP` (where `X` $\in$ {`MLP`, `LSTM`}) interaction encoding scheme, visually illustrated in Figure 5. Without loss of generality, let pedestrian i be the primary pedestrian. Vertex



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

Fig. 6. Visualization of our high-level defined trajectory categories.

V_i is modelled using an LSTM sequence encoder. Edge E_{ij} takes as input the state of the neighbours and updates over time using an MLP or an LSTM (*input state embedding*). At each time-step, the information of all connected edges is aggregated using attention mechanism (*aggregation strategy*), popularly referred to as graph attention (GAT) pooling [85] in GNN literature. Finally the aggregated vector is optionally passed through an MLP to obtain the interaction vector p_i which is the input to the LSTM sequence encoder for V_i. Social-BiGAT [67] utilizes the `S-MLP-Attn-MLP` design, Social Attention [82] utilizes the `O-LSTM-Attn-MLP` design while recently, STAR [75] utilizes the `S-MLP-Attn-MLP` design with the sequence encoder for vertex V_i being a Transformer [78].

C. Explaining Trajectory Forecasting Models

Trajectory forecasting models are deployed in many safety-critical applications like autonomous systems. In such scenarios, it becomes really important to gain insight into the decision-making of the 'blackbox' neural networks. Several works in literature attempt to explain the rationale behind the NN decisions [16], [86]–[89]. Out of these techniques, Layer-wise Relevance Propagation (LRP) is one of the most prominent methods in explainable machine learning.

LRP re-distributes the model output decision back to each of the input variables indicating the extent to which each input contributes to the output. This is done by reverse-propagating the model prediction through the network by means of heuristic rules that apply to each layer of a neural network [16]. These propagation rules are based on a local conservation principle: the net quantity or relevance, received by any higher layer neuron is redistributed in the same amount to neurons of the layer below. Mathematically, if j and k are indices for neurons in two consecutive layers, and denoting by $R_{j\to k}$ the relevance flowing between two neurons, we have the equations:

$$\Sigma_j R_{j\to k} = R_k \tag{5}$$

$$R_j = \Sigma_k R_{j\to k} \tag{6}$$

On applying the local conservation principle across all the layers, we obtain global conservation of the output score when reverse propagated back to the inputs. Recently, Arras *et al.* [90] have demonstrated that the principle of LRP can also be applied to LSTMs.

LRP has largely been explored in the domain of model classification *i.e.* the outputs are classification scores. In this work, we utilize LRP to determine on which neighbours (via the input interaction vector) and past velocities (via the input velocity embedding) of the primary pedestrian our model focuses on, when regressing to the next predicted velocity. We achieve this by reverse-propagating both the x-component v_x as well as y-component v_y of predicted velocity ($\mathbf{v_{pred}} = (v_x, v_y)$) and adding the obtained input relevance scores. To the best of our knowledge, we are the first work to empirically demonstrate that LRP provides reasonable explanations when extended to the regression task of trajectory forecasting. Moreover, the LRP technique is generic and can be applied on top of any trajectory forecasting network to analyze its predictions.

V. TrajNet++: A Trajectory Forecasting Benchmark

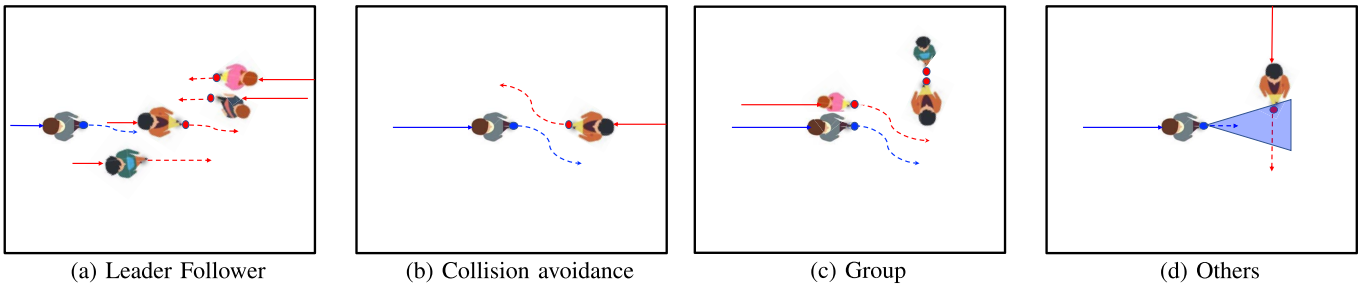
In this section, we present *TrajNet++*, our interaction-centric human trajectory forecasting benchmark. To demonstrate the efficacy of a trajectory forecasting model, the standard practice is to evaluate these models against baselines on a standard benchmark. However, current methods have been evaluated 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.

A. Trajectory Categorization

We provide a detailed trajectory categorization (Fig 8). 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.

We categorize each scene with respect to its corresponding pedestrian of interest, the *primary pedestrian*. We now explain in detail our proposed hierarchy for trajectory categorization. We also provide example scenarios for the same in Fig 6:

1) **Static (Type I)**: If the euclidean displacement of the primary pedestrian in the scene is less than a specific threshold.
2) **Linear (Type II)**: If the trajectory of the primary pedestrian can be *correctly forecasted* with the help of an



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

Fig. 7. Visualization of our Type III interactions commonly occurring in real world crowds.

Extended Kalman Filter (EKF). A trajectory is said to be *correctly forecasted* by EKF if the final displacement error 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).

3) **Interacting (Type III)**: These correspond to scenes where the primary pedestrian undergoes social interactions. For a detailed categorization coherent with commonly observed social interactions, we divide interacting trajectories into the following sub-categories (see Fig 7).

(a) **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.

(b) **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 is involved in collision avoidance.

(c) **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 the entire scene.

(d) **Other Interactions (Type IIId)**: These are scenes where the primary pedestrian undergoes social interactions other than LF, CA and Group. We define *social interaction* as follows: We look at the 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.

4) **Non-Interacting (Type IV)**: If a trajectory of the primary pedestrian is non-linear and undergoes no social interactions during prediction, the scene is categorized as non-interacting.

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. The details of the categorization thresholds as well as the datasets which comprise our TrajNet++ benchmark are provided in the supplementary material. A few examples of our categorization in the real world are displayed in Fig 9. In addition to a well-sampled dataset, TrajNet++ provides an extensive evaluation system to understand model performance better.

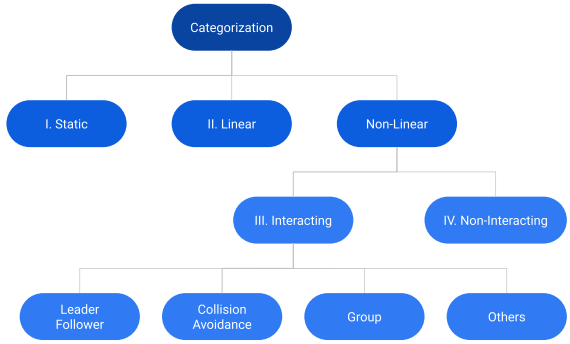


Fig. 8. Our proposed hierarchy for trajectory categorization. Using our defined trajectory categorization, we construct the TrajNet++ benchmark by sampling trajectories corresponding largely to 'Type III: Interacting' category.

B. Evaluation Metrics

1) 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 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 over all predicted time steps.
2) **Final Displacement Error (FDE)**: The L_2 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 collision-free trajectories, we propose two new collision-based metrics in our framework (see Fig 10):

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

Fig. 9. Sample scenes from our benchmark. In each of the samples, we illustrate a different social interaction between the primary pedestrian (yellow) and the corresponding interacting neighbours (red) in real world datasets.

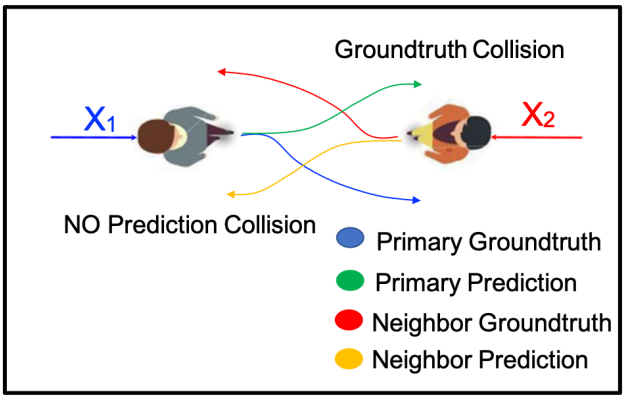


Fig. 10. Visual illustration of our proposed collision metrics. In the example, the model prediction exhibits ground-truth collision (Col-II = 1) but no prediction collision (Col-I = 0).

3) **Collision I - Prediction collision (Col-I)**: This metric calculates the percentage of collision between the primary pedestrian and the neighbors in the *predicted 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 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 the 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.

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

For the Top-k metrics, we propose k be small (3 as opposed to 20) as a model outputting 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 time-step, the authors obtain a Kernel Density Estimate (KDE) [91] of the predicted distribution. 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.

VI. Experiments

In this section, we perform extensive experimentation on both TrajNet++ synthetic and real-world datasets to understand the efficacy of various interaction module designs for human trajectory forecasting. Moreover, we demonstrate how our proposed metrics help to provide a complete picture of model performance.

A. Implementation Details

The velocity of each pedestrian is embedded into a 64-dimensional vector. The dimension of the interaction vector is 256. The dimension of the goal direction vector is 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 its own encoder and decoder. The batch size is fixed to 8. We train using ADAM optimizer [92] with a learning rate of 1e-3. We perform interaction encoding at every time-step. For the concatenation based models, we consider top-4 nearest neighbours based on euclidean distance unless stated

TABLE II

UNIMODAL COMPARISON OF INTERACTION ENCODER DESIGNS WHEN FORECASTING 12 FUTURE TIME-STEPS, GIVEN THE PREVIOUS 9 TIME-STEPS, ON TRAJNET++ SYNTHETIC DATASET. ERRORS REPORTED ARE ADE / FDE IN METERS, COL-I / COL-II REPORTED IN %. WE EMPHASIZE THAT OUR GOAL IS TO REDUCE COL-I WITHOUT COMPROMISING DISTANCE-BASED METRICS

Model (Acronym)	ADE/FDE	Col-I	Col-II
Grid based methods			
Vanilla	0.32/0.62	19.0	7.1
O-Grid [15]	0.27/0.52	11.7	4.9
S-Grid [15]	0.24/0.50	**2.2**	**4.6**
D-Grid [**Ours**]	0.24/0.49	**2.2**	4.8
Non-Grid based methods			
S-MLP-MaxP-MLP [52]	0.27/0.52	6.4	5.2
S-MLP-Attn-MLP [67]	0.26/0.52	3.7	5.4
D-MLP-SumP-LSTM [55]	0.29/0.57	13.8	6.6
O-LSTM-Attn-MLP [82]	0.24/0.48	0.8	5.2
D-MLP-MaxP-MLP	0.28/0.55	14.3	6.1
D-MLP-Attn-MLP	0.27/0.52	8.1	**5.0**
D-MLP-ConC-MLP	0.25/0.50	1.3	5.6
D-MLP-ConC-LSTM [**Ours**]	0.24/0.48	**0.6**	5.3

otherwise. For the attention aggregation strategy, we utilize the self-attention mechanism proposed in [78].

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.

B. Interaction Models: Synthetic Experiments

We utilize synthetic datasets to validate the efficacy of various interaction modules in a controlled setup. For the synthetic dataset, since ORCA (our underlying simulator) [26] has access to the goals of each pedestrian, we embed the goal direction and concatenate it to the velocity embedding (in Eq 1).

Table II quantifies the performance of the different designs of interaction modules published in the literature on TrajNet++ synthetic dataset. It is very interesting to note how our proposed Col-I metric provides a more complete picture of model performance. Observing only the distance-based metrics, one might wrongly conclude that the methods are similar in performance, however, they do not indicate the ability of the model to learn social etiquette (collision avoidance in this case). In safety-critical scenarios, it is more important for a model to prevent collisions in comparison to minimizing ADE/FDE.

1) 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 a significant difference in performance when we explicitly provide relative velocity of the neighbours as input. Further, since ORCA is a first-order trajectory simulator dependent only on relative configuration of neighbours, one can explain the performance of `D-Grid` being at par with `S-Grid` in the controlled setup.

2) Aggregation Strategy: We focus on the information aggregation strategies for non-grid based encoders. It is evident that the baseline `D-MLP-Conc-MLP` of *concatenating* the neighbourhood information performs better than the sophisticated attention-based `D-MLP-Attn-MLP` and max-pooling-based `D-MLP-MaxP-MLP` alternatives. This performance can be attributed to the simplicity of the concatenation scheme along with its property to preserve the identity of the surrounding neighbours. The MaxPooling strategy mixes up the different embeddings of the neighbours resulting in a high collision loss.

3) LSTM-Based Interaction Model: Among the non-grid LSTM-based designs, the drop in performance of `D-MLP-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. We notice that encoding the interaction information using LSTM [`O-LSTM-Att-MLP`, `D-MLP-Conc-LSTM`], improves performance over its MLP-based counterparts. MLP encoders, due to their non-recurrent nature, have no information regarding the interaction representation at the previous step. We argue that LSTMs can capture the evolution of interaction and therefore provide a better neighbourhood representation as the scene evolves.

C. Interaction Models: Real World Experiments

Now, we 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 Type III *interacting* category. 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?'

Table III provides an extensive evaluation of existing baselines on the Type III *interacting* trajectories of the TrajNet++ real dataset. We observe that Col-I metric is the differentiating factor for various model designs when compared on *identical grounds*. We hope that in future, researchers will incorporate the collision metrics while reporting their model performances on trajectory forecasting datasets. Moreover, the performance of ADE/FDE is similar (including submitted methods) indicating that there exists a lot of scope to improve the performance of current trajectory forecasting models on a well-sampled interaction-centric test set.

1) Classical Methods: We first compare with the classical trajectory forecasting models, namely, Extended Kalman Filter (EKF), Constant Velocity (CV) [95], Social Force [17], and ORCA [26]. The high error of EKF and CV can be attributed to the fact that these methods do not model social interactions. Both Social Force and ORCA models forecast the future trajectory based on the assumption that each pedestrian has

TABLE III

UNIMODAL COMPARISON OF INTERACTION ENCODER DESIGNS WHEN FORECASTING 12 FUTURE TIME-STEPS, GIVEN THE PREVIOUS 9 TIME-STEPS, ON TYPE III *interacting* TRAJECTORIES OF TRAJNET++ REAL WORLD DATASET. ERRORS REPORTED ARE ADE / FDE IN METERS, COL-I / COL-II IN MEAN % (STD. DEV. %) ACROSS 5 INDEPENDENT RUNS. WE EMPHASIZE THAT OUR GOAL IS TO REDUCE COL-I WITHOUT COMPROMISING DISTANCE-BASED METRICS

Model (Acronym)	ADE/FDE	Col-I	Col-II
Hand-crafted methods			
Kalman Filter	0.87/1.69	16.20	22.1
Constant Velocity	0.68/1.42	14.30	15.2
Social Force	0.89/1.53	**0.0**	**13.1**
ORCA	0.68/1.40	**0.0**	15.0
Top submitted methods*[2]			
AMENet [93]	0.62/1.30	14.1	16.90
AIN [94]	0.62/1.24	10.7	17.10
PecNet [76]	0.57/1.18	15.0	14.3
Grid based methods			
Vanilla	0.6/1.3	13.6 (0.2)	14.8 (0.1)
O-Grid [15]	0.58/1.24	9.1 (0.4)	15.1 (0.3)
S-Grid [15]	0.53/1.14	6.7 (0.2)	13.5 (0.5)
D-Grid [**Ours**]	0.56/1.22	**5.4 (0.3)**	**13.0 (0.5)**
Non-Grid based methods			
S-MLP-MaxP-MLP [52]	0.57/1.24	12.6 (0.9)	14.6 (0.7)
S-MLP-Att-MLP [67]	0.56/1.22	7.2 (0.8)	14.8 (0.4)
D-MLP-SumP-LSTM [55]	0.60/1.28	13.9 (0.7)	15.4 (0.5)
O-MLP-Att-LSTM [82]	0.56/1.21	9.0 (0.3)	15.2 (0.4)
D-MLP-ConC-MLP	0.58/1.23	7.6 (0.6)	14.3 (0.2)
D-MLP-MaxP-MLP	0.60/1.25	12.9 (0.6)	14.8 (0.5)
D-MLP-Attn-MLP	0.56/1.22	6.9 (0.3)	14.3 (0.6)
D-MLP-ConC-LSTM (k=8)	0.56/1.22	8.5 (0.5)	**14.0 (0.1)**
D-MLP-ConC-LSTM [**Ours**]	0.55/1.19	**6.8 (0.4)**	14.5 (0.5)

TABLE IV

SPEED (IN SECONDS) COMPARISON WITH S-GRID AT TEST-TIME. D-GRID PROVIDES 3.7X SPEEDUP AS COMPARED TO S-GRID RENDERING IT MORE SUITABLE FOR REAL-WORLD DEPLOYMENT TASKS

	Vanilla	O-Grid	S-Grid	D-Grid
Time	0.01	0.022	0.081	0.022
Speed-Up	8.1x	3.7x	1x	**3.7x**

an intended direction of motion (driven by the goal) and a preferred velocity. We interpolate the observed trajectory to identify the *virtual goals* for each agent. `Social Force` and `ORCA` are calibrated to fit the TrajNet++ training data by minimizing ADE/FDE metrics. The interaction-based NN models outperform the handcrafted models in terms of the distance-based metrics, as NN have the ability to learn the subtle and diverse social interactions.

2) Grid-Based Modules: Our proposed `D-Grid` performs superior to `O-Grid` in the real world as well. It is interesting to compare the performances of `D-Grid` and `S-Grid`. The current design of `S-Grid` fails to learn the notion of prediction collision. This reaffirms the fact that while training to minimize ADE/FDE, the hidden-state of LSTM is unable to provide representations necessary to avoid collisions. In the `D-Grid` design, we force the model to focus explicitly on relative velocities based on our domain knowledge. The simplicity of our design slightly hampers the distance-based accuracy as we limit the expressibility of the model. However, it leads to safer predictions as the task of the model to learn social concepts is made easier thanks to our domain-knowledge based design. Further, as shown in Table IV, the `D-Grid` provides significant computational speed-up in comparison to `S-Grid` rendering it useful for real-time deployment.

3) Aggregation Strategy: We evaluate the performance of various aggregation strategies [`D-MLP-Attn-MLP`, `D-MLP-MaxP-MLP`, `D-MLP-ConC-MLP`] on real-world data keeping all the other factors constant. We observe that the max-pooling strategy performs the worst due to its design to hard-merge the embeddings of various neighbours. The concatenation strategy, despite its simplicity, performs only slightly worse in comparison to its sophisticated attention-based counterpart. We believe that the concatenation baseline is a simple yet powerful baseline to compare to when designing future information aggregating modules. One interesting point to note is that `D-MLP-Attn-MLP` performs superior to its social counterpart `S-MLP-Attn-MLP` further corroborating the strength of knowledge-based modules.

4) LSTM-Based Interaction Models: Among the LSTM-based non-grid designs, `D-MLP-SumPool-LSTM` module [55] demonstrates high Col-I metric due to (1) sum pooling strategy and (2) encoding of absolute neighbour coordinates. The Col-I metric for `O-LSTM-Att-MLP` [82] is relatively higher compared to `D-MLP-Concat-LSTM` in the real-world due to the absence of relative velocity as input to the interaction model. One can notice the importance of having an LSTM-based embedding in our proposed `DirectConcat` model by comparing the performance between `D-MLP-Concat-LSTM` and `D-MLP-Concat-MLP`. This design choice helps to model higher-order spatio-temporal interactions better and is more robust to noise in the real-world measurements as LSTM controls the evolution of the interaction vector. The top-k neighbours are chosen based on euclidean distance. We argue that imposing domain knowledge by considering nearest neighbours is one of the reasons for improvement in Col-I metric as compared to its attention-based and max-pooling-based counterparts. This is corroborated by observing that considering a large number of neighbours ($k = 8$), in comparison to ($k = 4$), results in an increase in the Col-I metric.

5) Comparison to Vanilla LSTM: The interaction-based models perform superior to `Vanilla LSTM` in terms of distance-based metrics. However, an important point to discuss is the performance comparison between `Vanilla LSTM` and interaction-based models in terms of the Col-II metric. We would like to remind that performance in Col-II metric represents the cases where the model predicts the correct mode for the primary pedestrian so that the collisions with the ground-truth trajectories of neighbours is minimal. Due to the multimodal nature of real-world data, it is quite possible that the interaction model predicts a different mode for one of the pedestrians (primary or neighbour) leading to the primary pedestrian not following the ground-truth mode. Indeed, two of the current interaction models [`O-MLP-Att-LSTM`, `D-MLP-SumP-LSTM`] struggle in accurately predicting the ground-truth mode compared to `Vanilla LSTM`. However,

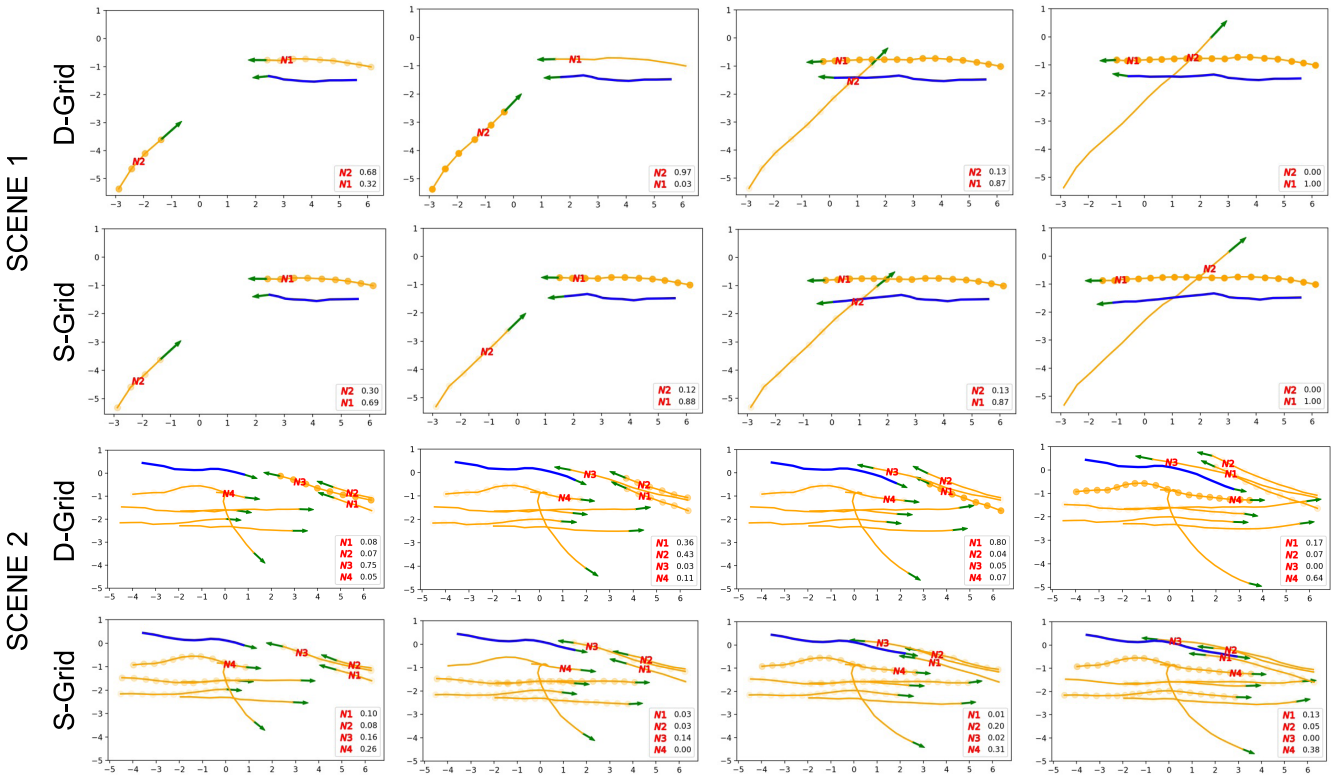


Fig. 11. Visualizing the decision-making of grid-based interaction modules using layer-wise relevance propagation. The darker the yellow circles, the more is the weight (also shown in the legend) provided by the primary pedestrian (blue) to the corresponding neighbour (yellow). Our proposed `D-Grid`, driven by domain knowledge, outputs more human-like trajectories with more intuitive focus on surrounding neighbours as compared to `S-Grid`.

this observation *does not* undermine the importance of modelling social interactions. The usefulness of modelling social interactions is justified by the Col-I metric comparison, which indicates that given the chosen mode for the primary pedestrian, the interaction models predicts a collision-free future for the entire scene, as opposed to `Vanilla LSTM`.

6) Modified Training Objective: We employ a modified training objective where we penalize only the primary pedestrian in comparison to the standard practice of penalizing all pedestrians in the scene [52], [55], [82]. In the TrajNet++ real world dataset, we know that the primary trajectories are largely interacting thanks to our defined categorization; however, there exist significant portion of trajectories among the neighbours which are static and linear. Penalizing such neighbouring trajectories during training might bias the network into learning linear and static behavior because of the resulting imbalanced distribution (caused by the neighbours).

Table V illustrates the effectiveness of our modified training objective in helping the model to learn collision avoidance better. During test time, we do *not* provide the ground truth neighbour trajectories.

7) Understanding NN Decision-Making: Now, using the popular technique of Layer-wise Relevance Propagation (LRP), we investigate how various input factors affect the decision-making of the NN at each time-step. This helps us in verifying whether the NN decision-making process follows human intuition. Fig 11 illustrates the score of each neighbour obtained on applying the LRP procedure

TABLE V
OUR PROPOSED TRAINING OBJECTIVE THAT PENALIZES ONLY THE PRIMARY PREDICTION PROVIDES SUPERIOR PERFORMANCE

Training Objective	Dataset	ADE	FDE	Col-I
Standard [52], [82], [55]	Synth	0.25	0.50	11.9
Proposed [**Ours**]	Synth	**0.24**	**0.49**	**2.2**
Standard [52], [82], [55]	Real	0.59	1.27	7.4
Proposed [**Ours**]	Real	**0.56**	**1.22**	**5.4**

on our proposed `D-Grid` module and baseline `S-Grid` in real-world scenarios.

In Scene 1, we demonstrate the application of LRP on a simple real-world example. In case of `D-Grid`, the primary pedestrian starts focusing on the potential collider $N2$ despite it being distant compared to $N1$ thereby preventing collision by staying closer to $N1$. On the other hand, `S-Grid` keeps focusing on the $N1$ which is not desirable. It is interesting to note that once $N2$ passes the primary pedestrian, both `D-Grid` and `S-Grid` shift the attention of the primary pedestrian back to $N1$.

In Scene 2, we demonstrate the effectiveness of our proposed `D-Grid` module in a complex real-world scenario. For `D-Grid`, initially the primary pedestrian focuses on $N3$ to prevent collision. On successfully avoiding collision with $N3$, `D-Grid` immediately shifts the focus to the pair $N1$ and $N2$ as they would potentially lead to a collision. On coming in close proximity to $N1$ and $N2$, the focus significantly shifts towards $N1$ as it is closer to the primary pedestrian. Finally,

on passing $N1$ and $N2$, the primary pedestrian attends to the pedestrian $N4$ in front. On the other hand, `S-Grid` passes in between $N1$ and $N2$, such behavior is not expected in human crowds.

Thus, we can see that LRP is an effective investigative tool to understand the rationale behind the NN decisions. We can observe that, along with having a lower Col-I metric as compared to `S-Grid` in Table III, the decision making of our domain-knowledge based `D-Grid` satisfies human intuition while navigating crowds. The LRP technique is generic and can be applied on top of any existing trained interaction module architecture.

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. The incorporation of Col-I metrics paints a more complete picture of model performance. Secondly, relative velocity plays a crucial role in learning collision avoidance in the real-world. Thirdly, a simple concatenation strategy performs at par with the sophisticated attention-based counterparts. We believe that the concatenation baseline should be a standard baseline to compare to when designing future information aggregating modules. Finally, the LRP technique is a useful investigative tool to gain insights regarding the decision-making process of NNs. We hope that such practices will help to accelerate the development of interaction modules in future research. There certainly exists room for improvement, and we hope that our benchmark provides the necessary resources to advance the field of trajectory forecasting. We open-source our code for reproducibility.

VII. Conclusion

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 model designs on identical grounds. We presented an in-depth analysis of the design of interaction modules proposed in the literature and proposed two domain-knowledge inspired interaction models.

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 [96], 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.

Our proposed models outperform 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 in terms of distance-based metrics, when compared on equal grounds. Our proposed designs show significant gains in reducing model prediction collisions. 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.

Acknowledgment

The authors would like to thank the reviewers for their helpful comments.

References

[1] B. Jiang, "SimPed: Simulating pedestrian flows in a virtual urban environment," *J. Geograph. Inf. Decis. Anal.*, vol. 3, 1999.

[2] A. Lerner, Y. Chrysanthou, and D. Lischinski, "Crowds by example," *Comput. Graph. Forum*, vol. 26, no. 3, pp. 655–664, Sep. 2007.

[3] S. Bitgood, "An analysis of visitor circulation: Movement patterns and the general value principle," *Curator, Museum J.*, vol. 49, no. 4, pp. 463–475, Oct. 2006.

[4] A. Horni, K. Nagel, and K. W. Axhausen, *The Multi-Agent Transport Simulation MATSim*. London, U.K.: Ubiquity Press, 2016.

[5] R. Mehran, A. Oyama, and M. Shah, "Abnormal crowd behavior detection using social force model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 935–942.

[6] D. Helbing, I. J. Farkas, P. Molnar, and T. Vicsek, "Simulation of pedestrian crowds in normal and evacuation situations," vol. 21, pp. 21–58, Jan. 2002.

[7] D. Helbing, I. Farkas, and T. Vicsek, "Simulating dynamical features of escape panic," *Nature*, vol. 407, no. 6803, pp. 487–490, Sep. 2000.

[8] X. Zheng, T. Zhong, and M. Liu, "Modeling crowd evacuation of a building based on seven methodological approaches," *Building Environ.*, vol. 44, pp. 437–445, 2009.

[9] M. Moussaid, D. Helbing, and G. Theraulaz, "How simple rules determine pedestrian behavior and crowd disasters," *Proc. Nat. Acad. Sci. USA*, vol. 108, no. 17, pp. 6884–6888, Apr. 2011.

[10] H. Dong, M. Zhou, Q. Wang, X. Yang, and F.-Y. Wang, "State-of-the-art pedestrian and evacuation dynamics," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 5, pp. 1849–1866, May 2020.

[11] *Waymo Safety Report 2018*. Accessed: Apr. 1, 2020. [Online]. Available: https://storage.googleapis.com/sdc-prod/v1/safety-report/safety%20report%2%02018.pdf

[12] *Uber ATG Safety Report 2020*. Accessed: Apr. 1, 2020. [Online]. Available: https://uber.app.box.com/v/uberatgsafetyreport

[13] C. Chen, Y. Liu, S. Kreiss, and A. Alahi, "Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 6015–6022.

[14] A. Rasouli and J. K. Tsotsos, "Autonomous vehicles that interact with pedestrians: A survey of theory and practice," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 3, pp. 900–918, Mar. 2020.

[15] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social LSTM: Human trajectory prediction in crowded spaces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 961–971.

[16] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PLoS ONE*, vol. 10, no. 7, Jul. 2015, Art. no. e0130140.

[17] D. Helbing and P. Molnár, "Social force model for pedestrian dynamics," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 51, no. 5, pp. 4282–4286, May 1995.

[18] J. Elfring, R. van de Molengraft, and M. Steinbuch, "Learning intentions for improved human motion prediction," *Robot. Auto. Syst.*, vol. 62, no. 4, pp. 591–602, Apr. 2014.

[19] A. Rudenko, L. Palmieri, A. J. Lilienthal, and K. O. Arras, "Human motion prediction under social grouping constraints," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 3358–3364.

[20] A. Rudenko, L. Palmieri, and K. O. Arras, "Joint long-term prediction of human motion using a planning-based social force approach," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 1–7.

[21] B. Yu, K. Zhu, K. Wu, and M. Zhang, "Improved OpenCL-based implementation of social field pedestrian model," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 7, pp. 2828–2839, Jul. 2020.

[22] W. Choi and S. Savarese, "A unified framework for multi-target tracking and collective activity recognition," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2012, pp. 215–230.

[23] W. Choi and S. Savarese, "Understanding collective activitiesof people from videos," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 6, pp. 1242–1257, Jun. 2014.

[24] C. Burstedde, K. Klauck, A. Schadschneider, and J. Zittartz, "Simulation of pedestrian dynamics using a two-dimensional cellular automaton," *Phys. A, Stat. Mech. Appl.*, vol. 295, pp. 507–525, 2001.

[25] G. Vizzari, L. Manenti, K. Ohtsuka, and K. Shimura, "An agent-based pedestrian and group dynamics model applied to experimental and real-world scenarios," *J. Intell. Transp. Syst.*, vol. 19, no. 1, pp. 32–45, Jan. 2015.

[26] J. van den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2008, pp. 1928–1935.

[27] G. Antonini, M. Bierlaire, and M. Weber, "Discrete choice models of pedestrian walking behavior," *Transp. Res. B, Methodol.*, vol. 40, no. 8, pp. 667–687, Sep. 2006.

[28] A. Treuille, S. Cooper, and Z. Popović, "Continuum crowds," in *Proc. ACM SIGGRAPH Papers (SIGGRAPH)*, 2006, pp. 1–9.

[29] C. T. M. Keat and C. Laugier, "Modelling smooth paths using Gaussian processes," in *Field and Service Robotics*. Berlin, Germany: Springer, 2007.

[30] K. Kim, D. Lee, and I. Essa, "Gaussian process regression flow for analysis of motion trajectories," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 1164–1171.

[31] R. Q. Mínguez, I. P. Alonso, D. Fernández-Llorca, and M. A. Sotelo, "Pedestrian path, pose, and intention prediction through Gaussian process dynamical models and pedestrian activity recognition," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 5, pp. 1803–1814, May 2019.

[32] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese, "Learning social etiquette: Human trajectory understanding in crowded scenes," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 549–565.

[33] A. Alahi, V. Ramanathan, and L. Fei-Fei, "Socially-aware large-scale crowd forecasting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2211–2218.

[34] A. Alahi, V. Ramanathan, and L. Fei-Fei, "Tracking millions of humans in crowded spaces," in *Group and Crowd Behavior for Computer Vision*. Amsterdam, The Netherlands: Elsevier, 2017, pp. 115–135.

[35] S. Yi, H. Li, and X. Wang, "Understanding pedestrian behaviors from stationary crowd groups," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3488–3496.

[36] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 6645–6649.

[37] F. Meng, "Neural machine translation by jointly learning to align and translate," *CoRR*, vol. abs/1409.0473, 2015.

[38] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3156–3164.

[39] C. Cao *et al.*, "Look and think twice: Capturing top-down visual attention with feedback convolutional neural networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2956–2964.

[40] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[41] P. Coscia, F. Castaldo, F. A. Palmieri, L. Ballan, A. Alahi, and S. Savarese, "Point-based path prediction from polar histograms," in *Proc. 19th Int. Conf. Inf. Fusion (FUSION)*, Jul. 2016, pp. 1961–1967.

[42] D. Varshneya and G. Srinivasaraghavan, "Human trajectory prediction using spatially aware deep attention models," 2017, *arXiv:1705.09436*. [Online]. Available: http://arxiv.org/abs/1705.09436

[43] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. S. Torr, and M. Chandraker, "DESIRE: Distant future prediction in dynamic scenes with interacting agents," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2165–2174.

[44] F. Bartoli, G. Lisanti, L. Ballan, and A. D. Bimbo, "Context-aware trajectory prediction," in *Proc. 24th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2018, pp. 1941–1946.

[45] H. Xue, D. Q. Huynh, and M. Reynolds, "SS-LSTM: A hierarchical LSTM model for pedestrian trajectory prediction," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2018, pp. 1186–1194.

[46] T. Zhao *et al.*, "Multi-agent tensor fusion for contextual trajectory prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019.

[47] M. Lisotto, P. Coscia, and L. Ballan, "Social and scene-aware trajectory prediction in crowded spaces," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 2567–2574.

[48] M. Pfeiffer, G. Paolo, H. Sommer, J. Nieto, R. Siegwart, and C. Cadena, "A data-driven model for interaction-aware pedestrian motion prediction in object cluttered environments," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 1–8.

[49] A. Alahi *et al.*, "Learning to predict human behavior in crowded scenes," in *Group and Crowd Behavior for Computer Vision*. Amsterdam, The Netherlands: Elsevier, 2017, pp. 183–207.

[50] X. Shi, X. Shao, Z. Guo, G. Wu, H. Zhang, and R. Shibasaki, "Pedestrian trajectory prediction in extremely crowded scenarios," *Sensors*, vol. 19, no. 5, p. 1223, 2019.

[51] N. Bisagno, B. O. Zhang, and N. Conci, "Group LSTM: Group trajectory prediction in crowded scenarios," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 213–225.

[52] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social GAN: Socially acceptable trajectories with generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2255–2264.

[53] P. Zhang, W. Ouyang, P. Zhang, J. Xue, and N. Zheng, "SR-LSTM: State refinement for LSTM towards pedestrian trajectory prediction," 2019, *arXiv:1903.02793*. [Online]. Available: http://arxiv.org/abs/1903.02793

[54] Y. Zhu, D. Qian, D. Ren, and H. Xia, "StarNet: Pedestrian trajectory prediction using deep neural network in star topology," 2019, *arXiv:1906.01797*. [Online]. Available: http://arxiv.org/abs/1906.01797

[55] B. Ivanovic and M. Pavone, "The Trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2019, pp. 2375–2384.

[56] J. Liang, L. Jiang, J. C. Niebles, A. Hauptmann, and L. Fei-Fei, "Peeking into the future: Predicting future person activities and locations in videos," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2019, pp. 5718–5727.

[57] A. Tordeux, M. Chraibi, A. Seyfried, and A. Schadschneider, "Prediction of pedestrian dynamics in complex architectures with artificial neural networks," *J. Intell. Transp. Syst.*, vol. 24, no. 6, pp. 556–568, Nov. 2020.

[58] Y. Ma, E. W. M. Lee, and R. K. K. Yuen, "An artificial intelligence-based approach for simulating pedestrian movement," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 11, pp. 3159–3170, Nov. 2016.

[59] I. Hasan, F. Setti, T. Tsesmelis, A. D. Bue, F. Galasso, and M. Cristani, "MX-LSTM: Mixing tracklets and vislets to jointly forecast trajectories and head poses," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6067–6076.

[60] I. Hasan *et al.*, "Forecasting people trajectories and head poses by jointly reasoning on tracklets and vislets," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 4, pp. 1267–1278, Apr. 2021.

[61] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, "Trajectron++: Multi-agent generative trajectory forecasting with heterogeneous data for control," 2020, *arXiv:2001.03093*. [Online]. Available: https://arxiv.org/abs/2001.03093

[62] I. Hasan, F. Setti, T. Tsesmelis, A. D. Bue, M. Cristani, and F. Galasso, "'Seeing is believing': Pedestrian trajectory forecasting using visual frustum of attention," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2018, pp. 1178–1185.

[63] Y. Xu, Z. Piao, and S. Gao, "Encoding crowd interaction with deep neural network for pedestrian trajectory prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5275–5284.

[64] T. Fernando, S. Denman, S. Sridharan, and C. Fookes, "Soft+ hardwired attention: An LSTM framework for human trajectory prediction and abnormal event detection," *Neural Netw., Off. J. Int. Neural Netw. Soc.*, vol. 108, pp. 466–478, Dec. 2018.

[65] J. Li, H. Ma, Z. Zhang, and M. Tomizuka, "Social-WaGDAT: Interaction-aware trajectory prediction via Wasserstein graph double-attention network," 2020, *arXiv:2002.06241*. [Online]. Available: http://arxiv.org/abs/2002.06241

[66] A. Sadeghian, V. Kosaraju, A. Sadeghian, N. Hirose, and S. Savarese, "SoPhie: An attentive GAN for predicting paths compliant to social and physical constraints," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 1349–1358.

[67] V. Kosaraju, A. Sadeghian, R. Martín-Martín, I. Reid, S. H. Rezatofighi, and S. Savarese, "Social-BiGAT: Multimodal trajectory forecasting using bicycle-GAN and graph attention networks," 2019, *arXiv:1907.03395*. [Online]. Available: http://arxiv.org/abs/1907.03395

[68] J. Amirian, J.-B. Hayet, and J. Pettre, "Social ways: Learning multi-modal distributions of pedestrian trajectories with GANs," 2019, *arXiv:1904.09507*. [Online]. Available: http://arxiv.org/abs/1904.09507

[69] T. Fernando, S. Denman, S. Sridharan, and C. Fookes, "GD-GAN: Generative adversarial networks for trajectory prediction and group detection in crowds," 2018, *arXiv:1812.07667*. [Online]. Available: http://arxiv.org/abs/1812.07667

[70] S. Haddad, M. Wu, H. Wei, and S. K. Lam, "Situation-aware pedestrian trajectory prediction with spatio-temporal attention model," 2019, *arXiv:1902.05437*. [Online]. Available: http://arxiv.org/abs/1902.05437

[71] Y. Huang, H. Bi, Z. Li, T. Mao, and Z. Wang, "STGAT: Modeling spatial-temporal interactions for human trajectory prediction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6271–6280.

[72] A. Mohamed, K. Qian, M. Elhoseiny, and C. Claudel, "Social-STGCNN: A social spatio-temporal graph convolutional neural network for human trajectory prediction," 2020, *arXiv:2002.11927*. [Online]. Available: http://arxiv.org/abs/2002.11927

[73] J. Li, F. Yang, M. Tomizuka, and C. Choi, "EvolveGraph: Heterogeneous multi-agent multi-modal trajectory prediction with evolving interaction graphs," 2020, *arXiv:2003.13924*. [Online]. Available: http://arxiv.org/abs/2003.13924

[74] J. Sun, Q. Jiang, and C. Lu, "Recursive social behavior graph for trajectory prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 657–666.

[75] C. Yu, X. Ma, J. Ren, H. Zhao, and S. Yi, "Spatio-temporal graph transformer networks for pedestrian trajectory prediction," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 507–523.

[76] K. Mangalam *et al.*, "It is not the journey but the destination: Endpoint conditioned trajectory prediction," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 759–776.

[77] C. Tao, Q. Jiang, L. Duan, and P. Luo, "Dynamic and static context-aware LSTM for multi-agent motion prediction," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 547–563.

[78] A. Vaswani *et al.*, "Attention is all you need," in *Proc. NIPS*, 2017, pp. 6000–6010.

[79] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *CoRR*, vol. abs/1409.0473, 2015.

[80] A. Rudenko, L. Palmieri, M. Herman, K. M. Kitani, D. M. Gavrila, and K. O. Arras, "Human motion trajectory prediction: A survey," 2019, *arXiv:1905.06113*. [Online]. Available: http://arxiv.org/abs/1905.06113

[81] J. F. P. Kooij, N. Schneider, F. Flohr, and D. Gavrila, "Context-based pedestrian path prediction," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2014, pp. 618–633.

[82] A. Vemula, K. Muelling, and J. Oh, "Social attention: Modeling attention in human crowds," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 1–7.

[83] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2014.

[84] A. Graves, "Generating sequences with recurrent neural networks," *ArXiv*, vol. abs/1308.0850, 2013.

[85] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," 2018, *arXiv:1710.10903*. [Online]. Available: https://arxiv.org/abs/1710.10903

[86] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, 2015.

[87] J. T. Springenberg, A. Dosovitskiy, M. A. Riedmiller, and T. Brox, "Striving for simplicity: The all convolutional net," in *Proc. ICLR*, 2015. [Online]. Available: http://lmb.informatik.uni-freiburg.de/Publications/2015/DB15a

[88] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *Proc. ICML*, 2017, pp. 3319–3328.

[89] M. Alber *et al.*, "Innvestigate neural networks!" *J. Mach. Learn. Res.*, vol. 20, no. 93, pp. 1–8, 2019. [Online]. Available: http://jmlr.org/papers/v20/18-540.html

[90] L. Arras *et al.*, "Explaining and interpreting LSTMs," 2019, *arXiv:1909.12114*. [Online]. Available: http://arxiv.org/abs/1909.12114

[91] E. Parzen, "On estimation of a probability density function and mode," *Ann. Math. Statist.*, vol. 33, pp. 1065–1076, 1962.

[92] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, 2014.

[93] H. Cheng, W. Liao, M. Y. Yang, B. Rosenhahn, and M. Sester, "AMENet: Attentive maps encoder network for trajectory prediction," *ArXiv*, vol. abs/2006.08264, 2020.

[94] Y. Zhu, D. Ren, M. Fan, D. Qian, X. Li, and H. Xia, "Robust trajectory forecasting for multiple intelligent agents in dynamic scene," 2020, *arXiv:2005.13133*. [Online]. Available: http://arxiv.org/abs/2005.13133

[95] C. Scholler, V. Aravantinos, F. Lay, and A. Knoll, "What the constant velocity model can teach us about pedestrian motion prediction," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 1696–1703, Apr. 2020.

[96] A. Sadeghian, V. Kosaraju, A. Gupta, S. Savarese, and A. Alahi, "Trajnet: Towards a benchmark for human trajectory prediction," 2018. Accessed: Nov. 8, 2019. [Online]. Available: http://trajnet.stanford.edu

Parth Kothari received the bachelor's degree (Hons.) in electrical engineering from the Indian Institute of Technology, Bombay, in 2018. He is currently pursuing the Ph.D. degree with the Visual Intelligence for Transportation (VITA) Laboratory, EPFL, Switzerland. His research interests include trajectory forecasting, modeling crowd behaviors, and deep learning.

Sven Kreiss is currently a Post-Doctoral Researcher with the Visual Intelligence for Transportation (VITA) Laboratory, EPFL, Switzerland, focusing on perception with composite fields. Before returning to academia, he was the Senior Data Scientist with Sidewalk Labs (Alphabet, Google Sister) and worked on geospatial machine learning for urban environments. Prior to his industry experience, he developed statistical tools and methods used in particle physics research.

Alexandre Alahi received the Ph.D. degree from EPFL. He spent five years with Stanford University as a Post-Doctoral Researcher and the Research Scientist. He is currently an Assistant Professor with EPFL. His research interests include machines to perceive the world and make decisions in the context of transportation problems and smart environments. He has worked on the theoretical challenges and practical applications of socially-aware artificial intelligence, that is, systems equipped with perception and social intelligence. He was awarded the Swiss NSF early and advanced researcher grants for his work on predicting human social behavior. He has also co-founded multiple startups, such as Visiosafe, and won several startup competitions. He was elected as one of the Top 20 Swiss Venture leaders in 2010.