RED: A simple but effective Baseline Predictor for the *TrajNet* Benchmark

Stefan Becker[0000−0001−7367−2519] ⋆, Ronny Hug[0000−0001−6104−710X] ⋆, Wolfgang Hübner[0000−0001−5634−6324], and Michael Arens[0000−0002−7857−0332]

Fraunhofer Institute for Optronics, System Technologies, and Image Exploitation IOSB
Gutleuthausstr. 1, 76275 Ettlingen, Germany

Abstract. In recent years, there is a shift from modeling the tracking problem based on Bayesian formulation towards using deep neural networks. Towards this end, in this paper the effectiveness of various deep neural networks for predicting future pedestrian paths are evaluated. The analyzed deep networks solely rely, like in the traditional approaches, on observed tracklets without human-human interaction information. The evaluation is done on the publicly available *TrajNet* benchmark dataset [39], which builds up a repository of considerable and popular datasets for trajectory prediction. We show how a Recurrent-Encoder with a Dense layer stacked on top, referred to as RED-predictor, is able to achieve top-rank at the *TrajNet* 2018 challenge compared to elaborated models. Further, we investigate failure cases and give explanations for observed phenomena, and give some recommendations for overcoming demonstrated shortcomings.

Keywords: Trajectory Forecasting, Path Prediction, Trajectory-based Activity Forecasting

1 Introduction

The prediction of possible future paths is a central building block for an automated risk assessment. The applications cover a wide range from mobile robot navigation, including autonomous driving, smart video surveillance to object tracking. Dividing the many variants of forecasting approaches can be roughly done by asking how the problem is addressed or what kind of information is provided. Firstly, addressing this problem reaches from traditional approaches such as the Kalman filter [25], linear [34] or Gaussian regression models [42], auto-regressive models [2], time-series analysis [37] to optimal control theory [27], deep learning combined with game theory [32], or the application of deep convolutional networks [21] and recurrent neural networks (RNNs) as a sequence generation problem [3, 4, 23]. Secondly, the grouping can be done by using the provided information. On the one hand, the approaches can solely rely on observations of

⋆ Equal contribution.

consecutive positions extracted by visual tracking or on the other hand, by using richer context information. This can be for example human-human interactions or human-space interactions or general additional visual extracted information such as pedestrian head orientation [28] or head poses [17]. For some representative approaches which model human-human interactions, one should mention the works of Helbing and Molnár [19] and Coscia et al. [10] or approaches in combination with RNNs such as the works of Alahi et al.[3, 4]. The spatial context of motion can in principle be learned by training a model on observed positions of a particular scene, but it is not guaranteed that the model successfully captures spatial points of interest and does not only implicitly keep spatial information by performing path integration in order to predict new positions. Nevertheless, here we distinguish such approaches from approaches where scene context is provided as further cue for example by semantic labeling [6] or scene encoding [44]. The challenges of *Trajectory Forecasting Benchmarking* (*TrajNet* 2018) [39] are designed to cover some inherent properties of human motion in crowded scenes. The *World H-H TrajNet* challenge in particular looks at predicting motions in world plane coordinates of human-human interactions. The aim of this paper is to find an effective baseline predictor only based on the partial history and find the maximum potential achievable prediction accuracy for this challenge. Achieving this objective involves an evaluation of different deep neural networks for trajectory prediction and analysis of the datasets properties. Further, we propose small changes and pre-processing steps to modify a standard RNN prediction model to result in a simple but effective RNN architecture that obtains comparable performance to more elaborated models, which additionally captures the interpersonal aspect of human-human interaction.
The paper is structured as follows. Firstly, the properties of the *TrajNet* benchmark dataset are analyzed in section 2. Then, some basic deep neural networks are shortly described and evaluated (section 3). Further, the modifications in order to increase the prediction performance are presented in section 4. The achieved results and an additional failure analysis are discussed in section 5. Finally, a conclusion is given in section 6.

2 *TrajNet* Benchmark Dataset Analysis

The trajectory forecasting challenges *TrajNet* [39] provide the community with a defined and repeatable way of comparing path prediction approaches as well as a common platform for discussions in the field. In this section some properties of the current repository for the *World H-H TrajNet* challenge of popular datasets for trajectory-based activity forecasting are analyzed and thereby design choices for the proposed predictor are deduced.

In most datasets, the scene is observed from a bird's eye view, but there are also scenarios where the scene is observed under a higher depression angle. The selected surveillance datasets cover real world scenarios with a varying crowd densities and varying complexity of trajectory patterns. Details of the datasets are summarized in table 1 (adapted from *TrajNet* website). The selection in-

Table 1. Training (green) and test (cyan) dataset of the world plane human-human dataset challenge (adapted from the *TrajNet* website [39]).

Name	Resolution	#Pedestrian	Framerate	Reference
BIWI Hotel	720 × 576	389	2.5	[36]
Crowds Zara	720 × 576	204	2.5	[30]
Crowds Students	720 × 576	415	2.5	[30]
Crowds Arxiepiskopi	720 × 576	24	2.5	[30]
PETS 2009	768 × 576	19	2.5	[14]
Stanford Drone Dataset (SDD)	595 × 326	3295	2.5	[38]
BIWI ETH	640 × 480	360	2.5	[36]
Crowds Zara	720 × 576	148	2.5	[30]
Crowds Uni Examples	720 × 576	118	2.5	[30]
Stanford Drone Dataset (SDD)	595 × 326	3297	2.5	[38]

cludes the following datasets. The *BIWI Walking Pedestrians Dataset* [36] also sometimes referenced as *ETH Walking Pedestrians (EWAP)*, which is split into two sets (*ETH* and *Hotel*). The *Crowds* dataset also called *UCY "Crowds-by-Example"* dataset [30] contains three scenes from an oblique view, where the first (Zara) shows a part of a shopping street, the second (*Students*/*Uni Examples*) captures a part of the uni campus and the third scene (*Arxiepiskopi*) captures a different part of the campus. Then, the *Stanford Drone Dataset (SDD)* [38] consists of multiple aerial images capturing different locations around the Stanford campus. And finally the *PETS 2009* dataset [14], where different outdoor crowds activities are observed by multiple static cameras. Sample images with full trajectories and tracklets are shown in figure 1.

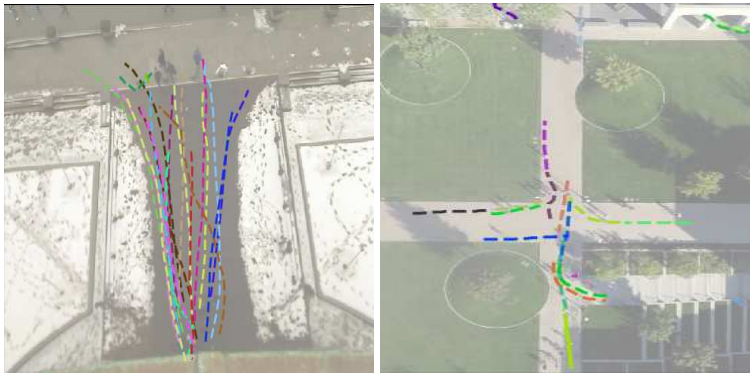

Fig. 1. Example trajectories from the *BIWI ETH* dataset and example tracklets from the sequence *Hyang_07* from the *Stanford Drone Dataset (SDD)*.

It is common and good practice to apply cross-validation. For the TajNet challenge, it is done by omitting complete datasets for testing. Because the behavior of humans in crowds is scene-independent and for measuring the generalization capabilities of various approaches across datasets this is very reasonable, in particular for providing a benchmark for human-human interactions. Nevertheless, by combining all training sets the spatial context of scene specific motion and the reference systems are lost. When only relying on observed motion trajectories positional information is crucial in order to learn spatio-temporal variation. For example, the sidewalks in the *Hyang* sequences (see figure 1) lead to a spatially depending change in the curvature of a trajectory. Since our focus is on deep neural networks including RNNs, the shift from position information to higher order motion helps to overcome some drawbacks. Before RNNs were successfully applied for tracking pedestrians in a surveillance scenario, they gained attention due to their success in tasks such as speech recognition [15, 9] and caption generation [11, 43]. Since these domain are particularly different to trajectory prediction in certain aspects, their position-dependent movement is not important. Accordingly, RNNs can benefit from conditioning on previous offsets for scene independent motion prediction. This insight is not new, yet utilizing offsets really helps not only stabilizing the learning process but also improves the prediction performance for the evaluated networks. This shift to offsets or rather velocities has been also successfully applied for example for the prediction of human poses based on RNNs [33]. In the context of deep networks the same effect can also be achieved by adding residual connections, which have been shown to improve performance on deep convolutional networks [18]. Presumably due to the limitation of the input and output spaces, for applying on the *TrajNet* challenge instead of prediction of the next position (where will the person be next) predicting the following offsets (where will the person go next) [23, 24] also contributed to increased prediction accuracy. This becomes immediately apparent by looking at the complete tracklets of the training and test set (see figure 2). Firstly, it takes a considerably higher modeling effort to represent all possible positions instead of modeling particular velocities. Further, input data outside the training range can lead to undefined states in the deep network, which result in an unreasonably random output. Some of the initialization tracklets clearly lie outside the training input space. Also, approaches with profit from human-human interaction such as [16, 17, 4, 3] in combination with deep networks lack here information about surrounding persons to interact, so that the decoding of relative distances is not possible because of a reduced person density.

Another factor for improving the prediction performance is becoming apparent when contemplating the offset distribution of the data. Figure 3 shows the offsets histograms for x and y separately. Due to the loss of the reference system, it is impossible to assume a reasonable location distribution a-priori. In contrast, the offset and magnitude distribution clearly reflects the preferred walking speeds in the data. The histograms also show that a large amount of persons is standing. In the recent work of Hasan et al. [17], it was emphasized

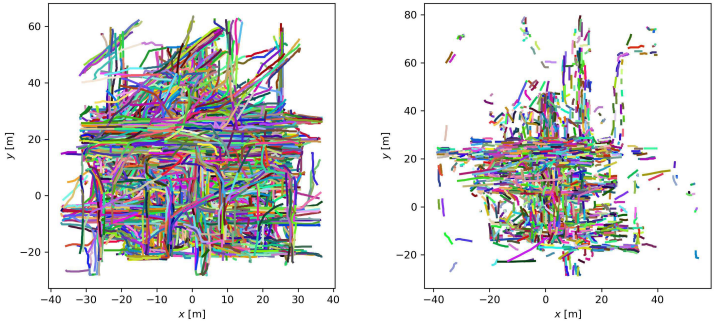

Fig. 2. (Left) Visualization of all tracklets of the training set from the *TrajNet* dataset collection. (Right) Visualization of all initialization tracklets of the test set.

that forecasting errors are in general higher when the speed of persons is lower and argued that when persons are walking slowly their behavior becomes less predictable, due to physical reasons (less inertia). During our testing we discovered the same phenomenon. In particular RNN based networks tend to overestimate slow velocities and do sometimes not accurately identify the standing behavior. Despite this problem, the range of offsets is very limited compared to the location distribution and shows a clear tendency towards expected a-priori values. Common techniques for sequence prediction problems are normalization and standardization of the input data. Whereby normalization has a similar role on the position data, applying standardization on position input data shows no benefit. In our experiments, standardization worked slightly better than normalization or an embedding layer for input encoding. Although the effect on the performance is quite low for the *TrajNet* challenge, our best result is achieved using standardized offsets as input. It is rarely strictly necessary to standardize the inputs, but there are practical reasons such as accelerating the training or reducing the chances of getting stuck in local optima [7]. Predicting offsets also guarantees that the output directly conforms better with the range of common activation functions.

Without discretization artifacts, the dynamic of humans is smooth and persistent. The trajectory data from the *TrajNet* dataset includes varying discretization artifacts or noise levels resulting from different methods with which ground truth data was generated. Part of the ground truth trajectories are generated by a visual tracker or manually annotated.

For approximating the amount of noise in the datasets, the distance between a smoothed spline fit through the complete tracklets is compared to the provided ground truth tracklet points. The spline fitting is done with a polynom of degree $k = 4$ independent for the x and y values. If the smoothing is too strong, it can drift too far away from the actual data. Nevertheless, the achieved fitted

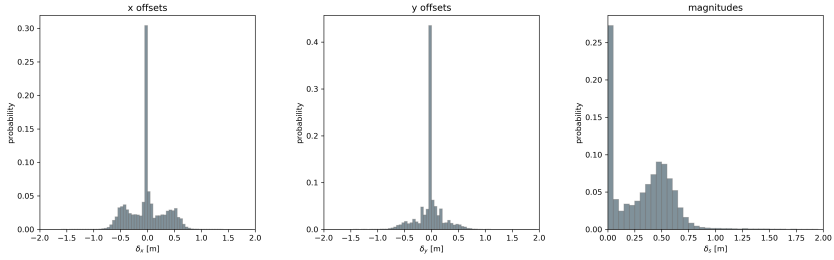

Fig. 3. (Left, Middle) Offset histograms of the training set. (Right) Magnitude histogram of the offsets.

trajectories form a smooth and natural path and are used as rough assessment for the noise levels in the ground truth trajectory data. The results for the training set are summarized in table 2.

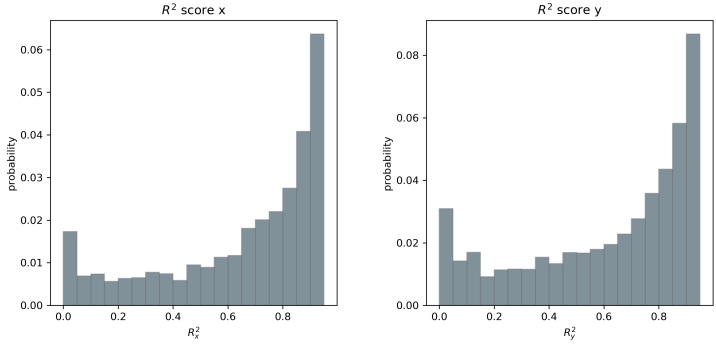

Fig. 4. Coefficient of determination R^2 for x and y for all training tracklets of the *World H-H TrajNet* challenge.

The approximated noise levels clearly show the variation in the ground truth data. In order to outperform a linear baseline predictor the learned model must be able to successfully model different velocity profiles and capture curved paths out of input data with different noise levels. Due to the varying noise levels, initial experiments to solely train on smoothed fitted trajectories with synthetic noise performed worse. Nevertheless, for the prediction of the future steps the best performing predictor is trained to forecast smoothed paths. Before the different evaluated models are introduced, the last data analysis of the training set is intended to assess the complexity in terms of the non-linearity of the trajec-

Table 2. Standard deviation of the distance between a smoothed spline fit and the ground truth trajectory data. The average R^2 score for all tracklets in the subsets.

Name	$\sigma_{x,\text{spline}}$ [m]	$\sigma_{y,\text{spline}}$ [m]	$\bar{R}_x^2$	$\bar{R}_y^2$
Overall	0.067	0.069	0.889	0.811
BIWI Hotel	0.042	0.031	0.637	0.876
Crowds Zara_02	0.029	0.035	0.952	0.758
Crowds Zara_03	0.026	0.031	0.935	0.716
Crowds Students_01	0.033	0.029	0.868	0.852
Crowds Students_03	0.039	0.040	0.915	0.76
Crowds Arxiepiskopi_01	0.050	0.027	0.959	0.677
PETS 2009 S2L1	0.037	0.026	0.781	0.877
SSD Bookstore_00	0.060	0.063	0.889	0.844
SSD Bookstore_01	0.054	0.053	0.879	0.878
SSD Bookstore_02	0.068	0.073	0.861	0.921
SSD Bookstore_03	0.069	0.061	0.951	0.830
SSD Coupa_03	0.057	0.043	0.954	0.937
SSD Deathcircle_00	0.072	0.079	0.893	0.808
SSD Deathcircle_01	0.086	0.103	0.850	0.818
SSD Deathcircle_02	0.151	0.158	0.772	0.591
SSD Deathcircle_03	0.116	0.134	0.816	0.770
SSD Deathcircle_04	0.215	0.160	0.738	0.713
SSD Gates_00	0.054	0.073	0.980	0.735
SSD Gates_01	0.064	0.084	0.859	0.890
SSD Gates_03	0.086	0.106	0.847	0.860
SSD Gates_04	0.071	0.155	0.820	0.906
SSD Gates_05	0.069	0.067	0.858	0.904
SSD Gates_06	0.077	0.072	0.840	0.905
SSD Gates_07	0.084	0.126	0.908	0.817
SSD Gates_08	0.076	0.088	0.922	0.820
SSD Hyang_04	0.048	0.050	0.829	0.842
SSD Hyang_05	0.059	0.081	0.872	0.740
SSD Hyang_06	0.070	0.066	0.875	0.811
SSD Hyang_07	0.040	0.079	0.879	0.894
SSD Hyang_09	0.036	0.088	0.998	0.652
SSD Nexus_00	0.076	0.082	0.886	0.742
SSD Nexus_01	0.067	0.095	0.929	0.771
SSD Nexus_02	0.069	0.074	0.934	0.726
SSD Nexus_03	0.188	0.113	0.786	0.572
SSD Nexus_04	0.097	0.073	0.847	0.724
SSD Nexus_07	0.053	0.069	0.935	0.764
SSD Nexus_08	0.067	0.070	0.926	0.681
SSD Nexus_09	0.052	0.094	0.913	0.816

tories. Therefore, the coefficient of determination R^2 for a linear interpolation is calculated separately for the x and y values. This linear interpolation serves as baseline predictor for the *TrajNet* challenge. The histograms of R^2 for the training set are shown in figure 4. R^2 is the percentage of the variation that is explained by the model and is used to determine the suitability of the regression fit as a linearity measure [12]. The average R^2 values are summarized in table 2. It can be seen that for most tracklets a linear interpolation works very well. In order to outperform the linear interpolation baseline, it is crucial to not only cover a variety of complex observed motions, but to also produce robust results in simpler situations. As mentioned above, the person velocity has to be effectively captured by the model.

3 Models and Evaluation

The goal of this work is by using a sort of coarse to fine searching strategy to reach the maximum achievable prediction accuracy without further cues such as human-human interaction or human-space interaction based on basic networks. Towards this end, we started with a set of networks with a limited set of hyperparameters to narrow it down to one network, in order to then extend the hyperparameter set for a more exhaustive tuning. The multi-modal aspect of trajectory prediction is hardly considerable when there is no fixed reference system. Thus, the performance is compared in accordance to the community with the two error metrics of the average displacement error (ADE) and the final displacement error (FDE) (see for example [3, 41, 36, 16, 44, 17]). The average of both combined values are then used as overall average to rank the approaches. The ADE is defined as the average L2 distance between ground truth and the prediction over all predicted time steps and the FDE is defined as the L2 distance between the predicted final position and the true final position. For the *World H-H TrajNet* challenge the unit of the error metrics is meter. For all experiments, 8 (3.2 seconds) consecutive positions are observed, before predicting the next 12 (4.8 seconds) positions.

Besides the provided approaches of the *World H-H TrajNet* challenge, the following basic neural networks for a coarse evaluation are selected:

Multi-Layer-Perceptron (MLP): The MLP is tested with different linear and non-linear activation functions. One variation concatenates all inputs and predicts 24 outputs directly. Further, cascaded architectures with a step-wise prediction are examined. We vary between different coordinate system of Euclidean and polar coordinates. As mentioned in section 2, positions and offsets (also orientation normalized) are considered as inputs and outputs.

RNN-MLP: RNNs extend feed-forward networks or rather the MLP model due to their recurrent connections between hidden units. Vanilla RNNs produce an output at each time step. For the evaluation of the RNN-MLP, we vary only the MLP layer which is used for the decoding of the positions and offsets.

RNN-Encoder-MLP: In contrast to the RNN-MLP network, the complete initialization tracklet is used to generate the internal representation before

a prediction is done. The RNN-Encoder-MLP is varied by alternating activation functions for the MLP and by alternatively predicting the complete future path/offsets instead of only next steps. As a further alternative, the full path is predicted as offsets to one reference point instead of applying path integration in order to predict the final position.

RNN-Encoder-Decoder-Model (Seq2Seq): In addition to RNN-Encoder-MLPs, Seq2Seqs include a second network. This second decoder network takes the internal representation of the encoder and then starts predicting the next steps. The different settings for the evaluation of this model where due to alternating activation functions for the MLP on top of the decoder RNN.

Temporal Convolutional Networks (TCN): As an alternative to RNNs and based on *WaveNets*[35], Bai et al. [5] introduced a general convolution architecture for sequence prediction. We tested their standard and extended architecture with a gating mechanism (GTCN). For a more detailed description, we refer to the original papers.

All networks were trained with varying number of layers (1 to 5) and hidden units (4 to 64) using stochastic gradient descent with a fixed learning rate of 0.005. The models are trained for 100 epochs using ADAM optimizer [26] and have been implemented in *Tensorflow* [1]. Firstly, only standard RNN cells are used for the experiments. Later, we also tested with RNNs variants Long Short-Term Memory [20] (LSTM) and Gated Recurrent Unit [8] (GRU). As loss the mean squared error between the predicted and the ground truth position or offsets over all time steps is used.
In order to emphasize trends a part from the result of the first experiments are summarized in table 3 (highlighted in gray). The best results were achieved with the RNN-Encoder-MLP. However, in most cases the different architectures perform very similar. These initial result also show that the best performing networks lie close to the result achieved with linear interpolation. Outlier weak performances are due some strong overestimation of slow person velocities and some undefined random predictions when using positions. Hasan et al. reduced this effect by integrating head pose information. We can only remark for the tested networks that this effect can also differ for different runs. Naturally it is important that during training the networks see enough samples from standing of slow moving situations. Excluding such samples through heuristic or probabilistic filtering only helps during application.

There is no network that is clearly performing best, thus the gap between a MLP predictor and a Seq2Seq model is very narrow in the test scenarios. However, besides the factors derived from the data analysis, a prediction of the full path instead of step-wise prediction helps to overcome an accumulation of errors that are fed back into the networks. For the *TrajNet* challenge with a fixed prediction horizon, we thus prefer the RNN-Encoder-MLP over a Seq2Seq model. In the domain of human pose prediction based on RNNs, Li et al [31] reduced this problem with an Auto-Conditioned RNN Network and Martinez et al. [33] propose using a Seq2Seq model along with a sampling-based loss. The TCNs

Table 3. Results for the world plane human-human dataset challenge (*World H-H TrajNet* challenge).

Approach	Overall Average ↓	FDE [m] ↓	ADE [m] ↓	Reference
RED	**0.797**	1.229	0.364	Ours
Social Forces (EWAP)	**0.819**	1.266	0.371	[19]
Predictor SUL	**0.887**	1.374	0.399	
Social Forces (ATTR)	0.904	1.395	0.412	[19]
OSG	1.385	2.106	0.664	
Social LSTM	1.387	2.098	0.675	[3]
Vanilla LSTM	2.107	3.114	1.100	
Occupancy LSTM	2.111	3.12	1.101	
Interactive Gaussian Processes	1.642	1.038	2.245	[13]
Linear Interpolation	0.894	1.359	0.429	
Linear MLP (Pos)	1.041	1.592	0.491	
Linear MLP (Off)	0.896	1.384	0.407	
Non-Linear MLP (Off)	2.103	3.181	1.024	
Linear RNN	0.951	1.482	0.420	
Non-Linear RNN	**0.841**	1.300	0.381	
Linear RNN-Encoder-MLP	0.892	1.381	0.404	
Non-Linear RNN-Encoder-MLP	**0.827**	1.276	0.377	
Linear Seq2Seq	0.923	1.429	0.418	
Non-Linear Seq2Seq	**0.860**	1.331	0.390	
TCN	**0.841**	1.301	0.381	[5]
Gated TCN	0.947	1.468	0.426	[5]

Results highlighted in blue are taken from the *TrajNet* website [39] (http://trajnet.stanford.edu/, accessed 22.06.2018)

perform here similar to RNNs. Since RNNs are more common, also as part of architectures which model interactions (see [3, 4, 17, 44]) to represent single motion, we keep the RNN-Encoder-MLP as our favored model.

4 RNN-Encoder-MLP: RED-predictor

According to the training set analysis and the comparison of architectures the selected model for the *TrajNet* challenge modeling only single human motion is a RNN-Encoder-MLP. In this section, the final design choices, which lead to the submitted predictor which achieved top-rank at the *World H-H TrajNet* challenge, are summarized. The RNN-Encoder as favored model can generalize to deal with varying noisy inputs and is thus able to better capture the person motion compared to the linear interpolation baseline. The main insight is that motion continuity is easier to express in offsets or velocities, because it takes considerably more modeling effort to represent all possible conditioning positions. Especially for the *World H-H TrajNet* challenge, with the different

range for positions in the training and test set, this has significant influence on whether a good performance can be obtained. Instead of using the given input sequence $\mathcal{X}^T = \{(x^t, y^t) \in \mathbb{R}^2 | t = 1, \ldots, t_{obs}\}$ of t_{obs} consecutive pedestrian positions along a trajectory, here the offsets are used for conditioning the network $\mathcal{X}^T = \{(\delta_x^t, \delta_y^t) \in \mathbb{R}^2 | t = 2, \ldots, t_{obs}\}$. Apart from the smaller modeling effort to represent conditioned offsets and the prevention of undefined states due to a suitable data range this domain shift makes data-preprocessing such as the used standardization more reasonable. Since the offset or rather velocity distribution follows a normal distribution around the expected walking speeds of pedestrians compared to the position distribution. In order to deal with the varying discretization artefacts of the ground truth trajectories and make further training easier, smoothed trajectories are used as desired output. Since the prediction length is fixed, the effect of error accumulation during a step-wise prediction is reduced by not feeding back RNN output and applying a full path prediction. Full path integration worked similarly well, but here offsets to the reference positions (last observed position) are predicted. In order to increase the amount of training data, data augmentation is done by reverting all training tracklets. With the combination of all listed factors the proposed simple but effective baseline predictor for the *TrajNet* challenge is ready. In its core the architecture is a Recurrent-Encoder with a dense MLP layer stacked on top. Hence, the predictor is referred to as RED-predictor and can be defined by:

$$h_{encoder}^t = \text{RNN}(h_{encoder}^{t-1}, \delta_{(x,y)}^t; W_{encoder})$$
$$\mathcal{Y}^T = \{(\delta_x^{t+k}, \delta_y^{t+k}) + (x^t, y^t) \in \mathbb{R}^2 | k = 1, \ldots, t_{pred}\} = \text{MLP}(h_{encoder}^t; W_{MLP})$$

Here, $RNN(\cdot)$ is the recurrent network, $h_{encoder}$ the hidden state of the RNN-Encoder with corresponding weight and biases $W_{encoder}$, which is used to generate the full, smoothed path. The multilayer perceptron $MLP(\cdot)$ including the conforming weights and biases W_{MLP} maps the vector $h_{encoder}$ to the coordinate space. The overall architecture is visualized in figure 5
The best achieved result is highlighted in red in table 3. After a fine search for this network, the shown result is produced with a LSTM cell (state size of 32) and one recurrent layer. The proposed predictor was able to produce sophisticated results compared to elaborated models which additionally rely on interaction information such as the model from Helbing and Molnár [19] and the Social-LSTM [3]. Compared to all submitted approaches of the *World H-H TrajNet* 2018 challenge, the RED predictor achieved the best result. All results highlighted in blue were either also officially submitted or provided by the organizers. Nevertheless, the Social-LSTM is one of the first proposed RNN-based architectures which includes human-human interaction and laid the basis for architectures such as presented in the work of Hasan et al.[17] or Xue et al. [44]. Single motion is modeled with an LSTM network. By applying some of the proposed factors to the model, it is expected that the model and equity accordingly model extensions are able to outperform the proposed single motion predictor.

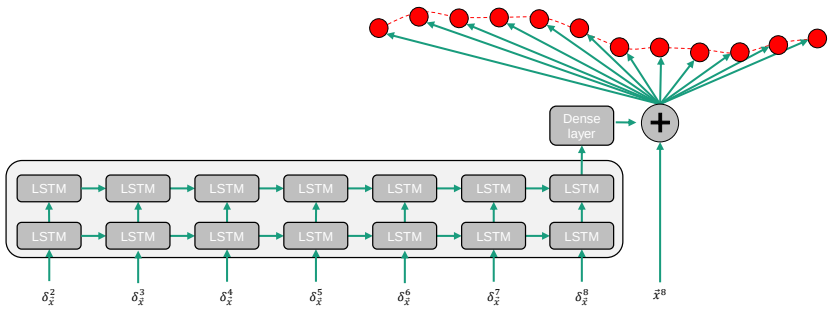

Fig. 5. Visualization of the RED architecture. The conditioning is done for the full initialization sequence $\mathcal{X}^T = \{(\delta_x^t, \delta_y^t) \in \mathbb{R}^2 | t = 2, \ldots, t_8\}$. The internal representation is then used to predict the desired path at once (all 12 positions) using the last observed position (x^8, y^8) as reference for localization.

5 Discussion and Failure Cases

After emphasizing the factors needed in order to achieve sophisticated results based on standard neural networks in the above sections, in this section we discuss some failure cases.
Without exploiting scene-specific knowledge for trajectory prediction, some particular changing behavior in the human motion is not predictable. For example, in the shown tracklet from *SSD Hyang* (see figure 6), there is no cue for a turning maneuver in the initialization tracklet. In order to correct the prediction, new observations are required. All methods tend to predict in such a situation a relatively straight line, resulting in a high prediction error. A scene-independent motion representation is pursuant to better generalize, but for overcoming some limitation in the achievable prediction accuracy, the spatial context is required. The sample tracklet also illustrates the multi-modal nature of the prediction problem. While the person is making a left turn, it is also possible to make a right turn. By using a single maximum-likelihood path the multi-modality of a motion and the uncertainty in the prediction is not covered. The prediction uncertainty can be considered by using the normalized estimation error square (nees) [22], also known as Mahalanobis distance, which corresponds to a weighted Euclidean distance of the errors. But most methods are designed as a regression model, thus for a unified evaluation system the Mahalanobis distance is not applicable. As mentioned, there are a few approaches which include the multi-modal aspect of the problem [27, 29, 24]. Without additional cues of the current scene, these approaches are limited to a fixed scene.

Independent of the question how to include all aspects of a problem in a unified benchmarking, they strongly influence the possible achievable results. The results presented in section 3 show that independent from the model com-

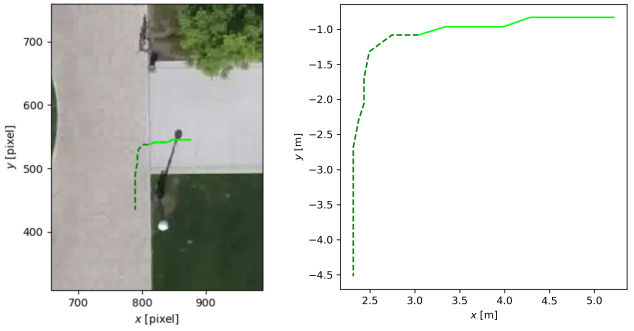

Fig. 6. Example where the scene context strongly influences the person trajectory. The initialization tracklet (solid line) delivers no evidence for a turning maneuver at the intersection. This also shows the multi-modal nature of the prediction problem.

plexity approaches restricted to observing only information from one trajectory are in range to their reachable performance limit on the current dataset repository. Of course due to the fast development in the field of deep neural networks there is still space for improvement, but the current benchmark cannot be completely solved. However, the *TrajNet* challenges also provides human-human and human-space information and recent work such as the approaches of Gupta et al. [16](human-human) or Xua et al. [44] and Sadeghian et al. [40] (human-human, human-space) show possibilities of how to further improve the performance accuracy.

6 Conclusion

In this paper, we presented an evaluation of deep learning approaches for trajectory prediction on *TrajNet* benchmark dataset. The initial results showed that without further cues such as human-human interaction or human-space interaction most basic networks achieve similar results in small range close to a maximum achievable prediction accuracy. By modifying a standard RNN prediction model, we were able to provide a simple but effective RNN architecture that achieves a performance comparable to more elaborated models and achieved the top-rank on the *World H-H TrajNet* 2018 challenge.

Acknowledgements: The authors thank the organizers of the *TrajNet* challenge for providing a framework towards a more meaningful, standardized trajectory prediction benchmarking.

References

1. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X.: TensorFlow: Large-scale machine learning on heterogeneous systems (2015), https://www.tensorflow.org/, software available from tensorflow.org
2. Akaike, H.: Fitting autoregressive models for prediction. Annals of the Institute of Statistical Mathematics **21**(1), 243–247 (1969), https://EconPapers.repec.org/RePEc:spr:aistmt:v:21:y:1969:i:1:p:243-247
3. Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., Savarese, S.: Social LSTM: Human trajectory prediction in crowded spaces. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 961–971. IEEE (2016)
4. Alahi, A., Ramanathan, V., Goel, K., Robicquet, A., Sadeghian, A., Fei-Fei, L., Savarese, S.: Learning to predict human behaviour in crowded scenes. In: Group and Crowd Behavior for Computer Vision. Elsevier (2017)
5. Bai, S., Kolter, J.Z., Koltun, V.: An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv preprint **abs/1803.01271** (2018), http://arxiv.org/abs/1803.01271
6. Ballan, L., Castaldo, F., Alahi, A., Palmieri, F., Savarese, S.: Knowledge Transfer for Scene-Specific Motion Prediction, pp. 697–713. Springer International Publishing (2016). https://doi.org/10.1007/978-3-319-46448-0_42
7. Brownlee, J.: Introduction to Time Series Forecasting with Python: How to Prepare Data and Develop Models to Predict the Future. Jason Brownlee (2017), https://books.google.de/books?id=bA5ItAEACAAJ
8. Cho, K., van Merriënboer, B., Gülçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder–decoder for statistical machine translation. In: Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1724–1734. Association for Computational Linguistics, Doha, Qatar (2014), http://www.aclweb.org/anthology/D14-1179
9. Chung, J., Kastner, K., Dinh, L., Goel, K., Courville, A., Bengio, Y.: A recurrent latent variable model for sequential data. In: Advances in Neural Information Processing Systems (NIPS) (2015)
10. Coscia, P., Castaldo, F., Palmieri, F.A., Alahi, A., Savarese, S., Ballan, L.: Long-term path prediction in urban scenarios using circular distributions. Image and Vision Computing **69**, 81–91 (2018). https://doi.org/https://doi.org/10.1016/j.imavis.2017.11.006, http://www.sciencedirect.com/science/article/pii/S0262885617301853
11. Donahue, J., Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., Darrell, T.: Long-term recurrent convolutional networks for visual recognition and description. In: Conference on Computer Vision and Pattern Recognition. IEEE (2015)
12. Draper, N.R., Smith, H.: Applied regression analysis. Wiley series in probability and mathematical statistics, Wiley, New York (1966)
13. Ellis, D., Sommerlade, E., Reid, I.: Modelling pedestrian trajectory patterns with gaussian processes. In: International Conference on Computer Vision Workshops (ICCVW). pp. 1229–1234. IEEE (2009). https://doi.org/10.1109/ICCVW.2009.5457470

14. Ferryman, J., Shahrokni, A.: Pets2009: Dataset and challenge. In: IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS). pp. 1–6 (2009). https://doi.org/10.1109/PETS-WINTER.2009.5399556
15. Graves, A., Mohamed, A., Hinton, G.: Speech recognition with deep recurrent neural networks. In: International Conference on Acoustics, Speech and Signal Processing. pp. 6645–6649 (2013). https://doi.org/10.1109/ICASSP.2013.6638947
16. Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S., Alahi, A.: Social gan: Socially acceptable trajectories with generative adversarial networks. In: Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (2018)
17. Hasan, I., Setti, F., Tsesmelis, T., Bue, A.D., Galasso, F., Cristani, M.: MX-LSTM: mixing tracklets and vislets to jointly forecast trajectories and head poses. In: Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (2018)
18. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778 (2016). https://doi.org/10.1109/CVPR.2016.90, https://doi.org/10.1109/CVPR.2016.90
19. Helbing, D., Molnár, P.: Social force model for pedestrian dynamics. Phys. Rev. E **51**, 4282–4286 (1995). https://doi.org/10.1103/PhysRevE.51.4282, https://link.aps.org/doi/10.1103/PhysRevE.51.4282
20. Hochreiter, S., Schmidhuber, J.: Long Short-Term Memory. Neural Computation **9**(8), 1735–1780 (1997). https://doi.org/10.1162/neco.1997.9.8.1735
21. Huang, S., Li, X., Zhang, Z., He, Z., Wu, F., Liu, W., Tang, J., Zhuang, Y.: Deep learning driven visual path prediction from a single image. IEEE Transactions on Image Processing **25**(12), 5892–5904 (2016). https://doi.org/10.1109/TIP.2016.2613686
22. Huber, M.: Nonlinear Gaussian Filtering : Theory, Algorithms, and Applications. Ph.D. thesis, Karlsruhe Institute of Technology (KIT) (2015)
23. Hug, R., Becker, S., Hübner, W., Arens, M.: On the reliability of lstm-mdl models for predicting pedestrian trajectories. In: Representations, Analysis and Recognition of Shape and Motion from Imaging Data (RFMI). Savoie, France (2017)
24. Hug, R., Becker, S., Hübner, W., Arens, M.: Particle-based pedestrian path prediction using LSTM-MDL models. In: IEEE International Conference on Intelligent Transportation Systems (ITSC)(accepted) (2018), http://arxiv.org/abs/1804.05546
25. Kalman, R.E.: A new approach to linear filtering and prediction problems. ASME Journal of Basic Engineering **82** (1960)
26. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. International Conference for Learning Representations (ICLR) (2015)
27. Kitani, K.M., Ziebart, B.D., Bagnell, J.A., Hebert, M.: Activity forecasting. In: European Conference on Computer Vision (ECCV). pp. 201–214. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)
28. Kooij, J.F.P., Schneider, N., Flohr, F., Gavrila, D.M.: Context-based pedestrian path prediction. In: European Conference on Computer Vision (ECCV). pp. 618–633. Springer International Publishing (2014). https://doi.org/10.1007/978-3-319-10599-4_40
29. Lee, N., Choi, W., Vernaza, P., Choy, C.B., Torr, P.H.S., Chandraker, M.: Desire: Distant future prediction in dynamic scenes with interacting agents. In: Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (2017)
30. Lerner, A., Chrysanthou, Y., Lischinski, D.: Crowds by example. Computer Graphic Forum **26**(3), 655–664 (2007)

31. Li, Z., Zhou, Y., Xiao, S., He, C., Li, H.: Auto-conditioned LSTM network for extended complex human motion synthesis. arXiv preprint **abs/1707.05363** (2017), http://arxiv.org/abs/1707.05363
32. Ma, W., Huang, D., Lee, N., Kitani, K.M.: Forecasting interactive dynamics of pedestrians with fictitious play. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4636–4644. IEEE (2017). https://doi.org/10.1109/CVPR.2017.493
33. Martinez, J., Black, M.J., Romero, J.: On human motion prediction using recurrent neural networks. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4674–4683. IEEE (2017). https://doi.org/10.1109/CVPR.2017.497, https://doi.org/10.1109/CVPR.2017.497
34. McCullagh, P., Nelder, J.A.: Generalized Linear Models. Chapman & Hall , CRC, London (1989)
35. van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A.W., Kavukcuoglu, K.: Wavenet: A generative model for raw audio. arXiv preprint **abs/1609.03499** (2016), http://arxiv.org/abs/1609.03499
36. Pellegrini, S., Ess, A., Schindler, K., van Gool, L.: You'll never walk alone: Modeling social behavior for multi-target tracking. In: International Conference on Computer Vision. pp. 261–268. IEEE (2009). https://doi.org/10.1109/ICCV.2009.5459260
37. Priestley, M.B.: Spectral analysis and time series. Academic Press, London ; New York : (1981)
38. Robicquet, A., Sadeghian, A., Alahi, A., Savarese, S.: Learning social etiquette: Human trajectory understanding in crowded scenes. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) Computer Vision – ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII. pp. 549–565. Springer International Publishing (2016). https://doi.org/10.1007/978-3-319-46484-8_33
39. Sadeghian, A., Kosaraju, V., Gupta, A., Savarese, S., Alahi, A.: Trajnet: Towards a benchmark for human trajectory prediction. arXiv preprint (2018)
40. Sadeghian, A., Kosaraju, V., Sadeghian, A., Hirose, N., Savarese, S.: SoPhie: An attentive GAN for predicting paths compliant to social and physical constraints. arXiv preprint arXiv:1806.01482 (2018)
41. Vemula, A., Muelling, K., Oh, J.: Modeling cooperative navigation in dense human crowds. In: International Conference on Robotics and Automation (ICRA). pp. 1685–1692. IEEE (May 2017). https://doi.org/10.1109/ICRA.2017.7989199
42. Williams, C.K.I.: Prediction with Gaussian Processes: From Linear Regression to Linear Prediction and Beyond, pp. 599–621. Springer Netherlands, Dordrecht (1998). https://doi.org/10.1007/978-94-011-5014-9_23
43. Xu, K., Ba, J., Kiros, R., K.Cho, Courville, A., Salakhudinov, R., Zemel, R., Bengio, Y.: Show, attend and tell: Neural image caption generation with visual attention. In: Bach, F., Blei, D. (eds.) International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 37, pp. 2048–2057. PMLR, Lille, France (2015)
44. Xue, H., Q., D., Reynolds, H.M.: SS-LSTM: A hierarchical LSTM model for pedestrian trajectory prediction. In: Winter Conference on Applications of Computer Vision (WACV). IEEE (2018)