CF-LSTM: Cascaded Feature-Based Long Short-Term Networks for Predicting Pedestrian Trajectory

Yi Xu,[1] **Jing Yang,**[1*] **Shaoyi Du**[2†]
[1]Institute of Control Engineering, Xi'an Jiaotong University, China
[2]Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, China
colorfulfuture@stu.xjtu.edu.cn, {jasmine1976, dushaoyi}@xjtu.edu.cn

Abstract

Pedestrian trajectory prediction is an important but difficult task in self-driving or autonomous mobile robot field because there are complex unpredictable human-human interactions in crowded scenarios. There have been a large number of studies that attempt to understand humans' social behavior. However, most of these studies extract location features from previous one time step while neglecting the vital velocity features. In order to address this issue, we propose a novel feature-cascaded framework for long short-term network (CF-LSTM) without extra artificial settings or social rules. In this framework, feature information from previous two time steps are firstly extracted and then integrated as a cascaded feature to LSTM, which is able to capture the previous location information and dynamic velocity information, simultaneously. In addition, this scene-agnostic cascaded feature is the external manifestation of complex human-human interactions, which can also effectively capture dynamic interaction information in different scenes without any other pedestrians' information. Experiments on public benchmark datasets indicate that our model achieves better performance than the state-of-the-art methods and this feature-cascaded framework has the ability to implicitly learn human-human interactions.

Introduction

Trajectory prediction of pedestrians has many applications in various fields such as autonomous driving (Lerner, Chrysanthou, and Lischinski 2010), robot navigation (Luo et al. 2018; Pellegrini et al. 2009; Vivacqua et al. 2017; Thrun et al. 2002) and surveillance camera analysis (Leonard and Durrant-Whyte 1990; Trautman and Krause 2010) Estimating the future positions of pedestrians accurately is necessary and beneficial for such tasks (Luber et al. 2010). A big challenge for trajectory prediction of pedestrians is that there are many different interactions which have great influences on the future trajectory occur in the scene. It mainly contains two kinds shown in Figure 1: (1) static interactions caused by stationary obstacles or some other certain objects in the scene, which is straightforward and understandable; (2) dynamic interactions that take place among pedestrians, which is nearly impossible to quantify such complex and often subtle human-human interactions.

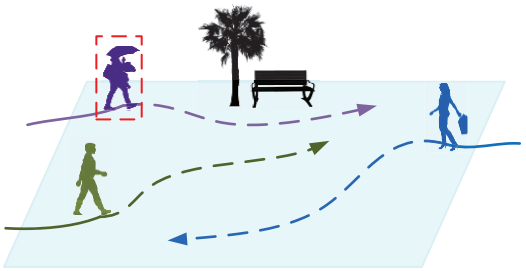


Figure 1: An illustration of a common scene where static interactions and dynamic interactions both occur. The pedestrian who is under an umbrella (framed by red dashed line) has static interactions with stationary obstacles (tree and bench), dynamic interactions with two other pedestrians, which has influences on his/her future trajectory.

Recently, a large number of LSTM-based models (Alahi et al. 2016; Bisagno, Zhang, and Conci 2018) are proposed to try to address this issue, and various mechanisms are designed to learn dynamic interactions (Vemula, Muelling, and Oh 2018; Chandra et al. 2019; Zhang et al. 2019), defining three inputs: (1) feature information of current location, (2) feature information from previous one time step, and (3) feature information of dynamic interactions. Although these models achieve competitive performance, they are still incomplete and far from comprehensive, here are two limitations.

a. Feature information from previous one time step is insufficient.

Usually, hidden state in LSTM from previous one time

[*]Yi Xu and Jing Yang contributed equally to this work.

[†]Corresponding author. This work was supported by the National Key Research and Development Program of China under Grant No. 2017YFA0700800, and the National Natural Science Foundation under Grant No. 61971343, 61773147 and 61573274.

step is used to capture previous spatial features of the pedestrian. However, it only represents the previous location information while the velocity information, is neglected. A simple example is that if one pedestrian stands still on the sidewalk and suddenly begins to walk or suddenly changes the walking velocity, which is common in real world, the previous location information does not have strong connection with his/her future location. It is crucial to acquire the velocity features for trajectory prediction in such cases. Generally, feature information from previous one time step is not adequate in some real situations, and the velocity information is also important.

b. Feature information of neighboring dynamic interactions is not reliable in different scenes.

Many researches extract feature information of neighboring interactions and share across networks by designing an extra layer or a sub-module (Alahi et al. 2016; Zhang et al. 2019; Sadeghian et al. 2019). However, these researches focus on modeling interactions at some instant while neglecting the different characteristics of different scenes. For instance, in a crowed scene, there are a lot of complex human-human interactions, the future trajectory of the observed pedestrian is very likely influenced by such interactions. On the contrary, in a sparse scene, the human-human interactions have a relative smaller impact on his/her future trajectory. Whether dynamic interactions have influences on future trajectory or the degree of such influences in different scenes with different characteristics (e.g. crowded or sparse pedestrian flows) has not been considered by recent studies.

In order to address above two limitations, we propose a novel feature-cascaded framework for LSTM (CF-LSTM) in our paper. We firstly extract the feature information of pedestrians from previous two time steps and then integrate them as one independent input of LSTM. With feature information from previous two time steps, the previous location information as well as the velocity features are both captured. Additionally, instead of measuring complex feature information of human-human interactions, we focus on the internal characteristics of these dynamic interactions. The essence of human-human interactions is to allow pedestrians to change their positions, this change of positions is exactly the external manifestation of these interactions, regardless of different scenes. So, the cascaded feature, which reflects the velocity features, can also implicitly explain the dynamic interactions occur in different scenes.

Contributions are summarized as follows:

- Our proposed framework can capture previous location information, dynamic velocity features, and feature information of dynamic interactions without other pedestrians' information at the same time, which is robust to different scenes with different characteristics.
- State-of-the-art results on public benchmark datasets with different scenes.

Related Work

Research on Trajectory Prediction Task Considering Interactions

A traditional approach having considered both static and dynamic interactions is a named "Social Force" model (Helbing and Molnar 1995), defining interactions as forces upon the pedestrians. Modern socially-aware methods usually use recurrent neural networks (Mikolov et al. 2010) for trajectory prediction (Fernando et al. 2018a; Kitani et al. 2011; Ryoo et al. 2014; Srivastava, Mansimov, and Salakhudinov 2015; Vemula, Muelling, and Oh 2018; Liang et al. 2019; Chung et al. 2014; Mohajerin and Rohani 2019; Liu et al. 2016; Shi et al. 2019) and introduce attention mechanism to interaction measure (Bhattacharyya, Fritz, and Schiele 2018; Choi and Dariush 2019) and social behavior understanding (Sadeghian et al. 2019; Haddad et al. 2019; Al-Molegi, Jabreel, and Martínez-Ballesté 2018; Varshneya and Srinivasaraghavan 2017). Also, Generative Adversarial Network (GAN) model is designed to generate multiple reasonable trajectories (Gupta et al. 2018; Fernando et al. 2018b; Amirian, Hayet, and Pettré 2019).

In comparison, we focus on the essential characteristics and external manifestations of such dynamic interactions instead of measuring them with neighboring information of other pedestrians in the scene.

LSTM models for Sequence Prediction Task

Long Short-Term Memory Networks (LSTM) is a recurrent neural networks variant (Hochreiter and Schmidhuber 1997) in order to solve the gradient vanishment or gradient explosion issue. LSTM has various successful applications in dealing with sequence prediction tasks such as Natural Language Process (Chung et al. 2015; Young et al. 2017), Image Generation (Den Oord, Kalchbrenner, and Kavukcuoglu 2016; Karpathy and Li 2015), Machine Translation (Sutskever, Vinyals, and Le 2014) and so on.

While in our work, we extend the LSTM network as feature extractor to capture features from past trajectories.

Feature-Cascaded Framework

Our proposed feature-cascaded framework is also inspired by the idea of Residual Learning (He et al. 2016; Srivastava, Greff, and Schmidhuber 2015), which can enhance the generalization ability of networks and make easier for deeper networks to optimize (Wang et al. 2017; Huang et al. 2017; 2016; Yang et al. 2018).

Residual learning can obtain historical information, namely feature map, from previous layers, while our proposed feature-cascaded framework is capable of extracting the feature information from previous two time steps and integrate them as an independent input of LSTM networks, which is different in essence.

Methodology

Problem Definition

First, we assume that frames of the video with a fixed interval are preprocessed to obtain the spatial coordinates of

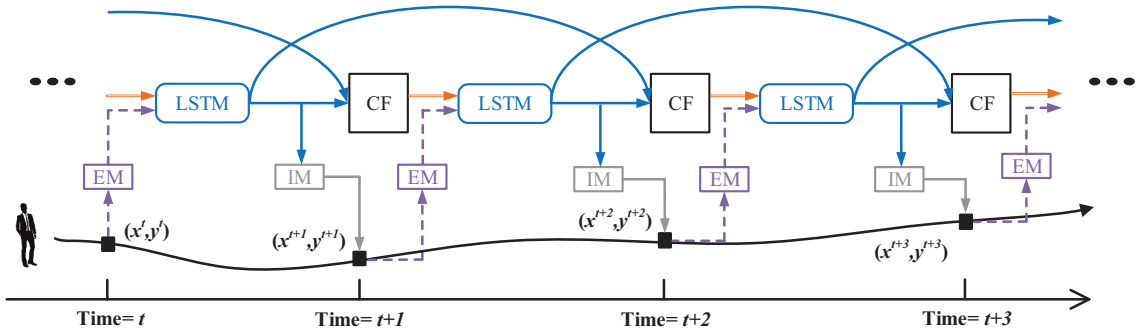


Figure 2: Overview of proposed CF-LSTM. 'EM' represents the extracting module, 'IM' represents the inference module, and 'CF' represents the cascaded feature module.

each pedestrian. We denote the coordinates $(x_i^t, y_i^t) \in \mathbb{R}^2$ of pedestrian i at time t as $\vec{p}_i^t$. Then, we formally describe the trajectory prediction problem as follow. Predict the future trajectory $\Gamma_i = (\vec{p}_i^{obs+1}, ..., \vec{p}_i^{pred})$ of the target pedestrian i from time steps $t = T_{obs+1}, ..., T_{pred}$, taking account his/her own past trajectory $\mathcal{H}_i = (\vec{p}_i^1, ..., \vec{p}_i^{obs})$ from time steps $t = 1, ..., T_{obs}$ and other pedestrians' trajectory in the scene $\{\mathcal{H}_j : j \in 1, 2, ..., N, j \neq i\}$, where N denotes the number of pedestrians in the scene.

Our goal is to learn the parameters W^* of a model $f(\cdot)$ in order to predict the future locations of each pedestrian between $t = T_{obs+1}$ and $t = T_{pred}$. Formally,

$$\Gamma_i = f(\mathcal{H}_1, \mathcal{H}_2, ..., \mathcal{H}_N; W^*) \tag{1}$$

where W^* is the collection of all parameters used in the model.

CF-LSTM Framework

The overview of our proposed framework is illustrated in Figure 2. Our framework consists of three key components: (1) extracting module, (2) cascaded feature module, and (3) inference module.

First, the extracting moudle is designed to capture current location information of pedestrians in the scene. In our case, at time t, we use the vanilla LSTM to encode the coordinates (x_i^t, y_i^t) of pedestrian i. Then, the hidden states h_{t-1} and h_{t-2}, representing the feature information of pedestrian i at time $t-1$ and $t-2$, are extracted and fed into the cascaded feature module. This module is designed to generate the cascaded feature from previous two time steps. Finally, the hidden states h_t at time t pass through the inference module to estimate the locations at time $t+1$.

Extracting Module In order to extract joint features from past locations of all pedestrians, we use the LSTM network to capture the temporal dependency between all states of the pedestrians and encode them into a higher dimensional feature representation.

In our case, the coordinates (x_i^t, y_i^t) of pedestrian i at time t are embedded into a vector $\boldsymbol{e_i^t}$ as follows:

$$\boldsymbol{e_i^t} = \phi\left(x_i^t, y_i^t; W_e\right) \tag{2}$$

where ϕ is the embedding function with ReLU non-linerlity, W_e are the embedding parameters.

The vector $\boldsymbol{e_i^t}$ represents the spatial feature information of the target pedestrian i at time t. And $\boldsymbol{e_i^t}$ is defined as one input of LSTM network in Figure 2.

Cascaded Feature Module We design a cascaded feature module consisting of two steps to obtain the feature information of pedestrians from previous two time steps. Note that the hidden state h_i^t of the LSTM at time t captures the latent representation of the pedestrian i in the scene. At the first step, the hidden states h_i^{t-1} and h_i^{t-2} of pedestrian i at time $t-1$ and $t-2$ are extracted and fed into the cascaded feature module for integration. Assume the dimension of the hidden state is D, and the output of the cascaded feature information module is h_i^{t*} (the dimension of h_i^{t*} is also D), the h_i^{t*} is calculated as follows:

$$h_i^{t*}(m) = \boldsymbol{\alpha}(m) \cdot h_i^{t-1}(m) + \boldsymbol{\beta}(m) \cdot h_i^{t-2}(m) \tag{3}$$

where $m \in \{1, 2, ..., D\}$ is the column index, vector $\boldsymbol{\alpha}$ and vector $\boldsymbol{\beta}$ are integration factors of h_i^{t-1} and h_i^{t-2} respectively.

Equation 3 can be rewritten as:

$$h_i^{t*}(m) = (\boldsymbol{\alpha}(m) + \boldsymbol{\beta}(m)) \cdot h_i^{t-1}(m) + (-\boldsymbol{\beta}(m)) \cdot (h_i^{t-1}(m) - h_i^{t-2}(m)) \tag{4}$$

where the first term represents the previous location information and the second term represents the dynamic velocity information. So the h_i^{t*} extracts not only the previous location information but also the dynamic velocity information.

At the second step, the output h_i^{t*} is defined as the other input, along with the $\boldsymbol{e_i^t}$, to the LSTM network in Figure 2, which is formulated as follows:

$$h_i^{t+1} = \text{LSTM}\left(h_i^{t*}, \boldsymbol{e_i^t}; W_l\right) \tag{5}$$

where W_l are the LSTM parameters.

Table 1: Comparison results of CF-LSTM and P-LSTM

Dataset	Performance (ADE/FDE)	
	CF-LSTM	P-LSTM
ETH	**0.45/0.82**	0.48/0.89
Hotel	**0.49**/0.81	0.50/**0.80**
ZARA01	**0.39/0.30**	0.45/0.37
ZARA02	**0.36/0.50**	0.43/0.60
UCY	**0.47/0.73**	0.48/**0.73**
Average	**0.43/0.63**	0.47/0.68

Inference Module In inference module, we assume a bivariate Gaussian distribution parameterized by the mean $\mu_i^t = (\mu_x, \mu_y)_i^t$, standard deviation $\sigma_i^t = (\sigma_x, \sigma_y)_i^t$, and correlation coefficient ρ_i^t to estimate the predicted coordinates. These parameters at time t are determined by the hidden state h_i^t at time t passing through a linear layer W_o as follows:

$$\left(\mu_i^t, \sigma_i^t, \rho_i^t\right) = W_o h_i^t \tag{6}$$

The predicted coordinates are given by:

$$(\hat{x}_i^t, \hat{y}_i^t) \sim \mathcal{N}\left(\mu_i^t, \sigma_i^t, \rho_i^t\right) \tag{7}$$

Our model is jointly trained by minimizing the negative log-likelihood loss L_i (L_i represents the i^{th} trajectory) as follows:

$$L^i\left(W_e, W_a, W_l, W_o\right) = -\sum_{t=T_{obs}+1}^{T_{pred}} \log\left(\mathbb{P}\left(x_i^t, y_i^t | \sigma_i^t, \mu_i^t, \rho_i^t\right)\right) \tag{8}$$

Note that the loss is calculated over the entire trajectories in the training datasets. We jointly back-propagate through our model at every time step and tuning the parameters to minimize the loss.

Perceptive Feature Module Our proposed cascaded feature module introduced above can integrate feature information from previous two time steps. We also design a different perceptive feature module using a multi-layer perception for integration, which we referred to as P-LSMT. Same as the definitions of dimension in cascaded feature module, the output of perceptive feature module $h_i^{t'}$ is formulated as follows:

$$h_i^{t'} = \gamma(h_i^{t-1}, h_i^{t-2}) \tag{9}$$

where γ represents a MLP.

This can be considered as a variant of the operation in equation 3, and is used to replace the h_i^{t*} in equation 5 while training. The difference between these two integration operation is that, in CF-LSTM, the value in each column of the h_i^{t*} is calculated from corresponding columns of h_i^{t-1} and h_i^{t-2}, but in P-LSTM, this correspondence no longer exists.

Experiments and Analysis

In this section, we demonstrate the experimental results of our approach in two public datasts: ETH (Pellegrini, Ess, and Van Gool 2010) and UCY (Leal-Taixé et al. 2014). The ETH datast has total 750 pedestrians and two scenes (ETH and Hotel). The UCY dataset has totoal 786 pedestrians and three scenes (ZARA01, ZARA02, and UCY). These two datasets are both collected from real world, containing complex situations such as pedestrians walking in groups, non-linear trajectories with different velocities, intentionally avoiding collisions and other challenging behaviors, which is suitable for our experiments.

Evaluation Metrics: Similar to following baselines, we use two following metrics. Assume N is the number of the trajectories in the testing process, $\vec{p}_{i,pred}^t$ represents the predicted spatial coordinates (x^t, y^t) of the i^{th} pedestrian at time t and $\vec{p}_{i,obs}^t$ represents the respective observed location.

- **Average Displacement Error (ADE):** This error calculates the mean distance between all predicted points and the actual points in one trajectory .

$$ADE = \frac{\sum_{i=1}^{N}\sum_{t=T_{obs}+1}^{T_{pred}}\left(\vec{p}_{i,pred}^t - \vec{p}_{i,obs}^t\right)^2}{N\left(T_{pred} - (T_{obs}+1)\right)} \tag{10}$$

- **Final Displacement Error (FDE):** This error calculates the mean distance between the final predicted point and the final actual point at the end of the prediction process T_{pred}.

$$FDE = \frac{\sum_{i=1}^{N}\sqrt{\left(\vec{p}_{i,pred}^t - \vec{p}_{i,obs}^t\right)^2}}{N} \tag{11}$$

Baselines: We compare our model with several representative existing models:

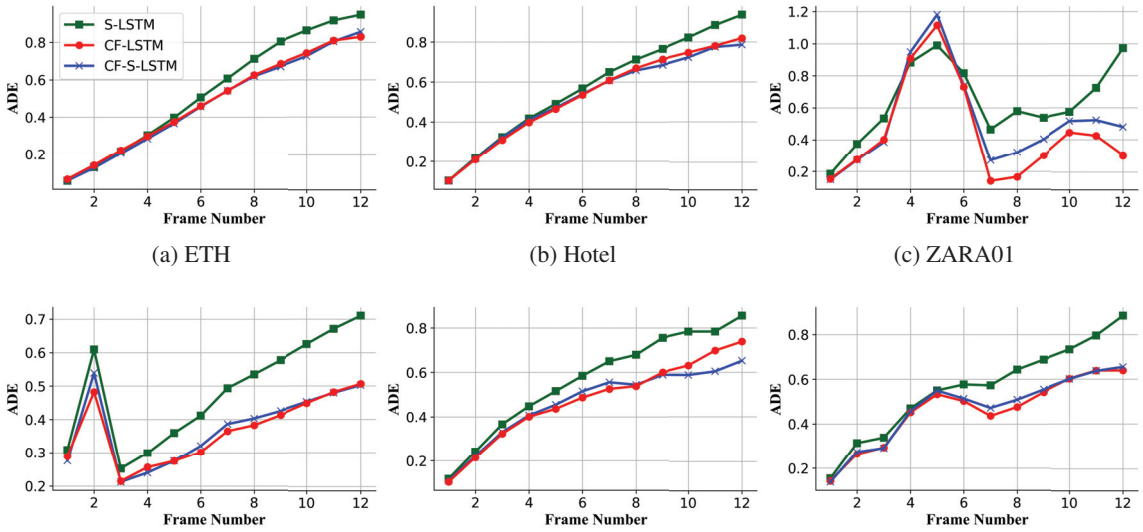
- The vanilla LSTM model (LSTM*). Predict the trajectory with vanilla LSTM.
- Social LSTM model (S-LSTM*) (Alahi et al. 2016). An extra pooling layer is designed to connect LSTM for passing the interaction information among pedestrians.
- Social GAN model (S-GAN*) (Gupta et al. 2018). A generative adversarial network (GAN) is used to generate multiple possible trajectories. It has an extra pooling module to share interaction information among pedestrians.
- SoPhie model (SoPhie*) (Sadeghian et al. 2019). An improved GAN-based model with attention mechanisms on social relationship and physical acceptability.

We try our best to reproduce the Social LSTM model (S-LSTM) following implementation details in the paper(Alahi et al. 2016). In addition, we introduce our proposed framework to the Social LSTM model (CF-S-LSTM) for comparison.

Implementation Details: During training, we use a leave-one-out approach where we train and validate our model on 4 sets and test on the remaining one. During testing, we observe the trajectory for 8 frames and predict the next 12 frames. The frame rate is 0.4, which means $T_{obs} = 3.2secs$, $T_{pred} - T_{obs} = 4.8secs$. We set the dimension of the hidden state D as 128 for all the LSTM models. All the inputs are embedded into a 64 dimensional vector

Table 2: Quantitative results of baselines and our models on all datasets.

Dataset	Performance (ADE/FDE)						
	LSTM*	S-LSTM*	S-GAN*	SoPhie*	S-LSTM	CF-LSTM(Ours)	CF-S-LSTM(Ours)
ETH	1.09/2.41	0.70/1.40	0.81/1.52	0.70/1.43	0.50/0.95	**0.45/0.82**	**0.45**/0.86
Hotel	0.86/1.91	**0.37/0.73**	0.72/1.61	0.76/1.67	0.53/0.94	0.49/0.81	0.49/0.78
ZARA01	0.61/1.31	0.49/1.15	0.34/0.69	**0.30**/0.63	0.54/0.97	0.39/**0.30**	0.46/0.48
ZARA02	0.41/0.88	0.39/0.89	0.42/0.84	0.38/0.78	0.48/0.71	**0.36/0.50**	0.37/**0.50**
UCY	0.52/1.11	0.60/1.32	0.60/1.26	0.54/1.24	0.57/0.85	0.47/0.73	**0.46/0.65**
Average	0.7/1.52	0.51/1.10	0.58/1.18	0.54/1.15	0.52/0.88	**0.43/0.63**	0.45/0.65



(a) ETH (b) Hotel (c) ZARA01

(d) ZARA02 (e) UCY (f) Average

Figure 3: Variations of the error of each frame. The x-axis represents the frame number to be predicted, ranking from 1 to 12, and the y-axis represents the average displacement error.

with ReLU nonlinearity. The batch size is 8 and the model is trained for 150 epochs using Adam with an initial learning rate of 0.001.

During inference process, we use our trained model to determine the parameters of the bivariate Gaussian distribution and then sample from it to obtain the coordinates $(\hat{x}, \hat{y})_i^t$ of the i^{th} pedestrians according to equation 7. From time T_{obs+1} to T_{pred}, we replace the actual coordinates (x_i^t, y_i^t) in equation 2 with the predicted coordinates $(\hat{x}_i^t, \hat{y}_i^t)$ to make predictions.

Quantitative Analysis

Comparison Between CF-LSTM and P-LSTM Table 1 shows the results of our proposed models with two different integration operations. As shown in Table 1, the CF-LSTM outperforms the P-LSTM for $8.5\%/7.4\%$. An viable explanation is that the value in different columns of the hidden state h represents a distinctive feature in higher dimensional feature space, according to equation 3, the integration operation in CF-LSTM does not change such correspondence.

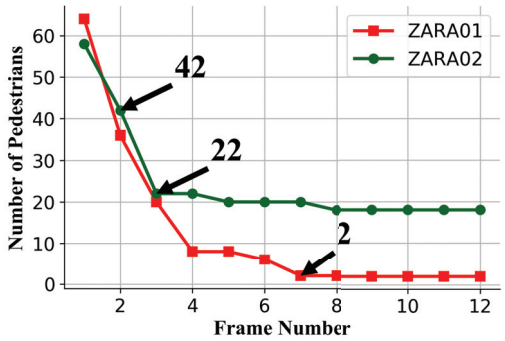


Figure 4: Variations of the number of pedestrians whose trajectories are predicted.

However, it is highly possible that the integration operation in P-LSTM would lead to confusion of features in higher dimensional feature space.

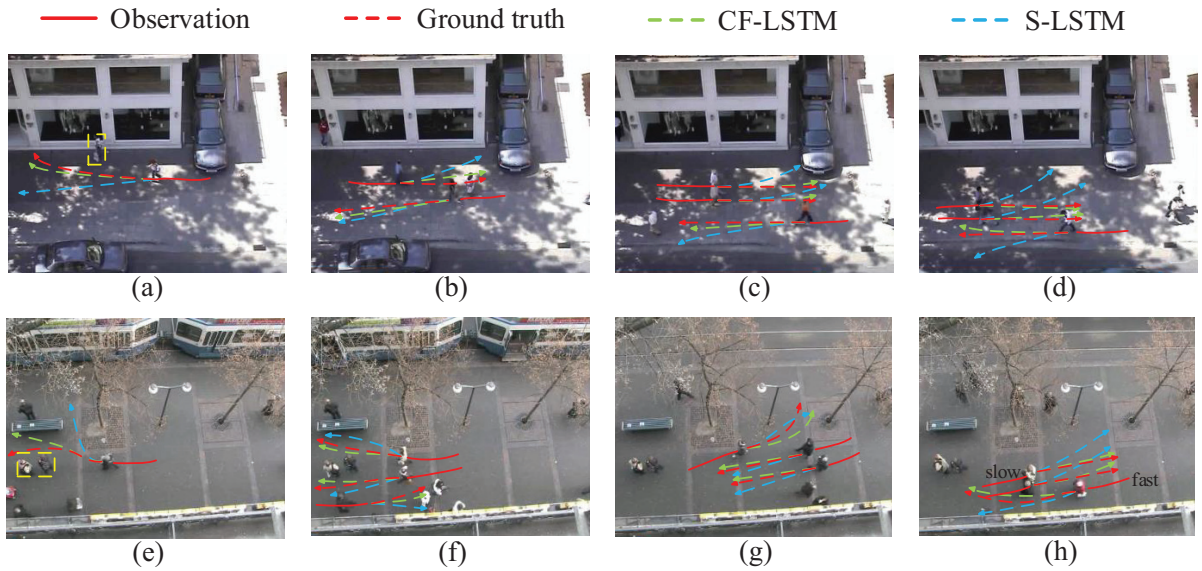


Figure 5: Illustration of predicted trajectories. These examples consist of different complex but common situations in real world, such as collisions avoidance, walking alone or in groups, walking with different velocities and so on.

Comparison Between Baselines and Ours Table 2 indicates the quantitative results of baselines and our models.

S-LSTM* vs S-LSTM: In Table 2, the results of our S-LSTM model implementation are different from what are reported in (Alahi et al. 2016). It is possible because there are so many factors that account, such as the hyper-parameters, the randomness of inference process, the pre-process of datasets and so on. And the ADE between these two models are about the same.

CF-LSTM vs others: In general, our CF-LSTM model achieves the best performance based on ADE and FDE metrics, increasing $38.6\%/58.6\%$, $15.7\%/42.7\%$, $25.9\%/46.6\%$, $20.4\%/45.2\%$, and $17.3\%/28.4\%$ relative to LSTM* model, S-LSTM* model, S-GAN* model, SoPhie* model and S-LSTM model, respectively.

Although interaction information is not directly past through our model, the cascaded feature can still implicitly represent the interaction information because the change of positions is the external performance of interactions. With such feature information considered during training, our model is capable of handling complicated situations within subtle human-human interactions.

S-LSTM vs CF-LSTM vs CF-S-LSTM: We also introduce our cascaded feature into S-LSTM model (CF-S-LSTM). The CF-S-LSTM model outperforms the S-LSTM model for $13.5\%/26.1\%$, but slightly worse than CF-LSTM model. In particular, CF-LSTM model performs better in scene ZARA01 and ZARA02, which are relatively sparser compared with other scenes, and the interactions have relative much smaller influences on future trajectories. Our CF-LSTM model, without directly passing the interaction information, can have a better prediction in sparse scenes.

Also, with the guidance of cascaded feature, CF-S-LSTM model is able to capture the velocity information, which helps to improve the performance of S-LSTM.

ADE of Each Frame Figure 3 plots the change of average displacement error along with the frame number on five datasets. Generally, in Figure 3(f), our proposed CF-LSTM model has the smallest ADE of 12 frame, and with the guidance of cascaded feature, CF-S-LSTM model is better than the original S-LSTM model.

In Figure 3(a), 3(b) and 3(e), the ADE increases when frame number increases, it is congenial with reason and common sense. However, in Figure 3(c) and 3(d), the change is dramatic, the ADE increases at the beginning and then plunges.

In order to find out the reason, the number of pedestrians who participate in the trajectory prediction in each frame is illustrated in Figure 4. The number of pedestrians whose effective trajectories are longer than 20 frames (8 frames for observation and 12 frames for predictions) is small in ZARA01 and ZARA02. In case ZARA01, there are only two pedestrians participating in prediction since frame 6 and these two pedestrians' predicted locations determine the whole ADE. So, the accumulative error before frame 6 is largely eliminated and this situation is consistent with the sudden drop in Figure 3(c). Similar in case ZARA02, from frame 2 to frame 3, the number of pedestrians dramatically reduces from 42 to 22, this sudden change of the number also greatly decreases the accumulative error and it is a possible reason for the sudden drop in Figure 3(d).

Qualitative Analysis

In this section, we visualize some of predicted trajectories in Figure 5.

In the first column of Figure 5 (Figure 5(a), 5(e)) there are pedestrians (framed with yellow dashed line) standing in front of the observed pedestrian, the S-LSTM model and CF-LSTM model are both capable of avoiding the collision. Meanwhile, our proposed CF-LSTM model has smaller error and the direction of predicted trajectory is closer to the ground truth. In the first row (Figure 5(b), 5(c), 5(d)), there are three examples that the observed pedestrians walking in opposite direction alone or in groups where human-human interactions occur. The distance between pedestrians who are walking towards each other is safe enough according to the ground truth, but the trajectories predicted by S-LSTM model indicates that pedestrians want to enlarge the distance between them and walk away from each other, namely the deviation phenomena. Our proposed model, considering the dynamic velocity information, can make better predictions.

In the second row (Figure 5(f), 5(g), 5(h)), there are some examples where situations are more complex, such as avoid collisions along with interactions in Figure 5(f), sudden change of the direction in Figure 5(g), and walking in different velocities in Figure 5(h). It can be seen that our CF-LSTM model has better performance than S-LSTM model in such complicated situations.

Above all, our proposed CF-LSTM model is able to understand human-human interactions without extracting extra other pedestrians' neighboring information. Additionally, with the guidance of velocity information, our model is more robust to different scenes with complicated situations.

Conclusion

In this paper, we propose a novel feature-cascaded framework for LSTM to address the limitations of the pedestrian trajectory prediction. In our work, we extract the feature information from previous two time steps and integrate them as the cascaded feature, which can obtain three kinds of feature information at the same time: (1) previous location information, (2) feature information of dynamic velocity, and (3) feature information of dynamic interactions. Then, we define the cascaded feature along with the current location information as inputs to LSTM for learning. Experiments indicate that our scene-agnostic model achieves better performance than the state-of-the-art methods on public benchmark datasets.

References

Al-Molegi, A.; Jabreel, M.; and Martínez-Ballesté, A. 2018. Move, attend and predict: An attention-based neural model for people's movement prediction. *Pattern Recognition Letters* 112:34–40.

Alahi, A.; Goel, K.; Ramanathan, V.; Robicquet, A.; Fei-Fei, L.; and Savarese, S. 2016. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 961–971.

Amirian, J.; Hayet, J.-B.; and Pettré, J. 2019. Social ways: Learning multi-modal distributions of pedestrian trajectories with gans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 0–0.

Bhattacharyya, A.; Fritz, M.; and Schiele, B. 2018. Long-term on-board prediction of people in traffic scenes under uncertainty. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4194–4202.

Bisagno, N.; Zhang, B.; and Conci, N. 2018. Group lstm: Group trajectory prediction in crowded scenarios. In *European Conference on Computer Vision*, 213–225. Springer.

Chandra, R.; Bhattacharya, U.; Bera, A.; and Manocha, D. 2019. Traphic: Trajectory prediction in dense and heterogeneous traffic using weighted interactions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8483–8492.

Choi, C., and Dariush, B. 2019. Learning to infer relations for future trajectory forecast. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 0–0.

Chung, J.; Gulcehre, C.; Cho, K.; and Bengio, Y. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Chung, J.; Kastner, K.; Dinh, L.; Goel, K.; Courville, A. C.; and Bengio, Y. 2015. A recurrent latent variable model for sequential data. *Neural Information Processing Systems* 2980–2988.

Den Oord, A. V.; Kalchbrenner, N.; and Kavukcuoglu, K. 2016. Pixel recurrent neural networks. In *International Conference on Machine Learning*, 1747–1756.

Fernando, T.; Denman, S.; McFadyen, A.; Sridharan, S.; and Fookes, C. 2018a. Tree memory networks for modelling long-term temporal dependencies. *Neurocomputing* 304:64–81.

Fernando, T.; Denman, S.; Sridharan, S.; and Fookes, C. 2018b. Gd-gan: Generative adversarial networks for trajectory prediction and group detection in crowds. In *Asian Conference on Computer Vision*, 314–330. Springer.

Gupta, A.; Johnson, J.; Fei-Fei, L.; Savarese, S.; and Alahi, A. 2018. Social gan: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2255–2264.

Haddad, S.; Wu, M.; Wei, H.; and Lam, S. K. 2019. Situation-aware pedestrian trajectory prediction with spatio-temporal attention model. *arXiv preprint arXiv:1902.05437*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Helbing, D., and Molnar, P. 1995. Social force model for pedestrian dynamics. *Physical review E* 51(5):4282.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural Computation* 9(8):1735–1780.

Huang, G.; Sun, Y.; Liu, Z.; Sedra, D.; and Weinberger, K. Q. 2016. Deep networks with stochastic depth. In *European conference on computer vision*, 646–661. Springer.

Huang, G.; Liu, Z.; Van Der Maaten, L.; and Weinberger, K. Q. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4700–4708.

Karpathy, A., and Li, F. F. 2015. Deep visual-semantic alignments for generating image descriptions. In *Computer Vision & Pattern Recognition*, 3128–3137.

Kitani, K. M.; Okabe, T.; Sato, Y.; and Sugimoto, A. 2011. Fast unsupervised ego-action learning for first-person sports videos. In *CVPR 2011*, 3241–3248. IEEE.

Leal-Taixé, L.; Fenzi, M.; Kuznetsova, A.; Rosenhahn, B.; and Savarese, S. 2014. Learning an image-based motion context for

multiple people tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3542–3549.

Leonard, J. J., and Durrant-Whyte, H. F. 1990. Application of multi-target tracking to sonar-based mobile robot navigation. In *29th IEEE Conference on Decision and Control*, 3118–3123. IEEE.

Lerner, A.; Chrysanthou, Y.; and Lischinski, D. 2010. Crowds by example. *Computer Graphics Forum* 26(3):655–664.

Liang, J.; Jiang, L.; Niebles, J. C.; Hauptmann, A. G.; and Fei-Fei, L. 2019. Peeking into the future: Predicting future person activities and locations in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5725–5734.

Liu, Q.; Wu, S.; Wang, L.; and Tan, T. 2016. Predicting the next location: A recurrent model with spatial and temporal contexts. In *Thirtieth AAAI Conference on Artificial Intelligence*.

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

Luo, Y.; Cai, P.; Bera, A.; Hsu, D.; Lee, W. S.; and Manocha, D. 2018. Porca: Modeling and planning for autonomous driving among many pedestrians. *IEEE Robotics & Automation Letters* PP(99):1–1.

Mikolov, T.; Karafiát, M.; Burget, L.; Černockỳ, J.; and Khudanpur, S. 2010. Recurrent neural network based language model. In *Eleventh annual conference of the international speech communication association*.

Mohajerin, N., and Rohani, M. 2019. Multi-step prediction of occupancy grid maps with recurrent neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 10600–10608.

Pellegrini, S.; Ess, A.; Schindler, K.; and Gool, L. J. V. 2009. You'll never walk alone: Modeling social behavior for multi-target tracking. In *IEEE International Conference on Computer Vision*, 261–268.

Pellegrini, S.; Ess, A.; and Van Gool, L. 2010. Improving data association by joint modeling of pedestrian trajectories and groupings. In *European conference on computer vision*, 452–465. Springer.

Ryoo, M.; Fuchs, T. J.; Xia, L.; Aggarwal, J.; and Matthies, L. 2014. Early recognition of human activities from first-person videos using onset representations. *arXiv preprint arXiv:1406.5309*.

Sadeghian, A.; Kosaraju, V.; Sadeghian, A.; Hirose, N.; Rezatofighi, H.; and Savarese, S. 2019. Sophie: An attentive gan for predicting paths compliant to social and physical constraints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1349–1358.

Shi, X.; Shao, X.; Guo, Z.; Wu, G.; Zhang, H.; and Shibasaki, R. 2019. Pedestrian trajectory prediction in extremely crowded scenarios. *Sensors* 19(5):1223.

Srivastava, R. K.; Greff, K.; and Schmidhuber, J. 2015. Training very deep networks. In *Advances in neural information processing systems*, 2377–2385.

Srivastava, N.; Mansimov, E.; and Salakhudinov, R. 2015. Unsupervised learning of video representations using lstms. In *International conference on machine learning*, 843–852.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. *neural information processing systems* 3104–3112.

Thrun, S.; Bennewitz, M.; Burgard, W.; Cremers, A. B.; Dellaert, F.; Fox, D.; Hahnel, D.; Rosenberg, C.; Roy, N.; and Schulte, J. 2002. Minerva: A second-generation museum tour-guide robot. In *IEEE International Conference on Robotics & Automation*, volume 3, 1999–2005.

Trautman, P., and Krause, A. 2010. Unfreezing the robot: Navigation in dense, interacting crowds. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 797–803. IEEE.

Varshneya, D., and Srinivasaraghavan, G. 2017. Human trajectory prediction using spatially aware deep attention models. *arXiv preprint arXiv:1705.09436*.

Vemula, A.; Muelling, K.; and Oh, J. 2018. Social attention: Modeling attention in human crowds. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 1–7. IEEE.

Vivacqua, R. P. D.; Bertozzi, M.; Cerri, P.; Martins, F. N.; and Vassallo, R. F. 2017. Self-localization based on visual lane marking maps: An accurate low-cost approach for autonomous driving. *IEEE Transactions on Intelligent Transportation Systems* PP(99):1–16.

Wang, F.; Jiang, M.; Qian, C.; Yang, S.; Li, C.; Zhang, H.; Wang, X.; and Tang, X. 2017. Residual attention network for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3156–3164.

Yang, Y.; Zhong, Z.; Shen, T.; and Lin, Z. 2018. Convolutional neural networks with alternately updated clique. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2413–2422.

Young, T.; Hazarika, D.; Poria, S.; and Cambria, E. 2017. Recent trends in deep learning based natural language processing. *ArXiv: Computation and Language*.

Zhang, P.; Ouyang, W.; Zhang, P.; Xue, J.; and Zheng, N. 2019. Sr-lstm: State refinement for lstm towards pedestrian trajectory prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 12085–12094.