SR-LSTM: State Refinement for LSTM towards Pedestrian Trajectory Prediction

Pu Zhang [1], Wanli Ouyang [2], Pengfei Zhang [1], Jianru Xue [1][*], Nanning Zheng [1]

[1] Institute of Artificial Intelligence and Robotics, Xian Jiaotong University, China

[2] The University of Sydney, SenseTime Computer Vision Research Group, Australia

zhangpu2016,zpengfei@stu.xjtu.edu.cn,

jrxue,nnzheng@mail.xjtu.edu.cn,

wanli.ouyang@sydney.edu.au

Abstract

In crowd scenarios, reliable trajectory prediction of pedestrians requires insightful understanding of their social behaviors. These behaviors have been well investigated by plenty of studies, while it is hard to be fully expressed by hand-craft rules. Recent studies based on LSTM networks have shown great ability to learn social behaviors. However, many of these methods rely on previous neighboring hidden states but ignore the important current intention of the neighbors. In order to address this issue, we propose a data-driven state refinement module for LSTM network (SR-LSTM), which activates the utilization of the current intention of neighbors, and jointly and iteratively refines the current states of all participants in the crowd through a message passing mechanism. To effectively extract the social effect of neighbors, we further introduce a social-aware information selection mechanism consisting of an element-wise motion gate and a pedestrian-wise attention to select useful message from neighboring pedestrians. Experimental results on two public datasets, i.e. *ETH and UCY, demonstrate the effectiveness of our proposed SR-LSTM and we achieve state-of-the-art results.*

1. Introduction

Pedestrian trajectory prediction is strongly required by various applications, *e.g.*, autonomous driving and robot navigation. The trajectory of pedestrian can be influenced by multiple factors such as scene topologies, pedestrian beliefs, and the most complex one, human-human interactions. The intricate and subtle interactions are often taken place among the pedestrians. For example, strangers avoid collisions, but fellows walk in group. Broken groups can re-
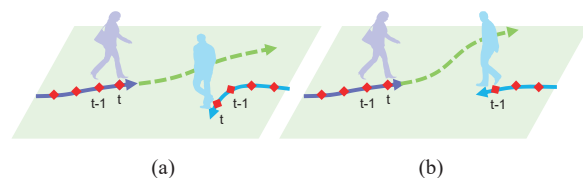
---

[*]Corresponding author.

Figure 1. When predicting for the lady at time t, considering the trajectory of the man on the right up to time t (a), or the one up to time $t-1$ (b), can cause great deviation in predicting results (dashed lines).

group to keep the unity [10, 30]. When individuals meet groups, singles are statistically walking faster and are more likely to adjust their routes [7, 11]. Stationary groups act as obstacles [50, 51].

Although various social behaviors have been investigated, it is challenging to take a comprehensive consideration of them. Some recent data-driven methods [1, 12, 13, 35, 38, 39, 41, 46] try to leverage from Long-Short Term Memory networks (LSTM) [16], to learn social behaviors from large scale data. In this paper, we point out two factors which are important but neglected in different levels:

1). *Current states of neighbors are important for timely interaction inference.*

Many of the recent RNN-based approaches make use of the previous hidden states of neighbors [1, 12, 13, 35, 38, 39]. However, the previous states fail to reveal the newest status of neighbors especially when they have just change their intentions in short time period. This effect of lagging depends on the size of the time step. Within a common time step in recent works [1, 12, 35], *e.g.*, 0.4s, human can take one stride, in which the intentions of them could change unexpectedly. Fig. 1 shows an example. The man on the right in Fig. 1(a) changes his intention to turn left at time t. Based on this observation, the predictions of the lady can be straight on or turning slightly. But if the algorithm only
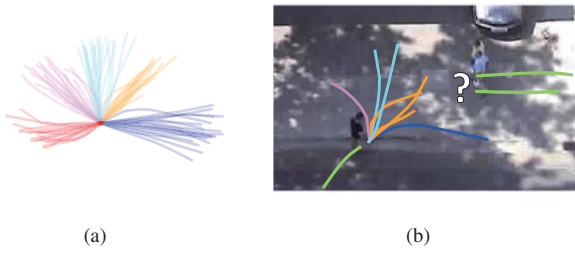
(a)   (b)

Figure 2. (a) Activation trajectory patterns of hidden neurons in L-STM, which start from the origin. Each trajectory pattern marked by certain color contains trajectories from database which has top-20 responses for the hidden neuron. (b) A sample of three pedestrian interaction. How will the dyad pay attention to the other pedestrian on the left?

considers the neighbors' trajectory till $t - 1$ (Fig. 1(b)), the man tends to go straight and forces the lady to largely turn and avoid collision, which results in large prediction error. Therefore, we are motivated to take advantage of the current neighboring states into consideration.

2). *Useful information should be adaptively selected from neighbors, based on their motions and locations.*

Neural networks, *e.g.*, LSTM, can be used for extracting the features representing the trajectory. To better explain these features, Fig. 2(a) visualizes the trajectory pattern captured by each feature in the LSTM. It can be seen that these neurons are responsible for various motion patterns covering the walking direction and speed. Many approaches utilize the features of neighboring pedestrians to estimate the trajectory of a pedestrian. However, the features (motion patterns) of neighboring pedestrians are not equally important for predicting the trajectory of a pedestrian. As shown in Fig. 2(b), the two pedestrians on the right mostly pay more attention to the situation of collision, which can be represented by the trajectory features of the other pedestrian on the left walking towards them. This potential attention depends on the pairwise motion and relative location of the pedestrian to be predicted and his neighbor. Notably, each neighbor should be particularly treated because different kinds of attention should be assigned to pedestrians according to different interaction conditions. Based on this motivation, we introduce a motion gate to select the most useful features from each neighbor, based on the pairwise motion character and relative location.

In this paper, we propose a states refinement module for LSTM (SR-LSTM), which aligns all pedestrians together and adaptively refine the state of each participant through a message passing framework. Further more, the refinement process can be performed for multiple times, indicating deeper interactions among the crowds. SR-LSTM focuses on the adjustment for current LSTM states, which is quite

different from existing RNN-based approaches. To adaptively extract social effects from neighbors for feature refinement, we further introduce a social-aware information selection mechanism, consisting of an element-wise motion gate and a pedestrian-wise attention layer.

Contributions of this paper are summarized as follows:

- A novel interactive recurrent structure, SR-LSTM, is proposed as a new pipeline for jointly predicting the future trajectories of pedestrians in the crowd.
- SR-LSTM aligns all pedestrians in the scene to adaptively refine the current states of each other. The refinement can be performed for multiple times to model the deep interactions between humans.
- Motion gate is introduced to effectively focus on the most useful neighborhood features.

2. Related Work

Research on human-human interaction. Early work from Helbing and Molnar [15] models the interaction between humans as "social force", which is proved to be effective and applied to crowd analysis [14, 31] and robotics [9, 34]. Succeeding methods take more potential factors into account, such as pedestrian attributes [50, 54], walking group [32, 47], stationary group [50, 51]. Some studies based on game theory model the interaction among pedestrian flows [18, 52] and evacuation process [3, 17, 55], Ma *et al.* [29] predict pedestrians from a static frame using fictitious play. Most of these methods are based on hand-craft functions and rules, which might fail to generalize for more complex interaction cases.

RNNs based approaches for trajectory prediction. Recently, Recurrent Neural Networks (RNN) and its variant structures such as LSTM [16] and Gated Recurrent Units (GRU) [6] is widely used in various tasks including pedestrian trajectory forecasting [1, 2, 12, 13, 20, 22, 35, 37–39, 41, 46], where each pedestrian is modeled by RNN with shared parameters. In order to model the human interactions, researchers follow two primary ways to involve information of neighbors, using their current observations [12, 22, 35, 38, 41] (such as velocity, location, *etc*.) or introducing previous states into current RNN recursion [1, 12, 13, 22, 35, 37–39]. These methods treat the information of neighboring pedestrians as input which serves in an input-to-output mechanism. In comparison, we treat the information from neighboring pedestrians as message provider and construct a message passing mechanism to refine the features of each other. Therefore, our approach uses the information from current time step and can refine the information through multiple message passing iterations.

Attention based approaches for trajectory prediction. Attention mechanisms have been proven to be significantly effective for relevant data selection in various tasks [24, 40, 43, 45]. Some RNN-based works for pedestrian trajectory

prediction utilize the attention mechanism to distinguish the importance of different neighbors [8,35,37,38,41]. Vemula *et al.* [41] compute a soft attention score from the hidden states of the designed edgeRNNs, which gives an importance value for each neighbor. Sadeghian *et al.* [35] utilize the soft attention similar with [45] to highlight the important neighbors. Su *et al.* [37, 38] calculate the pairwise velocity correlation, and emphasis the neighbors who are in similar velocity. However, our motion gate aims to selects motion features from each neighboring pedestrian during the refinement, which can extract more socially aware neighboring features and has not been employed in previous approaches.

Graph-based and message passing framework. This work is also inspired by Graph Convolution Networks (GCN) [4, 21] and message passing frameworks used for other applications such as object detection [19, 53], action recognition [36, 48], semantic segmentation [27, 28], scene graph generation [25, 26, 44, 49], video recognition [42], *etc.*

Our method treats the pedestrian walking space as a fully connected graph and which can be regarded as a variant of GCN specially designed for the trajectory prediction task. We consider message passing for pedestrians within constrained regions, and use pairwise motion character and relative spatial location between pedestrians for guiding message passing.

3. Method

Problem formulation In this paper, we address the problem of pedestrian trajectory prediction in the crowd scenes. We focus on the two-dimensional spatial coordinates at specific time intervals. For the given observed trajectories including T_{obs} frames and N pedestrians, the trajectory point of the ith pedestrian on the tth frame is represented by (x_i^t, y_i^t). The problem is defined to predict the future trajectories $(\hat{x}_i^t, \hat{y}_i^t)$, where $t = T_{obs}+1, T_{obs}+2, \ldots$

3.1. Vanilla LSTM

Vanilla LSTM (V-LSTM) model infers all pedestrian independently, without considering the interactions among them. At time t, the location of the ith pedestrian is embedded as a vector $e_i^t = \phi_e(x_i^t, y_i^t; W_e)$, where ϕ_e is the embedding function parameterized by W_e. The vector e_i^t is used as the input to the LSTM cell as follows:

$$\begin{aligned}
g_i^{u,t} &= \delta(W^u e_i^t + U^u h_i^{t-1} + b^u), \\
g_i^{f,t} &= \delta(W^f e_i^t + U^f h_i^{t-1} + b^f), \\
g_i^{o,t} &= \delta(W^o e_i^t + U^o h_i^{t-1} + b^o), \\
g_i^{c,t} &= \tanh(W^c e_i^t + U^c h_i^{t-1} + b^c), \\
c_i^t &= g_i^{f,t} \odot c_i^{t-1} + g_i^{u,t} \odot g_i^{c,t}, \\
h_i^t &= g_i^{o,t} \odot \tanh(c_i^t),
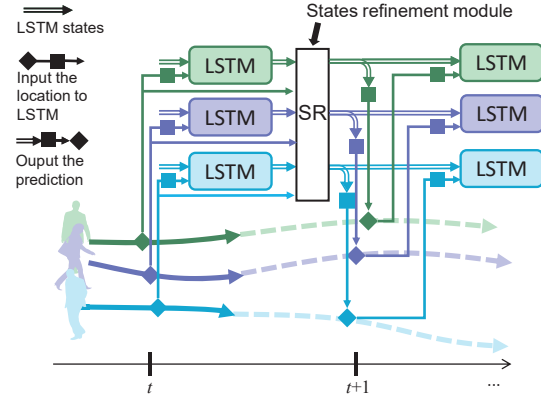\end{aligned} \tag{1}$$

States refinement module

Figure 3. Framework overview of proposed SR-LSTM. States refinement module is considered as an additional subnetwork of the LSTM cells, which aligns pedestrians together and updates current states of them. The refined states are used to predict the location at the next time step.

where g denotes the gate function inside the LSTM cell, the superscripts $u,f,o,$ and c denote the update gate, forget gate, output gate and cell gate, respectively. W and U denote the weight matrix connecting input and hidden state to the LSTM cell. A pedestrian will be treated as a sample when using LSTM. All LSTM parameters are shared across pedestrians.

With the hidden states h_i^t extracted from LSTM, we directly predict the coordinates at time step $t + 1$ following [12]:

$$[\hat{x}_i^{t+1}, \hat{y}_i^{t+1}]^{\mathrm{T}} = W_p h_i^t, \tag{2}$$

where W_p is the learned parameter. The parameters of the LSTM model are directly learned by minimizing the L2 loss between predicted position and ground truth. In the inference stage, the coordinates predicted from the previous time step are used as the input at the current time step.

3.2. The SR-LSTM Framework

The overview of SR-LSTM framework is illustrated in Fig. 3. In this framework, the LSTM in Section 3.1 is used for extracting features from the trajectory of each pedestrian separately. The main difference is that the States Refinement (SR) module is used for refining the *i.e.* cell states c_i^t in Eq. 1 by passing message among pedestrians.

The SR module takes the following three information sources of all pedestrians as input: the current locations of pedestrians, hidden states and cell states from LSTM. The output of the SR module is the refined cell states. Mathematically, the SR module for refining the cell states can be formulated as follows:

$$\hat{c}_i^{t,l+1} = \sum_{j \in \mathcal{N}(i)} M_j(\hat{h}_j^{t,l}, \hat{h}_i^{t,l}) + \hat{c}_i^{t,l}, \tag{3}$$

where M is the message passing function detailed in Section 3.2.2. $\mathcal{N}(i)$ denotes the neighbors of pedestrian i. For the ith pedestrian, the hidden states $\hat{h}_j^{t,l}$ from neighboring pedestrians for $j \in \mathcal{N}(i)$ are integrated through the function M and then combined with the cell state of i to obtain the refined cell state. Message passing can be done for multiple times. l denotes the message passing iteration index. The states with $l = 0$ are initialized by the original LSTM states in Eq. 1.

After the cell states are refined by L refinement iterations in the SR module, they are used for producing predicted coordinates as follows:

$$\hat{c}_j^t = c_j^{t,L}.$$
$$\hat{h}_i^t = g_i^{o,t} \odot \tanh(\hat{c}_i^t), \tag{4}$$

$$[\hat{x}_i^{t+1}, \hat{y}_i^{t+1}]^{\mathrm{T}} = W_p \hat{h}_i^t, \tag{5}$$

where $g_i^{o,t}$ is from the LSTM. In the task of pedestrian trajectory prediction, further refinement could improve the quality of the interaction model, indicating the intention negotiation of human interaction natures.

3.2.1 A simple implementation of message passing

A simple implementation of message passing can be formulated as follows:

$$\hat{c}_i^{t,l+1} = \sum_{j \in \mathcal{N}(i)} W^{mp} \hat{h}_j^{t,l} / |\mathcal{N}_i| + \hat{c}_i^{t,l}, \tag{6}$$

where $|\mathcal{N}_i|$ denotes the number of elements in $\mathcal{N}(i)$. Message passing function $M_j(\hat{h}_j^{t,l}, \hat{h}_i^{t,l}) = W^{mp} \hat{h}_j^{t,l} / |\mathcal{N}(i)|$ does not depend on $\hat{h}_i^{t,l}$ in this formula. W^{mp} is a linear transformation using for transmitting the message from neighboring pedestrians to the pedestrian i.

When using the features from other pedestrians, treating all their features equally is not an appropriate solution. We design more effective message passing term M in following section.

3.2.2 Social-aware information selection

To adaptively focus on the most useful neighboring information and guide the message passing, we design the following message passing term M with a social-aware information selection mechanism:

$$\hat{c}_i^{t,l+1} = \sum_{j \in \mathcal{N}(i)} M_j(\hat{h}_j^{t,l}, \hat{h}_i^{t,l}) + \hat{c}_i^{t,l},$$
$$= \sum_{j \in \mathcal{N}(i)} W^{mp} \alpha_{i,j}^{t,l} \cdot (g_{i,j}^{m,t,l} \odot \hat{h}_j^{t,l}) + \hat{c}_i^{t,l}, \tag{7}$$

where $\odot$ denotes the element-wise product operation. As that in Eq. 6, W^{mp} is the linear transform parameter. The

pedestrian-wise attention $\alpha_{i,j}$ and motion gate $g_{i,j}$ in Eq. 7 are introduced below.

Pedestrian-wise attention. $\alpha_{i,j}$ in Eq. 7 is a scalar. It is the attention for pedestrian j formulated as follows:

$$u_{i,j}^{t,l} = w^{a\mathrm{T}}[r_{i,j}^{t,l}; \hat{h}_j^{t,l}; \hat{h}_i^{t,l}],$$
$$\alpha_{i,j}^{t,l} = \frac{\exp(u_{i,j}^{t,l})}{\sum_k \exp(u_{i,k}^{t,l})}, \tag{8}$$

where $r_{i,j}^{t,l}$ is the relative spatial location, which is an important factor to guide the information selection. It is embedded by embedding function ϕ_r as follows:

$$r_{i,j}^{t,l} = \phi_r(x_i^t - x_j^t, y_i^t - y_j^t; W^r), \tag{9}$$

where (x_i^t, y_i^t) is the location of pedestrian i at time t, similarly for (x_j^t, y_j^t). W^r denotes the parameters for the embedding function ϕ_r.

Motion gate. $g_{i,j}^m$ is a vector, which is formulated as:

$$g_{i,j}^{m,t,l} = \delta(W^m[r_{i,j}^{t,l}; \hat{h}_j^{t,l}; \hat{h}_i^{t,l}] + b^m), \tag{10}$$

where W^m, b^m are parameters and δ denotes the sigmoid function. $g_{i,j}^{m,t,l}$ selects features from $\hat{h}_j^{t,l}$ by using the element-wise product in Eq. 7.

The motion gate and the pedestrian-wise attention have different functionalities and jointly select the important information from neighboring pedestrians for message passing. Further explanation of these two components is as follows:

- The motion gate $g_{i,j}^m$ acts on each hidden state $\hat{h}_j^t$ to perform a pairwise feature selection. It is calculated based on the combination of $r_{i,j}^t$, $\hat{h}_j^t$, $\hat{h}_i^t$ (see Eq. 10), which suggests that the motion of pedestrian i and j and their relative spatial location are jointly considered for feature selection. This element-wise feature selection can not be provided by the pedestrian-wise attention.

- The pedestrian-wise attention is to emphasize important neighbors and control the amount of neighborhood message. If we only take the motion gate, training process could hardly converge due to the uncertain number of correlated neighbors.

- The simple implementation in Eq. 6, which assigns equal weights for all pedestrians and their features, performs worse than social-aware information selection, because the simple implementation does not pay sufficient attention to important neighbors and important trajectories extracted by the features.

4. Experiments

4.1. Datasets and Metrics

We evaluate our proposed model on two public pedestrian walking datasets, ETH [33] and UCY [23], which con-

ID	Pre-processing			Performance (MAD/FAD)					
	Rela/Nabs	EUf	RR	ETH-univ	ETH-hotel	UCY-zara01	UCY-zara02	UCY-univ	AVG
1	Rela	-	-	1.16/2.29	0.57/1.07	0.68/1.39	0.61/1.27	0.76/1.60	0.76/1.52
2	Nabs	-	-	1.00/2.04	0.50/1.08	0.58/1.30	0.40/0.87	0.64/1.38	0.63/1.33
3	Nabs	✓	-	0.84/1.90	0.45/0.94	0.43/0.94	0.38/0.87	0.63/1.42	0.55/1.21
4	Nabs	✓	✓	0.83/1.77	0.41/0.80	0.49/1.15	0.37/0.85	0.56/1.22	0.53/1.16

Table 1. Performance on V-LSTM with different data pre-processings. **Rela**: differentiate the sequences as relative spatial offsets. **Nabs**: use the absolute position but shift the origin to the latest observed time slot. **EUf**: frame rate correction on ETH-univ. **RR**: random rotation for each data mini-batch. We adopt the configuration of ID 4 for our experiments.

tain rich social interactions. These two datasets contain 5 crowd sets, including ETH-univ, ETH-hotel, UCY-zara01, UCY-zara02 and UCY-univ. There are 1536 pedestrians in total with thousands of non-linear trajectories. We evaluate our model on these 5 datasets. We follow the leave-one-out evaluation methodology in [12].

There are two types of metrics for evaluating the performance of trajectory prediction, including the Mean Average Displacement (MAD) error and Final Average Displacement (FAD) error [33] in meters.

- MAD: Mean Euclidean distance between ground truth and predict points of all predicted time steps.

- FAD: Euclidean distance between ground truth and predicted point of the last frame.

The interval of trajectory sequences is set to 0.4 seconds. We take 8 ground truth positions as observation, and predict the trajectories of following 12 time steps, which follows the setting of [1, 12, 33].

4.2. Implementation Details

We use single layer MLP to embed the input vectors to 32 dimensions, and set the dimension of LSTM hidden state as 64. A sliding time window with a length of 20 and a stride size of 1 is used to get the training samples. All trajectory segments in the same time window are regarded as a mini-batch, as they are processed in parallel. We set the size of mini-batch to 8 during the training stage. We use the single-step mode for training (Fig. 4 (a)), and multi-step mode for validating and testing (Fig. 4 (b)). Adam optimizer is adopted to train models in 300 epochs, with an initial learning rate of 0.001. For training the model with multiple states refinement layers, we fixed all basic parameters and only learn the parameters of the additional refinement layer.

4.3. Ablation Study

4.3.1 Data pre-processing

We detail our pretreatment as follows:

- Relative position or normalized absolute position (Rela/Nabs): Two alternative ways of pre-processing,

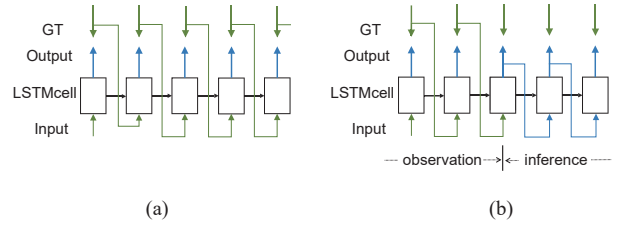(a)   (b)

Figure 4. Two kinds of teaching mode. (a) Single-step mode. Current ground truth (GT) annotation is given to the next time step as input. (b) Multi-step mode, where the current output is used as the input of next time step at inference stage.

differentiate the trajectory as relative location offset or shift the origin to the latest observed time step.

- ETH-univ frame rate issue (EUf): For ETH-Univ scenario, the original video from [5] is an accelerated version. We treat every 6 frames as 0.4s, rather than 10 frames in [12].

- Random Rotation (RR): For one mini-batch, random rotation is employed for data augmentation.

Table 1 shows the results of different data pre-processings on V-LSTM, which shows that: 1) Normalized absolute location is superior to relative position in our trials. 2) Correction of the ETH-Univ frame rate significantly promotes for about 12.7/9(%). 3) Random rotation is also helpful for reducing overfitting. We adopt data pre-processing configuration of ID4 and use the result of which as baseline.

4.3.2 Component analysis

We analyze the components of the proposed model, including the Motion Gate (MG) (Eq. 10), the Pedestrian-wise Attention layer (PA) (Eq. 8), and the number of refinement layers (L). As we consider the finite neighborhood region, we also take the region size as a variable denoted as Neighborhood Size (NS) in meters. To testify the efficiency of the utilize of current neighboring feature, we also consider using Current or Pervious(C/P) hidden states in Eq. 7. For all variants without PA, we divide the number of neighbors on

Variant	Components					Performance (MAD/FAD)					
ID	MG	PA	NS	L	C/P	ETH-univ	ETH-hotel	UCY-zara01	UCY-zara02	UCY-univ	AVG
1	-	-	2	1	C	0.76/1.64	0.37/0.77	0.44/0.97	0.37/0.82	0.55/1.21	0.50/1.08
2	-	-	10	1	C	0.79/1.71	0.41/0.89	0.47/1.07	0.38/0.85	0.56/1.27	0.52/1.16
3	✓	-	10	1	C	0.69/1.35	0.40/0.83	0.43/0.95	0.36/0.80	0.53/1.16	0.48/1.02
4	-	✓	10	1	C	0.67/1.43	0.39/0.81	0.47/1.09	0.36/0.80	0.54/1.19	0.49/1.06
5	✓	✓	10	1	C	0.64/1.28	0.39/0.78	0.42/0.92	0.34/0.74	0.52/1.13	0.46/0.97
6	✓	✓	2	1	C	0.71/1.45	0.37/0.75	0.43/0.93	0.40/0.97	0.54/1.21	0.49/1.06
7	✓	✓	10	2	C	**0.63/1.25**	**0.37/0.74**	**0.41/0.90**	**0.32/0.70**	**0.51**/1.10	**0.45/0.94**
8	✓	✓	10	3	C	0.64/1.27	0.38/0.75	0.42/0.91	0.32/0.71	0.51/**1.10**	0.45/0.95
9	✓	✓	10	1	P	0.71/1.42	0.39/0.87	0.47/1.05	0.35/0.78	0.53/1.16	0.49/1.06

Table 2. Ablation Study on SR-LSTM. **MG** denotes introducing the motion gate, **PA** denotes the pedestrian-wise attention layer. **NS** denotes the neighborhood size in meters, the value of 10 and 2 respectively give a neighborhood region of 20×20 and 4×4. **L** is the refinement iterations. **C/P** denotes that we use current or previous hidden states to perform the refinement. Variant 1,2 perform the simple states refinement without any feature selection (Eq.3).

message passing term for normalization. The quantitative results of different model variants are reported in Table 2.

Simple states refinement. Performing the simple states refinement (Eq.6) without any feature selection and considering the neighborhood size of 2 meters (Variant 1) outperforms V-LSTM by 6.4/6.8(%), as the human interaction is involved through the states refinement module. But the model with neighborhood size of 10 meters results in slight changes (1.4/-0.2(%)). The effect of neighborhood size is summarized in following paragraph.

Neighborhood size. We test two value of neighborhood size, 2 and 10, the effect of which are summarized: 1) Simple states refinement model with the equal treatment of all pedestrians within 10 meters (Variant 2), where useless features from far neighbors are still considered for message passing, causes performance deterioration of 5.6/7.5(%) relative to the same model with the neighborhood size of 2 meters (Variant 1). 2) With the proposed information selection mechanism, considering larger neighborhood size is generally better (Variant 5 vs 6). Therefore, our SR-LSTM could take advantage of useful information from farther neighbors.

Information selection. With neighborhood size fixed as 10 meter, only introducing the motion gate (Variant 3) or pedestrian-wise attention (Variant 4) is resultful, which respectively improves the performance by 7.8/12.2(%) and 6.7/8.3(%). Utilization of both these two components (Variant 5) achieves the improvement of 11.8/16.4/(%), which demonstrates the effectiveness of our information selection mechanism. When neighborhood size is set to 2 meters, adding motion gate and pedestrian-wise attention (Variant 6) still outperforms the simple refinement model (Variant 1) on average.

States refinement from current states. Utilization of the current states (Variant 5) outperforms the one using the previous states (Variant 9) by 6/8.3(%), which demonstrates the importance of latest features of neighbors.

Refinement iterations. Employing the second states re-

finement layer (Variant 7) performs consistently better than only refine the states once (Variant 5) by 2.8/3(%). While the third layer introduced could not bring further promotion. It may suggest that the choice of two refinement iterations is the appropriate for this task.

4.4. Comparison with Existing Works

We compare our model with several recent existing works: (1) Social-LSTM [1]: A cubic tensor is used in this approach to gather the social information. The recommended neighborhood size is 32 pixels in image space, we choose it as 2 and 10 meters respectively referred as S-LSTM_1 and S-LSTM_2. (2) SGAN [12]: A multimodal method to retrieve multiple possible future paths. (3) Sophie [35]: An improved multimodal method which introduces the attention on social relationship and physical acceptability.

The results are shown in Table 3. All of methods are under the same dataset setting and evaluation methodology. Note that SGAN and Sophie report the results that best match groundtruth in 20 samples, the other methods only produce one prediction; Sophie also requires the scene image.

V-LSTM vs V-LSTM*. V-LSTM models implemented by ourselves in Table 1 could not completely match the result of V-LSTM*, which is reported in [12]. This is possibly due to the deviation on hyper-parameters, data organization, or the teaching mode. In addition, we try our best to search for better data reprocessing which results in a considerable promotion.

SR-LSTM vs others. By the captured multimodality, S-GAN and Sophie improve significantly in comparison with V-LSTM*. But SGAN could not outperform V-LSTM* with only a single sample [12]. Our best model increases the performance relative to V-LSTM for 15.4/18.8(%), with only a single prediction.

S-LSTM_1 outperforms V-LSTM but still has higher prediction error than our approach, because it only takes advan-

Method	Notes	Performance (MAD/FAD)					
		ETH-univ	ETH-hotel	UCY-zara01	UCY-zara02	UCY-univ	AVG
V-LSTM*	-	1.09/2.41	0.86/1.91	0.41/0.88	0.52/1.11	0.61/1.31	0.7/1.52
SGAN*	20 samples	0.81/1.52	0.72/1.61	0.34/0.69	0.42/0.84	0.60/1.26	0.58/1.18
Sophie*	20 samples+scene	0.70/1.43	0.76/1.67	**0.30/0.63**	0.38/0.78	0.54/1.24	0.54/1.15
S-LSTM_1	NS=2, grid: 4×4	0.70/1.40	**0.37/0.73**	0.49/1.15	0.39/0.89	0.60/1.32	0.51/1.10
S-LSTM_2	NS=10 grid: 4×4	0.77/1.60	0.38/0.80	0.51/1.19	0.39/0.89	0.58/1.28	0.53/1.15
V-LSTM	-	0.83/1.77	0.41/0.80	0.49/1.15	0.37/0.85	0.56/1.22	0.53/1.16
SR-LSTM_1	ID 6 in Tab.2	0.64/1.28	0.39/0.78	0.42/0.92	0.34/0.74	0.52/1.13	0.46/0.97
SR-LSTM_2	ID 7 in Tab.2	**0.63/1.25**	0.37/0.74	0.41/0.90	**0.32/0.70**	**0.51/1.10**	**0.45/0.94**

Table 3. Comparison with several baselines models. **NS** denotes the neighborhood size in meters. The results of methods marked with * are directly obtained from [12, 35].

tage of previous hidden states of local neighbors. In addition, S-LSTM is not able to take advantage of the far neighbors according to the results of S-LSTM_2. Our SR-LSTM makes it possible to consider far neighbors and utilize their current states to refine each other.

4.5. Qualitative Results

Feature refinement from current states. Benefiting from our states refinement module, SR-LSTM is able to take advantage from the current neighboring states. Fig. 5(a) shows examples in which pedestrians' walking direction have suddenly changed before few time steps. V-LSTM (first column) does not consider the interaction and results in large error. S-LSTM (S-LSTM_1 in Table 3, second column) utilizes the previous neighboring LSTM states, but is still insensitive to these cases. Our SR-LSTM (SR-LSTM_2 in Table 3, third column) refines the current LSTM states through message passing, which can timely capture changes of the others' intention and make suitable adjustment.

Social behaviors. SR-LSTM can moderately explain implicit social behaviors. In Fig.5(b), we illustrate three cases, consistent group walking, collision avoidance and group avoidance. In V-LSTM, pedestrians are walking in their own. S-LSTM performs weaker to model pedestrian interactions and ignores the potential effect from far neighbors. Our SR-LSTM shows pretty ability to make appropriate prediction towards social interaction.

4.6. Social-aware Information Selection

Motion gate. When predicting the position of pedestrian i, motion gate acted on the hidden features of his/her neighbor j is calculated based on the pairwise features between pedestrian i and j (Eq.10). Fig.6 shows how motion gate selects the features, where each row is related to a certain dimension of hidden feature.

In Fig.6, the first column shows the trajectory patterns captured by hidden features started from origin and ended at the dots, which are extracted in similar way as Fig.2(a). The motion gate for a feature considers pairwise input tra-
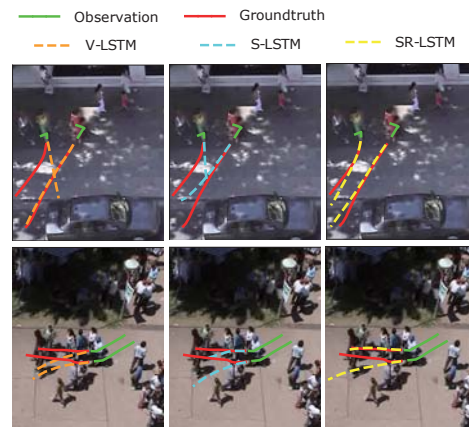
(a)
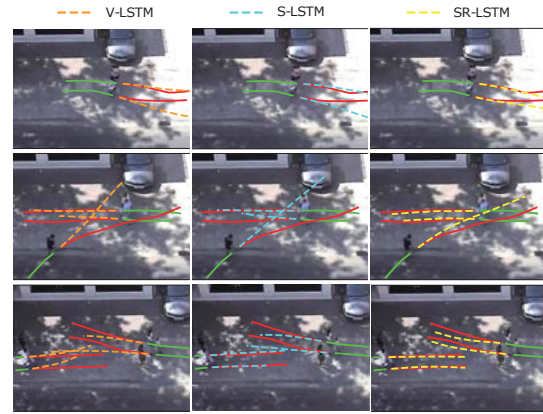
(b)

Figure 5. Illustration of the prediction trajectories. (a). In SR-LSTM, current states of pedestrians can timely refine each other, particularly in the case where pedestrians change their intentions. (b). SR-LSTM are able to implicitly explain for common social behaviors, which gives moderate future predictions and relatively low errors.

jectories with similar configurations. Some examples for high response of the gate are shown in the other columns of

Fig.6. In these pairwise trajectory samples, the red and blue ones are respectively the trajectories of pedestrian i and j, and the time step we calculate the motion gate are shown with dots (where the trajectory ends). These pairwise samples are extracted by searching from database with highest activation for the motion gate neuron. High response of gate means that the corresponding feature is selected.
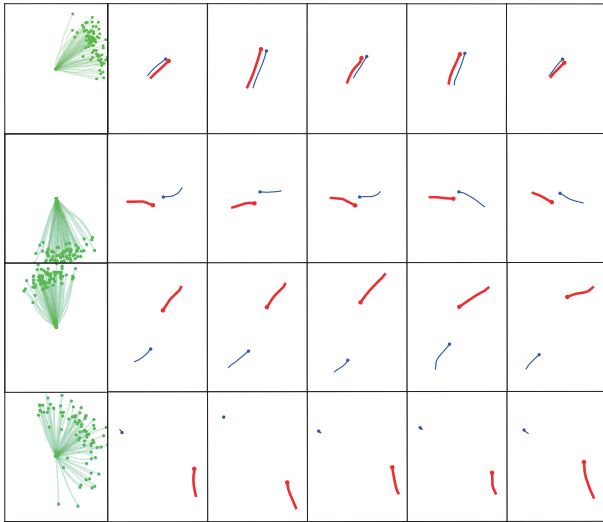

Figure 6. Selected feature patterns by motion gate. Each row is related to a hidden neuron (feature) of LSTM. Column 1: Activation trajectory pattern of the hidden feature. Column 2-6: Pairwise trajectory examples (end with solid dots) having high activation to the motion gate. Prediction for the pedestrian in red is mostly sensitive to the other's potential trajectories showed in first column, which are selected by our motion gate.

As shown in Fig.6, a gate for the same feature is responsible for roughly similar interaction conditions. When predicting the trajectory of pedestrian i (red), our motion gate attentively select features of pedestrian j (blue). These selected features shown in first column represent the potential trajectories that the pedestrian j might cause future interaction with the pedestrian i.

We explain effects of four gate elements in each row of Fig.6: 1) Row 1: The trajectory pairs are very close and are walking together. The selected hidden feature follows the walking direction. 2) Row 2: The trajectories are somewhat close but walking in opposite direction. The pedestrian i in red cares about whether the other will walk towards him/her. 3) Row 3: This case is similar to row 2. This gate element considers more distant neighbor walking in opposite direction. 4) Row 4: The neighbor in blue is static, the selected hidden feature shows that pedestrian i in red potentially pay attention on this stationary neighbor in case he is about to walk towards him/her.

Pedestrian-wise attention. We illustrate some examples of the pedestrian-wise attention expected by our SR-

LSTM in Fig.7. It shows that 1) dominant attention is paid to the close neighbors, while the others also take slight attention, 2) the attention given by the first refinement layer often largely focuses on the close neighbors, and the second refinement tends to strengthen the effect of farther neighbors with group behavior or may influence the pedestrian in longer time range.
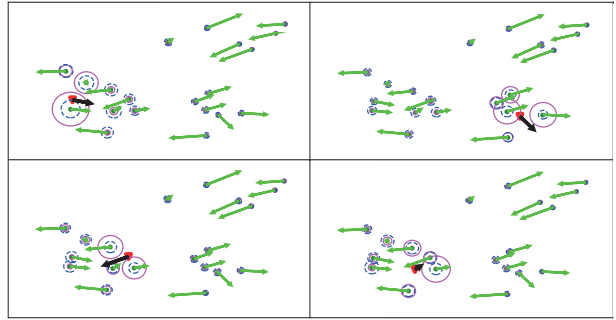

Figure 7. Illustration of the pedestrian-wise attention. Circle in magenta represents the attention in first round states refinement, the dashed circle represents for the attention in the second refinement. Larger circle corresponds to higher attention. Red triangle represents the target pedestrian for trajectory prediction, and green ones are his/her neighbors, the arrows on each of them represent their walking directions.

5. Conclusion

In this paper, we propose a states refinement module for LSTM network to address the the problem of joint trajectory prediction for pedestrians in the crowd. Our states refinement module treats LSTM as feature extractor, which adaptively refines current features of all pedestrians based on a message passing mechanism. In addition, we introduce a social-aware information selection mechanism consisting of an element-wise motion gate and a pedestrian-wise attention, to select useful features of each neighbor. The states refinement module with information selection outperforms the state-of-the-art approaches.

References

[1] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *CVPR*, pages 961–971, 2016. 1, 2, 5, 6

[2] S. Becker, R. Hug, W. Hübner, and M. Arens. Red: A simple but effective baseline predictor for the trajnet benchmark. In *ECCV Workshops*, pages 138–153, 2018. 2

[3] S. Bouzat and M. Kuperman. Game theory in models of pedestrian room evacuation. *Physical Review E*, 89(3):032806, 2014. 2

[4] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013. 3

[5] Y. Chrysanthou. https://graphics.cs.ucy.ac.cy/research/downloads/crowd-data. 2007. 5

[6] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014. 2

[7] T. Do, M. Haghani, and M. Sarvi. Group and single pedestrian behavior in crowd dynamics. *Transportation Research Record: Journal of the Transportation Research Board*, (2540):13–19, 2016. 1

[8] T. Fernando, S. Denman, S. Sridharan, and C. Fookes. Soft+ hardwired attention: An lstm framework for human trajectory prediction and abnormal event detection. *Neural Networks*, 2018. 3

[9] G. Ferrer, A. Garrell, and A. Sanfeliu. Robot companion: A social-force based approach with human awareness-navigation in crowded environments. In *IROS*, pages 1688–1694. IEEE, 2013. 2

[10] A. Gorrini, S. Bandini, and G. Vizzari. Empirical investigation on pedestrian crowd dynamics and grouping. In *Traffic and Granular Flow'13*, pages 83–91. Springer, 2015. 1

[11] A. Gorrini, G. Vizzari, and S. Bandini. Age and group-driven pedestrian behaviour: from observations to simulations. *Collective Dynamics*, 1:1–16, 2016. 1

[12] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *CVPR*, 2018. 1, 2, 3, 5, 6, 7

[13] I. Hasan, F. Setti, T. Tsesmelis, A. Del Bue, F. Galasso, and M. Cristani. Mx-lstm: mixing tracklets and vislets to jointly forecast trajectories and head poses. *arXiv preprint arXiv:1805.00652*, 2018. 1, 2

[14] D. Helbing, L. Buzna, A. Johansson, and T. Werner. Self-organized pedestrian crowd dynamics: Experiments, simulations, and design solutions. *Transportation science*, 39(1):1–24, 2005. 2

[15] D. Helbing and P. Molnar. Social force model for pedestrian dynamics. *Physical review E*, 51(5):4282, 1995. 2

[16] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 1, 2

[17] S. Hoogendoorn, W. Daamen, Y. Shu, and H. Ligteringen. Modeling human behavior in vessel maneuver simulation by optimal control and game theory. *Transportation research record*, 2326(1):45–53, 2013. 2

[18] S. Hoogendoorn and P. HL Bovy. Simulation of pedestrian flows by optimal control and differential games. *Optimal Control Applications and Methods*, 24(3):153–172, 2003. 2

[19] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei. Relation networks for object detection. In *CVPR*, volume 2, 2018. 3

[20] R. Hug, S. Becker, W. Htibner, and M. Arens. Particle-based pedestrian path prediction using lstm-mdl models. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2684–2691, 2018. 2

[21] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 3

[22] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. Torr, and M. Chandraker. Desire: Distant future prediction in dynamic scenes with interacting agents. In *CVPR*, pages 336–345, 2017. 2

[23] A. Lerner, Y. Chrysanthou, and D. Lischinski. Crowds by example. In *Computer Graphics Forum*, volume 26, pages 655–664. Wiley Online Library, 2007. 4

[24] J. Li, Y. Wei, X. Liang, J. Dong, T. Xu, J. Feng, and S. Yan. Attentive contexts for object detection. *IEEE Transactions on Multimedia*, 19(5):944–954, 2017. 2

[25] Y. Li, W. Ouyang, X. Wang, and X. Tang. Vip-cnn: Visual phrase guided convolutional neural network. In *CVPR*, pages 7244–7253, 2017. 3

[26] Y. Li, W. Ouyang, B. Zhou, K. Wang, and X. Wang. Scene graph generation from objects, phrases and region captions. In *ICCV*, pages 1270–1279, 2017. 3

[27] X. Liang, L. Lin, X. Shen, J. Feng, S. Yan, and E. P. Xing. Interpretable structure-evolving lstm. In *CVPR*, pages 2175–2184, 2017. 3

[28] X. Liang, X. Shen, J. Feng, L. Lin, and S. Yan. Semantic object parsing with graph lstm. In *ECCV*, pages 125–143. Springer, 2016. 3

[29] W.-C. Ma, D.-A. Huang, N. Lee, and K. M. Kitani. Forecasting interactive dynamics of pedestrians with fictitious play. In *CVPR*, pages 4636–4644. IEEE, 2017. 2

[30] R. McCool, J. M. Usher, L. Strawderman, D. Carruth, C. Bethel, and D. May. Simulating group formations that arise in pedestrian traffic. In *IIE Annual Conference. Proceedings*, pages 133–138. Institute of Industrial and Systems Engineers (IISE), 2017. 1

[31] R. Mehran, A. Oyama, and M. Shah. Abnormal crowd behavior detection using social force model. In *CVPR*, pages 935–942. IEEE, 2009. 2

[32] M. Moussaïd, N. Perozo, S. Garnier, D. Helbing, and G. Theraulaz. The walking behaviour of pedestrian social groups and its impact on crowd dynamics. *PloS one*, 5(4):e10047, 2010. 2

[33] S. Pellegrini, A. Ess, K. Schindler, and L. J. Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *ICCV*, volume 9, pages 261–268, 2009. 4, 5

[34] P. Ratsamee, Y. Mae, K. Ohara, T. Takubo, and T. Arai. Human–robot collision avoidance using a modified social force model with body pose and face orientation. *International Journal of Humanoid Robotics*, 10(01):1350008, 2013. 2

[35] A. Sadeghian, V. Kosaraju, A. Sadeghian, N. Hirose, and S. Savarese. Sophie: An attentive gan for predicting paths compliant to social and physical constraints. *arXiv preprint arXiv:1806.01482*, 2018. 1, 2, 3, 6, 7

[36] L. Shi, Y. Zhang, J. Cheng, and H. Lu. Adaptive spectral graph convolutional networks for skeleton-based action recognition. *arXiv preprint arXiv:1805.07694*, 2018. 3

[37] H. Su, Y. Dong, J. Zhu, H. Ling, and B. Zhang. Crowd scene understanding with coherent recurrent neural networks. In *IJCAI*, volume 1, page 2, 2016. 2, 3

[38] H. Su, J. Zhu, Y. Dong, and B. Zhang. Forecast the plausible paths in crowd scenes. In *IJCAI*, pages 2772–2778, 2017. 1, 2, 3

[39] D. Varshneya and G. Srinivasaraghavan. Human trajectory prediction using spatially aware deep attention models. *arXiv preprint arXiv:1705.09436*, 2017. 1, 2

[40] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017. 2

[41] A. Vemula, K. Muelling, and J. Oh. Social attention: Modeling attention in human crowds. In *ICRA*, pages 1–7. IEEE, 2018. 1, 2, 3

[42] X. Wang and A. Gupta. Videos as space-time region graphs. *arXiv preprint arXiv:1806.01810*, 2018. 3

[43] Y. Wang, S. Wang, J. Tang, N. O'Hare, Y. Chang, and B. Li. Hierarchical attention network for action recognition in videos. *arXiv preprint arXiv:1607.06416*, 2016. 2

[44] D. Xu, Y. Zhu, C. B. Choy, and L. Fei-Fei. Scene graph generation by iterative message passing. In *CVPR*, volume 2, 2017. 3

[45] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, pages 2048–2057, 2015. 2, 3

[46] Y. Xu, Z. Piao, and S. Gao. Encoding crowd interaction with deep neural network for pedestrian trajectory prediction. In *CVPR*, pages 5275–5284, 2018. 1, 2

[47] K. Yamaguchi, A. C. Berg, L. E. Ortiz, and T. L. Berg. Who are you with and where are you going? In *CVPR*, pages 1345–1352, 2011. 2

[48] S. Yan, Y. Xiong, and D. Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. *arXiv preprint arXiv:1801.07455*, 2018. 3

[49] J. Yang, J. Lu, S. Lee, D. Batra, and D. Parikh. Graph r-cnn for scene graph generation. In *ECCV*, pages 690–706. Springer, 2018. 3

[50] S. Yi, H. Li, and X. Wang. Understanding pedestrian behaviors from stationary crowd groups. In *CVPR*, pages 3488–3496, 2015. 1, 2

[51] S. Yi and X. Wang. Profiling stationary crowd groups. In *ICME*, pages 1–6. IEEE, 2014. 1, 2

[52] W. Yu and D. Helbing. Game theoretical interactions of moving agents. In *Simulating Complex Systems by Cellular Automata*, pages 219–239. Springer, 2010. 2

[53] Y. Yuan, X. Liang, X. Wang, D.-Y. Yeung, and A. Gupta. Temporal dynamic graph lstm for action-driven video object detection. In *ICCV*, pages 1819–1828, 2017. 3

[54] Y. Zhang, L. Qin, H. Yao, and Q. Huang. Abnormal crowd behavior detection based on social attribute-aware force model. In *ICIP*, pages 2689–2692. IEEE, 2012. 2

[55] X. Zheng and Y. Cheng. Conflict game in evacuation process: A study combining cellular automata model. *Physica A: Statistical Mechanics and its Applications*, 390(6):1042–1050, 2011. 2