Pedestrian Trajectory Prediction Using a Social Pyramid

Hao Xue(✉), Du Q. Huynh, and Mark Reynolds

The University of Western Australia, Perth, Australia
hao.xue@research.uwa.edu.au, {du.huynh,mark.reynolds}@uwa.edu.au

Abstract. Understanding and forecasting human movement paths are vital for a wide range of real world applications. It is not an easy task to generate plausible future paths as the scenes and human movement patterns are often very complex. In this paper, we propose a social pyramid based prediction method (SPP), which includes two encoders to capture motion and social information. Specifically, we design a social pyramid map structure for the Social encoder, which can differentiate the influence of other pedestrians in nearby areas or remote areas based on their spatial locations. For the Motion encoder, a mixing attention mechanism is proposed to combine the location coordinates and velocity vectors. The two encoded features are then merged and passed to the decoder which generates future paths of pedestrians. Our extensive experimental results demonstrate competitive prediction performance from our method compared to state-of-art methods.

Keywords: Trajectory prediction · LSTM · Social pyramid

1 Introduction

Research on predicting trajectories of pedestrians has gained much attention in the past few years due to its relevance to a large number of applications such as traffic analysis [6,25], crowd management [42,44], pedestrian tracking [8], autonomous vehicles [3], and anomaly detection [24]. The prediction of pedestrians' moving paths depends on two crucial factors. First is *Motion information*: the movement of a target person (or person of interest, POI) relies on his/her own motion information. This provides the main clue for future path prediction. Second is *Social information*: a pedestrian should not be treated in isolation during the prediction process. The route of one pedestrian is often under the influence of other people in the scene. This is often referred to as *social influence* or *social interaction* in the literature.

Pedestrian trajectory prediction is not a new problem. Research work in this field dates back to the 90s. Classical prediction models [13,16,26,41] reported in the literature rely heavily on hand-crafted features and they have been shown to give poorer performance compared to more modern techniques. With a large amount of video data collected in many public areas today, we see a growing

A. C. Nayak and A. Sharma (Eds.): PRICAI 2019, LNAI 11671, pp. 439–453, 2019.
https://doi.org/10.1007/978-3-030-29911-8_34

Fig. 1. In a crowded scene, the immediate neighbourhood region (blue) and a remote area (green) have different degrees of *social influence* to the POI (red dot). The crowd in the blue region would affect the POI's walking pace; the crowd in the green area would affect the POI's future path if his/her destination is the exit at the upper left corner. (Color figure online)

number of data-driven deep learning methods [1,11,12,18,19,28–30,34,38] being applied to pedestrian trajectory forecasting. Social influence is known to play an important role in pedestrians' walking paths in a scene. Indeed, quite a few papers on incorporating social influence into trajectory prediction framework have already been reported in the literature. For example, social pooling layers have been used in the Social LSTM [1] network to model the social relationship between nearby neighbours; different shapes of neighbourhood regions together with scene information have been used to capture pedestrian movement patterns [40]; a social pooling module, which expands on the social pooling idea, have been used in Social GAN [11]. The pooling mechanism in this method considers all the pedestrians in the scene rather than only the surrounding neighbours.

To illustrate the complexity of social influence in a crowded scene, consider the image of the busy Central Station shown in Fig. 1. With respect to the POI (red dot), the immediate neighbourhood (blue rectangle) is likely to have more influence on his/her walking pace: generally pedestrians prefer to keep some distance from strangers in public areas. However, the more remote areas (e.g., the region marked in green) should not be ignored. When pedestrians move in a scene, remote areas serve as a guidance for their future path planning. Suppose that the POI intends to go to the exit on the top left corner of the image. He/she would more likely detour slightly to avoid the crowd in the green region rather than taking a straight-line path. In this paper, we propose to handle social influence by dividing the scene into grids and looking at the pedestrians in each grid in turn, *i.e.*, our method does not focus on just the immediate neighbourhood of the POI. However, unlike the existing methods above which model social influence using either a small neighbourhood region around the POI (*e.g.*, [1,35,40]) or all the pedestrians in the scene (*e.g.*, [11,21]), our method

does not consider the exact location coordinates of pedestrians in the remote areas. Our method can be considered as between the two categories above.

To handle the motion information, we design an attention mechanism that merges the location and velocity information in our network. This is different from existing techniques which use only the location coordinates [1,29,30,46] or only the displacement vectors [21,38,43].

We name our model *SPP*, which is short for *Social Pyramid based Prediction*. The research contributions of this work are summarized below: (i) Our SPP network embodies information from both the motion clue of pedestrians' own trajectories and the social neighbourhood. We evaluate our method on publicly available datasets with different experimental settings and compare its performance with state-of-art methods. We also justify the effectiveness of our method by comparing it with its three variants. (ii) The Motion encoder of our network includes information from location and the velocity coordinates. We also design a mixing attention mechanism to merge these two parts for each trajectory. (iii) We propose to use a novel social pyramid structure to handle the social relationship from all the other pedestrians at different levels, starting from the whole scene and gradually zooming into each person of interest.

The paper is organized as follows. Related work about different trajectory prediction methods are given in Sect. 2. Section 3 presents the proposed SPP. The datasets used for evaluation, details of our experiments, quantitative and qualitative results are discussed in Sect. 4. Finally, the last section concludes the paper.

2 Related Work

Classical Models

Traditionally, hand crafted features were widely used for human motion modelling and trajectory prediction. The Social Force model [13] (SFM), a classical and pioneer model designed to describe pedestrian behaviour, encompasses two interactive social forces: the attractive force towards the pedestrians' destinations and the repulsive force for collision avoidance. This method has been recently revisited by Pellegrini *et al.* [26]. The authors design a trajectory prediction model, named as Linear Trajectory Avoidance (LTA), which takes into account both simple scene information in the form of destinations or desired direction and interactions between different targets. This LTA model has also been used for tracking people in their work. Since then, further research built on top of the SFM includes: Yamaguchi *et al.* [41] extend the Social Force model by incorporating more factors such as damping and social interactions; Xie *et al.* [37] add scene context information into the cost function. The main shortcoming of these methods is that their performance highly depends on energy cost functions manually designed based on relative distances or specific rules.

Another research branch is trajectory prediction based on Gaussian Processes. Kim *et al.* [15] model trajectories as a continuous dense flow field and use Gaussian Process regression to classify trajectories and detect anomaly. For

pedestrian trajectory prediction, Wang *et al.* [36] propose to use Gaussian Process Dynamical models to capture the motion dynamics of trajectories. Vemula *et al.* [35] design an interactive Gaussian Process model to describe the cooperative behaviour in dense human crowds, targeting at collision avoidance problems in robotics.

CNN Based Models
Convolutional Neural Networks are commonly used in classification or recognition problems that are associated with images. However, CNN models have been used successfully in prediction problems. The behaviour-CNN of Yi *et al.* [43] encodes the pedestrians' walking paths as displacement volumes, which are then used to predict future trajectories. Nikhil *et al.* [25] design a CNN that has highly parallelizable convolutional layers to deal with motion dependencies in the trajectory data. For vehicle trajectories, a convolutional social pooling structure has been used to encode the past motion of neighbouring vehicles [6] and a multiple layer CNN has been adopted to combine multi-scale trajectory patterns for trajectory prediction [23].

LSTM Based Models
Recurrent models such as the basic Recurrent Neural Network (RNN), Long Short Term Memory (LSTM) [14], and Gated Recurrent Unit (GRU) [5] have been designed and successfully applied to various sequence data tasks including speech recognition [10,27], language translation [32], action recognition [22,45] and image captioning [4]. In the last few years, several methods have also been proposed to use LSTMs to handle trajectories represented as sequences of 2D coordinates. Motivated by the sequence generation model from [9], Alahi *et al.* [1] propose the Social LSTM model which includes a social pooling layer to model the neighbourhood and to avoid collisions between pedestrians.

Since the proposal of Social LSTM [1], a tremendous number of LSTM based trajectory prediction methods have been published [31,38,39,46]. From the Social Force model [13] to Social LSTM [1] and Social GAN [11], the social influence aspect in trajectory prediction has particularly drawn much interest. Typical work includes: Gupta *et al.*'s paper [11] on using a pooling module to expand the social neighbourhood modelling to the entire scene; and Li's [20,21] handling of social information using a convolutional LSTM network to incorporate social information.

Apart from the work above, we also see attention mechanisms being incorporated into trajectory forecasting, *e.g.*, [7,34], scene context being encoded into prediction networks, *e.g.*, [2,18,40], and head pose information being used, *e.g.*, the MX-LSTM architecture of [12].

3 Proposed Method

3.1 Problem Definition and System Overview

The trajectory of pedestrian i, where $i \in [1, N]$ assuming that there are N pedestrians in the scene, is represented as a sequence of two dimensional coordinates

(x_t^i, y_t^i). The trajectory prediction problem can be described as one where we observe the coordinates from $t = 1$ to $t = T_{\text{obs}}$ and make predictions for the time span $t = T_{\text{obs}}+1, \cdots, T_{\text{obs}}+T_{\text{pred}}$. Here, T_{obs} and T_{pred} are the observation length and the prediction length, respectively.

Our proposed network is illustrated in Fig. 2. To handle both the motion and social dependencies in the trajectory prediction process, we propose to use two encoders: one for the motion information and one for the social information. The predicted trajectories are generated through the decoder which takes the output from the two encoders as input. In the subsections below, we will describe the details of these three key components of our network separately.

3.2 The Motion Encoder

This encoder focuses on the POI's own history path, which is the dominating information for generating his/her future path. Unlike other methods, both the location and velocity terms of each trajectory are taken as input in our proposed network. The velocity (u_t^i, v_t^i) is obtained from the finite differences of (x_t^i, y_t^i) with respect to time t. As the velocity term depicts the instantaneous moving direction and stride of the pedestrian, it is independent of the POI's absolute location on the ground. The velocity term therefore captures important walking pace of the pedestrian regardless of which part of the scene he/she is located. Instead of passing the coordinates and displacements directly to the LSTM encoder, a mixing attention layer is used to firstly merge the terms (x_t, y_t) and (u_t, v_t) so that our model has the attentiveness capability. The idea of adding a mixing attention layer is based on the observation that the location and velocity terms do not usually contribute equally to the trajectory prediction process. To simplify the explanation, the subscript i is dropped from hereon.

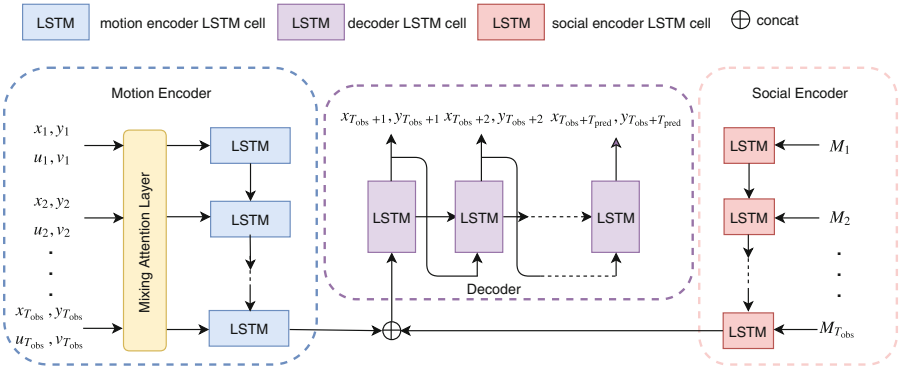

Fig. 2. The proposed SPP framework. Location (x_t, y_t) and velocity (u_t, v_t) are jointly considered through the mixing attention layer in the Motion encoder. The social pyramid tensors $M_t, \forall t$, are the input of the Social encoder. The encoded motion and social features are concatenated as input to the decoder.

Using the renowned scaled dot-product attention mechanism proposed by Vaswani *et al.* [33], an attention function can be considered as one that maps a query vector and a set of key-value pairs to an output. At each time step, our (x_t, y_t) and (u_t, v_t) terms can thus be embedded into the query vectors $\mathbf{q}_t^l$ and $\mathbf{q}_t^v$ and key vectors $\mathbf{k}_t^l$ and $\mathbf{k}_t^v$. These embedding query and key vectors are r-dimensional, where r is a chosen integer suitable for the problem.

The query vectors $\mathbf{q}_t^l$ and $\mathbf{q}_t^v$ are stacked row-wise to form a matrix $\mathbf{Q}_t \in \mathbb{R}^{2\times r}$. Similarly, $\mathbf{k}_t^l$ and $\mathbf{k}_t^v$ are stacked row-wise to form $\mathbf{K}_t \in \mathbb{R}^{2\times r}$. The attention matrix $\mathbf{A}_t \in \mathbb{R}^{2\times 2}$ is then calculated as:

$$\mathbf{A}_t = \text{softmax}\left(\frac{\mathbf{Q}_t \mathbf{K}_t^\top}{\sqrt{r}}\right), \tag{1}$$

where $\sqrt{r}$ is used as a scaling factor and the softmax function is applied to the matrix in a row-wise fashion.

We use the diagonal entries of $\mathbf{A}_t$ as the weights for the location and velocity terms. That is, we let $\alpha_{t,1} = \mathbf{A}_{t,1,1}$ and $\alpha_{t,2} = \mathbf{A}_{t,2,2}$. Note that it is not necessary to normalize so that $\alpha_{t,1} + \alpha_{t,2} = 1$ as the scale would be absorbed by the downstream LSTM layer. The combined weighted output $(\hat{x}_t, \hat{y}_t)$ after the mixing attention is computed using Eq. (2) and the encoded hidden state, $\mathbf{h}_t^{\text{m}}$, of the Motion encoder part is defined using Eq. (3):

$$(\hat{x}_t, \hat{y}_t) = \alpha_{t,1}\,(x_t, y_t) + \alpha_{t,2}\,(u_t, v_t) \tag{2}$$

$$\mathbf{h}_{t+1}^{\text{m}} = \text{LSTM}_{\text{m}}\left(\mathbf{h}_t^{\text{m}}, (\hat{x}_t, \hat{y}_t)\,; \mathbf{W}_{\text{m}}\right), \tag{3}$$

where $\mathbf{W}_{\text{m}}$ is the weight matrix of LSTM_{m}, the LSTM layer of the Motion encoder.

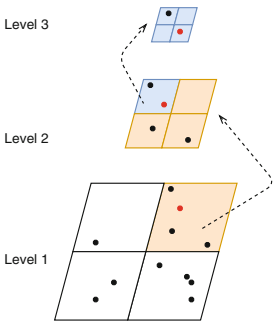

Fig. 3. An illustration of the proposed Social Pyramid Map construction process. The red dot represents the POI and black dots are other pedestrians in the scene at the same time step. In this example, the grid size n is set to 2 and the pyramid map has 3 levels. The ROI of level 1 is the whole scene. The peach colour grid at level 1 becomes the ROI at level 2. The light blue grid of level 2 becomes the ROI at level 3. (Color figure online)

3.3 The Social Encoder

To incorporate the social information, current existing prediction methods in the literature consider either the neighbours of the POI [1,35,40] or the whole scene [11,21]. However, people in remote areas do not have the same degree of influence as pedestrians in the immediate neighbourhood around a POI. Thus, we use a pyramid structure (Fig. 3) of L levels and grid size n to capture the social information around each POI at different scales. At level 1, the region of interest (ROI) containing the POI is just the entire scene. As one moves up to the next level of the pyramid structure, the size of ROI is reduced by the factor given by n. This structure can be seen as a spotlight following the POI and gradually zooming into his/her neighbourhood level by level.

With the setting of grid size n and map level L, for each pedestrian at time step t, a social pyramid tensor $M_t \in \mathbb{R}^{2\times n^2\times L}$ is built using Algorithm 1. For simplification, we drop superscript i but note that the social pyramid tensor is computed for each pedestrian. In the algorithm, $\mathcal{P}_t = \{(x_t^j, y_t^j)\}_{j=1}^{J_t}$ denotes a set of pedestrians' location coordinates, where J_t is the total number of pedestrians inside the boundary of the ROI at the current level at time t. At the beginning, $\mathcal{P}_t$ is initialized to contains all the pedestrians in the whole scene. The 3D entities $g_t^l \in \mathbb{R}^{2\times n\times n}$, for $l = 1, \cdots, L$, are the social map tensors for the construction of M_t. At the grid indexed by $[a, b]$, the entity $g_t^l[a, b]$ is a $\mathbb{R}^2$ vector holding information at that region at level l of the scene. We use $\mathcal{I}_{ab}^l(x, y)$ to denote the

Algorithm 1. Building the Social Pyramid Map for a POI at location (x_t, y_t) at time t

Input: n: grid size; L: maximum map level; (x_t, y_t): location of POI; ROI: region of interest containing the POI; $\mathcal{P}_t$: $\{(x_t^j, y_t^j)\}_{j=1}^{J_t}$, set of pedestrians' coordinates in ROI
Output: Social pyramid map tensor M_t at time step t

for $1 \le l \le L$ **do**
  divide ROI into $n \times n$ grids
  for each grid $[a, b]$ at level l in ROI **do**
    if $\mathcal{I}_{ab}^l(x_t, y_t) = 1$ **then**
      $g_t^l[a, b] \leftarrow (x_t, y_t)$
      $\mathcal{P}_t \leftarrow \left\{(x_t^j, y_t^j) \in \mathcal{P}_t, \forall j \,|\, \mathcal{I}_{ab}^l\left(x_t^j, y_t^j\right) = 1\right\}$
      update ROI using grid $[a, b]$
    else
      if $\sum_{j=1}^{J} \mathcal{I}_{ab}^l\left(x_t^j, y_t^j\right) = 0$ **then**
        $\bar{x}_{a,b}^l \leftarrow 0$, $\bar{y}_{a,b}^l \leftarrow 0$
      else
        $\bar{x}_{a,b}^l \leftarrow \frac{\sum_{j=1}^{J_t} \mathcal{I}_{ab}^l\left(x_t^j, y_t^j\right) x_t^j}{\sum_{j=1}^{J_t} \mathcal{I}_{ab}^l\left(x_t^j, y_t^j\right)}$
        $\bar{y}_{a,b}^l \leftarrow \frac{\sum_{j=1}^{J_t} \mathcal{I}_{ab}^l\left(x_t^j, y_t^j\right) y_t^j}{\sum_{j=1}^{J_t} \mathcal{I}_{ab}^l\left(x_t^j, y_t^j\right)}$
      $g_t^l[a, b] \leftarrow \left(\bar{x}_{a,b}^l, \bar{y}_{a,b}^l\right)$
$M_t \leftarrow g_t^1 \oplus g_t^2 \oplus \cdots \oplus g_t^L$
return M_t

indicator function which returns 1 if the coordinates (x, y) fall inside the grid $[a, b]$ at the l^{th} level of the social map tensor, and returns 0 otherwise.

The way M_t is constructed is based on the observation that, with respect to a POI, the exact coordinates of each pedestrian in those remote areas are not important. To the POI, each remote area at different parts of the scene is perceived to have a small group of pedestrians scattering around. Only until the POI moves into one of these areas, more detailed information about the pedestrians in that area would become relevant. As a result, for those grids $g_t^l[a, b]$ that do not contain the POI, the algorithm computes only the mean coordinates $(\bar{x}_{a,b}^l, \bar{y}_{a,b}^l)$ (lines 12–13 of Algorithm 1). At each level, the grid containing the POI becomes the ROI for the next level. The social pyramid map M_t is finally formed by the concatenation of the 3D tensors $g_t^l, \forall l$.

The grid size n and the total number of levels L are two related parameters that define the size of M_t. Let H and W be the height and width of the scene in pixels. As it is impossible for two persons to fall onto the same pixel, once n is determined, it is necessary that $n^L / \min(H, W) \geq 1$. This means that

$$L \geq \lceil \log \min(H, W) / \log n \rceil \,, \tag{4}$$

where $\lceil . \rceil$ denotes rounding up to the nearest integer. Since the size of the ROI needs to shrink by a factor of n when moving up one level of the social pyramid tensor M_t, the only constraint on n is $n \geq 2$. For large n, fewer levels (smaller L) would be needed to zoom into the POI. In the case where n becomes so large that each grid contains at most 1 pedestrian only, M_t would collapse to a degenerate 1-level tensor. This would be undesirable as it would not be able to distinguish remote areas versus immediate neighbourhood any more. So, for the social pyramid tensor to be useful, n should not be too large. In Sect. 4.3, we explore various values of n and the associated prediction errors.

The constructed social pyramid tensor M_t is finally passed to the Social encoder LSTM_s for encoding the hidden state $\mathbf{h}_t^\text{s}$ as:

$$\mathbf{h}_{t+1}^\text{s} = \text{LSTM}_\text{s}\left(\mathbf{h}_t^\text{s}, M_t; \mathbf{W}_\text{s}\right), \tag{5}$$

where $\mathbf{W}_\text{s}$ is the weight matrix of LSTM_s.

3.4 Generating Predictions

We apply a straightforward encoder-decoder architecture for the trajectory prediction process. The input to the decoder LSTM_dec is the concatenated encoded hidden states $(\mathbf{h}_t^\text{m} \oplus \mathbf{h}_t^\text{s})$. The predicted trajectory coordinates are calculated using Eq. (7) below:

$$\mathbf{h}_{t+1}^\text{dec} = \text{LSTM}_\text{dec}\left(\mathbf{h}_t^\text{dec}, \mathbf{h}_t^\text{m} \oplus \mathbf{h}_\text{s}^s; \mathbf{W}_\text{dec}\right), \tag{6}$$

$$\left(x_t^i, y_t^i\right) = \mathbf{W}_o \mathbf{h}_t^\text{dec} + \mathbf{b}_o, \tag{7}$$

where $\mathbf{W}_\text{dec}$ is the weight matrix of the LSTM decoder, and $\mathbf{W}_o$ and $\mathbf{b}_o$ are the weight matrix and bias term of the output layer.

Our proposed network is implemented using the Pytorch framework in Python. The dimension of each LSTM layer is set to 128 and the dimension r of the query and key vectors for the mixing attention layer in the Motion encoder is 64. The mean squared error (MSE) loss function is used to train the network and the Adam optimizer [17] is used for optimization. The initial learning rate is set to 0.001, which is decreased by half at every 1000 epochs.

4 Experiments

4.1 Datasets, Baselines and Evaluation Metrics

The Central Station dataset [44] contains over 10,000 trajectories represented in pixel coordinates. The resolution of the videos is 720 (width) × 480 (height) pixels. For the sake of fair comparison, we follow the same setting as in [46] and set $T_{\text{obs}} = 9$ and $T_{\text{pred}} = 8$. Short trajectories are firstly filtered out and then the rest trajectories are split into the training set (80%) and the test set (20%). Our SPP and its variants were trained on the training set only. We report the prediction results on the test set for all experiments. We also preprocess the dataset by normalizing the coordinates of each trajectory so that all the coordinates in the dataset are within $[0, 1]$.

We compare the prediction performance of our SPP network with the following baselines and state-of-art methods:

- *Constant Velocity*: A linear prediction method that assumes each pedestrian keeping the same velocity for the whole journey.
- *Social Force Model (SFM)* [13]: Prediction based on the pioneer Social Force model.
- *Linear Trajectory Avoidance (LTA)* [26]: Prediction based on energy minimization to avoid collision.
- *Behaviour-CNN* [43]: A CNN based method for pedestrian trajectory prediction.
- *Vanilla LSTM:* This is the basic vanilla LSTM using only the location information as input.
- *SA-GAIL* [46]: Social-Aware Generative Adversarial Imitation Learning, a generative adversarial network designed for trajectory prediction.

Our proposed SPP method has three variants for the ablation study:

- ***SPP-social-only*:** The Motion encoder is removed in this simplified version of SPP. Only the Social encoder is connected to the decoder for prediction.
- ***SPP-motion-only*:** This variant of SPP does not use any social pyramid tensors and does not include the Social encoder in the network.
- ***LV-vanilla*:** This variant is a simplified version of SPP-motion-only, with the mixing attention layer removed.

We adopt two common metrics used in evaluating trajectory prediction: Average Displacement Error (ADE) and Final Displacement Error (FDE). Where appropriate, the normalized ADE (normADE), which scales the ADE with respect to the size of the image, is also used.

Table 1. Prediction errors (in pixels) on the Central Station dataset.

Method	normADE	ADE	FDE
Constant velocity[a]	5.86%	-	-
SFM[a] [13]	4.45%	-	-
LTA[a] [26]	4.35%	-	-
Behaviour CNN[a] [43]	2.52%	-	-
LSTM[a]	2.39%	14.57	27.78
SA-GAIL[a] [46]	1.98%	11.98	23.05
SPP (ours)	1.67%	10.05	18.48
SPP-social-only (ours)	1.76%	10.65	20.64
SPP-motion-only (ours)	1.89%	11.46	22.54
LV-vanilla (ours)	2.31%	14.27	24.74

[a] Results taken from [46].

4.2 Quantitative Results

The results on the Central Station dataset are shown in Table 1. We follow the same experimental settings (*e.g.*, same T_{obs} and T_{pred} values) and take some of their reported results for comparison. As the most straightforward method, using constant velocity for trajectory prediction gives the worst result. Small improvements are evident when more factors, such as collision avoidance in the LTA or the attractive/repulsive forces in the SFM, are included in the prediction model. Compared to the above classical methods that use manually designed energy functions, there is a large gain in prediction accuracy from data-driven deep learning methods. All LSTM based methods, including the vanilla one, have better accuracy than the Behaviour CNN method. This makes perfect sense as the LSTM architecture is specifically designed for analyzing sequence data. The results show that our proposed SPP algorithm outperforms other methods on all the three metrics.

4.3 Ablation Study

The numerical results of our ablation study on the Central Station dataset are shown on the last four rows of Table 1. The better predication result of SPP over SPP-social-only and SPP-motion-only demonstrates the importance of having the Motion and Social encoders working together. Compared to the Motion encoder part, the Social encoder and the social pyramid tensors appear to have more contributions to the improvement of prediction. Our variant LV-vanilla only performs slightly better than Vanilla LSTM. The much poorer performance of LV-vanilla versus that of SPP-motion-only confirms the significance of the mixing attention layer in the Motion encoder.

Table 2. Prediction errors (in pixels) of using different grid sizes.

Method	SPP			SPP-social-only		
n	3	5	7	3	5	7
normADE	1.72%	1.67%	1.65%	1.82%	1.76%	1.74%
ADE	10.45	10.05	9.97	10.94	10.65	10.48
FDE	20.12	18.48	19.24	21.19	20.64	20.43

Different Grid Sizes

The grid size n is an important parameter in SPP as it determines the size of the social pyramid tensor. Table 2 shows the performance of SPP and SPP-social-only on the Central Station dataset for $n = 3$, 5, and 7. The value of L is computed accordingly using Eq. (4) (by replacing the $\geq$ sign by $=$). With larger n, the prediction errors are smaller for both methods. The results are not surprising as a larger n value gives finer grids and reveals more detailed information of the neighbourhood. However, it also results in more entries in the social pyramid tensor M_t and would require longer time to train and predict. Table 2 shows that there is a larger gain in performance from $n = 3$ to $n = 5$ compared to that from $n = 5$ to $n = 7$. So we choose $n = 5$ and set L to 4 (the smallest value computed from Eq. (4)) for a trade-off between speed and accuracy. We use these as the default values for n and L in all the experiments reported in this paper.

Different Prediction Lengths

Table 3 shows how our proposed method performs on predicting trajectories of different lengths. We train two SPP models using the $T_{\text{obs}} = 9$ setting in the training phase: **SPP-p8**, trained with $T_{\text{pred}} = 8$; and **SPP-p16**, trained with $T_{\text{pred}} = 16$. In the testing phase, these two models can be used to predict trajectories of any lengths. In our experiments, we use them to generate trajectories of 8, 12 and 16 frames long. Table 3 shows that SPP-p8 performs slightly better for 8-frame prediction. However, SPP-p16 is clearly the winner in predicting longer trajectories.

Table 3. ADE/FDE (in pixels) on different prediction lengths. SPP-p8 is trained with $T_{\text{pred}} = 8$; SPP-p16 is trained with $T_{\text{pred}} = 16$.

Model	Prediction length		
	8	12	16
SPP-p8	10.05/18.48	16.43/34.47	23.25/49.36
SPP-p16	10.27/19.91	15.52/31.47	21.69/45.62

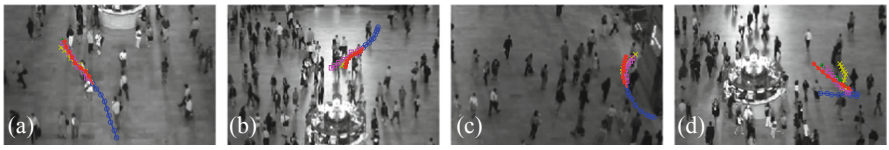

Fig. 4. Comparison of predicted trajectories generated by SPP (pink), SPP-motion-only (red) and SPP-social-only (yellow) on the Central Station dataset. Input observed trajectories are shown in blue; ground truth trajectories are in green. (Color figure online)

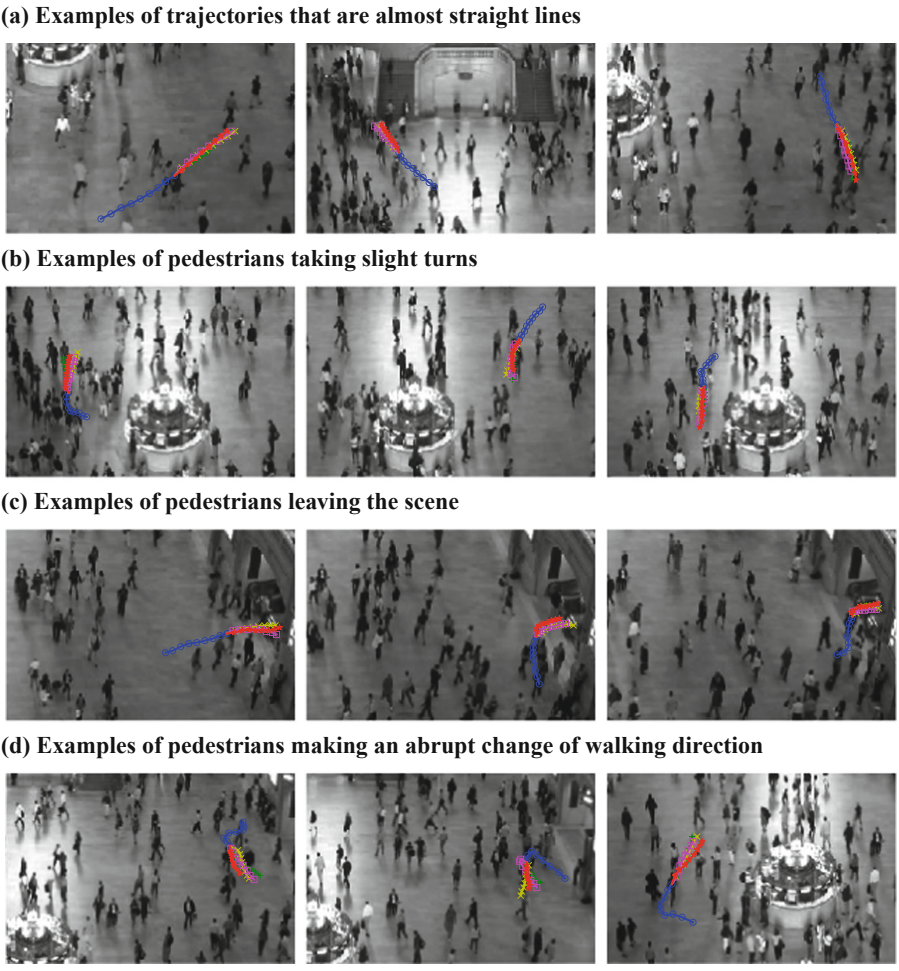

Fig. 5. Comparison of predicted trajectories generated by SPP on the Central Station dataset for grid size $n = 3$ (yellow), $n = 5$ (red), and $n = 7$ (pink). Input observed trajectories are shown in blue; ground truth trajectories are in green. (Color figure online)

4.4 Qualitative Results

We illustrate in Fig. 4 some qualitative prediction results generated by SPP (pink curves), SPP-social-only (red), and SPP-motion-only (yellow) for the 8-frame prediction length. The observed and ground truth trajectories are shown in blue and green, respectively. In the first three relative simple cases (parts (a)–(c)), the trajectories are of the form of a straight path and two slight turning paths. In these 3 examples, all the methods exhibit similar prediction results and are very close to the ground truth trajectories. Figure 4(d) shows the case of an abrupt turn of a pedestrian. With both the Motion and Social encoders incorporated, SPP clearly gives a better predicted trajectory compared to its two variants.

In Fig. 5, we compare the predicted trajectories of SPP when different grid sizes are used. We set the grid size n to 3 (shown as yellow trajectories), 5 (red) and 7 (pink). Figure 5(a) shows simple prediction cases where the trajectories are almost straight lines. In these examples, it appears that the value of n does not have much effect on the prediction results. For slightly turning examples (Fig. 5(b)), the SPP method trained for the 3 grid sizes can also generate plausible trajectories. In Fig. 5(c), three POIs respectively leave the scene with an almost straight line path, abrupt 90° change of direction, and an S-turn. For these cases, all the three settings are still working fine, but $n = 7$ is slightly better. More abrupt turning examples are shown in Fig. 5(d). Compared to smaller grid sizes, the SPP model with $n = 7$ gives the best predicted trajectories.

5 Conclusion

We have presented an LSTM based method for pedestrian trajectory prediction which combines both motion and social information. Our proposed SPP method has a Motion encoder and a Social encoder. The former merges the location and velocity terms of the input trajectories through a mixing attention layer while the latter analyzes the social information captured in a social pyramid tensor. Our SPP method has been evaluated on different real world datasets and the effectiveness of the two encoders has been analyzed in an ablation study on three variants of SPP. Both quantitative and qualitative results in our experiments demonstrate competitive prediction accuracy from our method compared to state-of-art trajectory prediction methods.

References

1. Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Li, F.F., Savarese, S.: Social LSTM: human trajectory prediction in crowded spaces. In: CVPR, pp. 961–971, June 2016
2. Bartoli, F., Lisanti, G., Ballan, L., Del Bimbo, A.: Context-aware trajectory prediction. arXiv preprint arXiv:1705.02503 (2017)
3. Bhattacharyya, A., Fritz, M., Schiele, B.: Long-term on-board prediction of people in traffic scenes under uncertainty. In: CVPR, June 2018

4. Chen, M., Ding, G., Zhao, S., Chen, H., Liu, Q., Han, J.: Reference based LSTM for image captioning. In: AAAI, pp. 3981–3987 (2017)
5. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555 (2014)
6. Deo, N., Trivedi, M.M.: Convolutional social pooling for vehicle trajectory prediction. In: CVPR, June 2018
7. Fernando, T., Denman, S., Sridharan, S., Fookes, C.: Soft+ hardwired attention: an LSTM framework for human trajectory prediction and abnormal event detection. arXiv preprint arXiv:1702.05552 (2017)
8. Fernando, T., Denman, S., Sridharan, S., Fookes, C.: Tracking by prediction: a deep generative model for mutli-person localisation and tracking. In: IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1122–1132. IEEE (2018)
9. Graves, A.: Generating sequences with recurrent neural networks. arXiv preprint arXiv:1308.0850 (2013)
10. Graves, A., Jaitly, N.: Towards end-to-end speech recognition with recurrent neural networks. In: ICML, vol. 14, pp. 1764–1772 (2014)
11. Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S., Alahi, A.: Social GAN: socially acceptable trajectories with generative adversarial networks. In: CVPR, June 2018
12. Hasan, I., Setti, F., Tsesmelis, T., Del Bue, A., Galasso, F., Cristani, M.: MX-LSTM: mixing tracklets and vislets to jointly forecast trajectories and head poses. In: CVPR, June 2018
13. Helbing, D., Molnar, P.: Social force model for pedestrian dynamics. Phys. Rev. E **51**(5), 4282 (1995)
14. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
15. Kim, K., Lee, D., Essa, I.: Gaussian process regression flow for analysis of motion trajectories. In: ICCV, pp. 1164–1171. IEEE (2011)
16. Kim, S., et al.: BRVO: predicting pedestrian trajectories using velocity-space reasoning. Int. J. Robot. Res. **34**(2), 201–217 (2015)
17. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
18. 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: CVPR (2017)
19. Li, S., Li, W., Cook, C., Zhu, C., Gao, Y.: Independently Recurrent Neural Network (IndRNN): building a longer and deeper RNN. In: CVPR, June 2018
20. Li, Y.: A deep spatiotemporal perspective for understanding crowd behavior. IEEE Trans. Multimed., 1–8 (2018). https://doi.org/10.1109/TMM.2018.2834873
21. Li, Y.: Pedestrian path forecasting in crowd: a deep spatio-temporal perspective. In: Proceedings of the ACM on Multimedia Conference, pp. 235–243. ACM (2017)
22. Liu, J., Wang, G., Hu, P., Duan, L.Y., Kot, A.C.: Global context-aware attention LSTM networks for 3D action recognition. In: CVPR, pp. 1647–1656 (2017)
23. Lv, J., Li, Q., Sun, Q., Wang, X.: T-CONV: a convolutional neural network for multi-scale taxi trajectory prediction. In: 2018 IEEE International Conference on Big Data and Smart Computing (BigComp), pp. 82–89. IEEE (2018)
24. Mehran, R., Oyama, A., Shah, M.: Abnormal crowd behavior detection using social force model. In: CVPR, pp. 935–942. IEEE (2009)
25. Nikhil, N., Morris, B.T.: Convolutional neural network for trajectory prediction. arXiv preprint arXiv:1809.00696 (2018)

26. Pellegrini, S., Ess, A., Schindler, K., Van Gool, L.: You'll never walk alone: modeling social behavior for multi-target tracking. In: ICCV, pp. 261–268. IEEE (2009)
27. Ren, J.S., et al.: Look, listen and learn - a multimodal LSTM for speaker identification. In: AAAI, pp. 3581–3587 (2016)
28. 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)
29. Su, H., Dong, Y., Zhu, J., Ling, H., Zhang, B.: Crowd scene understanding with coherent recurrent neural networks. In: IJCAI, pp. 3469–3476 (2016)
30. Su, H., Zhu, J., Dong, Y., Zhang, B.: Forecast the plausible paths in crowd scenes. In: IJCAI, pp. 2772–2778 (2017)
31. Sun, L., Yan, Z., Mellado, S.M., Hanheide, M., Duckett, T.: 3DOF pedestrian trajectory prediction learned from long-term autonomous mobile robot deployment data. arXiv preprint arXiv:1710.00126 (2017)
32. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: NIPS, pp. 3104–3112 (2014)
33. Vaswani, A., et al.: Attention is all you need. In: NIPS, pp. 5998–6008 (2017)
34. Vemula, A., Muelling, K., Oh, J.: Social attention: modeling attention in human crowds. In: ICRA, pp. 1–7, May 2018. https://doi.org/10.1109/ICRA.2018.8460504
35. Vemula, A., Muelling, K., Oh, J.: Modeling cooperative navigation in dense human crowds. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 1685–1692. IEEE (2017)
36. Wang, J.M., Fleet, D.J., Hertzmann, A.: Gaussian process dynamical models for human motion. IEEE Trans. Pattern Anal. Mach. Intell. **30**(2), 283–298 (2008)
37. Xie, D., Todorovic, S., Zhu, S.C.: Inferring "dark matter" and "dark energy" from videos. In: ICCV, December 2013
38. Xu, Y., Piao, Z., Gao, S.: Encoding crowd interaction with deep neural network for pedestrian trajectory prediction. In: CVPR, June 2018
39. Xue, H., Huynh, D., Reynolds, M.: Bi-Prediction: pedestrian trajectory prediction based on bidirectional LSTM classification. In: International Conference on Digital Image Computing: Techniques and Applications (DICTA), pp. 307–314 (2017)
40. Xue, H., Huynh, D.Q., Reynolds, M.: SS-LSTM: a hierarchical LSTM model for pedestrian trajectory prediction. In: IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1186–1194. IEEE (2018)
41. Yamaguchi, K., Berg, A.C., Ortiz, L.E., Berg, T.L.: Who are you with and where are you going? In: CVPR, pp. 1345–1352. IEEE (2011)
42. Yi, S., Li, H., Wang, X.: Understanding pedestrian behaviors from stationary crowd groups. In: CVPR, pp. 3488–3496 (2015)
43. Yi, S., Li, H., Wang, X.: Pedestrian behavior understanding and prediction with deep neural networks. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 263–279. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_16
44. Zhou, B., Wang, X., Tang, X.: Understanding collective crowd behaviors: learning a mixture model of dynamic pedestrian-agents. In: CVPR, pp. 2871–2878. IEEE (2012)
45. Zhu, W., et al.: Co-occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks. In: AAAI, pp. 3697–3703 (2016)
46. Zou, H., Su, H., Song, S., Zhu, J.: Understanding human behaviors in crowds by imitating the decision-making process. In: AAAI (2018)