Social LSTM:
Human Trajectory Prediction in Crowded Spaces

Alexandre Alahi*, Kratarth Goel*, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, Silvio Savarese
Stanford University
{alahi,kratarth,vigneshr,arobicqu,feifeili,ssilvio}@cs.stanford.edu

Abstract

Pedestrians follow different trajectories to avoid obstacles and accommodate fellow pedestrians. Any autonomous vehicle navigating such a scene should be able to foresee the future positions of pedestrians and accordingly adjust its path to avoid collisions. This problem of trajectory prediction can be viewed as a sequence generation task, where we are interested in predicting the future trajectory of people based on their past positions. Following the recent success of Recurrent Neural Network (RNN) models for sequence prediction tasks, we propose an LSTM model which can learn general human movement and predict their future trajectories. This is in contrast to traditional approaches which use hand-crafted functions such as Social forces. We demonstrate the performance of our method on several public datasets. Our model outperforms state-of-the-art methods on some of these datasets . We also analyze the trajectories predicted by our model to demonstrate the motion behaviour learned by our model.

1. Introduction

Humans have the innate ability to "read" one another. When people walk in a crowed public space such as a sidewalk, an airport terminal, or a shopping mall, they obey a large number of (unwritten) common sense rules and comply with social conventions. For instance, as they consider where to move next, they respect personal space and yield right-of-way. The ability to model these rules and use them to understand and predict human motion in complex real world environments is extremely valuable for a wide range of applications - from the deployment of socially-aware robots [41] to the design of intelligent tracking systems [43] in smart environments.

Predicting the motion of human targets while taking into account such common sense behavior, however, is an extremely challenging problem. This requires understanding

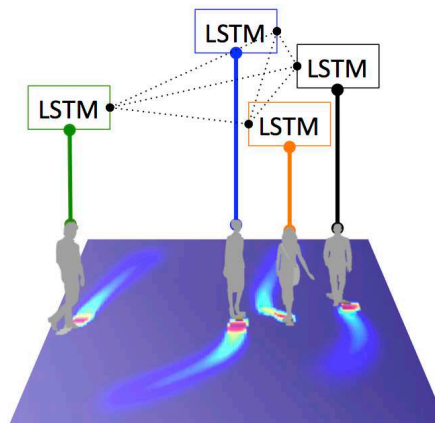*indicates equal contribution

Figure 1. The goal of this paper is to predict the motion dynamics in crowded scenes - This is, however, a challenging task as the motion of each person is typically affected by their neighbors. We propose a new model which we call "Social" LSTM (Social-LSTM) which can jointly predict the paths of all the people in a scene by taking into account the common sense rules and social conventions that humans typically utilize as they navigate in shared environments. The predicted distribution of their future trajectories is shown in the heat-map.

the complex and often subtle interactions that take place between people in crowded spaces. Recent research in computer vision has successfully addressed some of these challenges. Kitani *et. al.* [32] have demonstrated that the inferred knowledge about the semantics of the static environment (e.g., location of sidewalks, extension of grass areas, etc) helps predict the trajectory of pedestrians in future instants more accurately than a model which ignores the scene information. Pioneering works by [24, 50, 35] have also proposed ways to model *human-human* interactions (often called "social forces") to increase robustness and accuracy in multi-target tracking problems.

However, most of these works are limited by the following two assumptions. i) They use hand-crafted functions to model "interactions" for specific settings rather than inferring them in a data-driven fashion. This results in fa-

voring models that capture simple interactions (e.g. repulsion/attractions) and might fail to generalize for more complex crowded settings. ii) They focus on modeling interactions among people in close proximity to each other (to avoid immediate collisions). However, they do not anticipate interactions that could occur in the more distant future.

In this work, we propose an approach that can address both challenges through a novel data-driven architecture for predicting human trajectories in future instants. Inspired by the recent success of Long-Short Term Memory networks (LSTM) for different sequence prediction tasks such as handwriting [20] and speech [21] generation, we extend them for human trajectory prediction as well. While LSTMs have the ability to learn and reproduce long sequences, they do not capture dependencies between multiple correlated sequences.

We address this issue through a novel architecture which connects the LSTMs corresponding to nearby sequences. In particular, we introduce a "Social" pooling layer which allows the LSTMs of spatially proximal sequences to share their hidden-states with each other. This architecture, which we refer to as the "Social-LSTM", can automatically learn typical interactions that take place among trajectories which coincide in time. This model leverages existing human trajectory datasets without the need for any additional annotations to learn common sense rules and conventions that humans observe in social spaces.

Finally, we demonstrate that our Social-LSTM is capable of predicting trajectories of pedestrians much more accurately than state-of-the-art methods on two publicly available datasets: ETH [49], and UCY [39]. We also analyze the trajectory patterns generated by our model to understand the social constraints learned from the trajectory datasets.

2. Related work

Human-human interactions   Pioneering work from Helbing and Molnar [24] presented a pedestrian motion model with attractive and repulsive forces referred to as the *Social Force* model. This has been shown to achieve competitive results even on modern pedestrian datasets [39, 49]. This method was later extended to robotics [41] and activitiy understanding [43, 73, 50, 38, 37, 9, 10].

Similar approaches have been used to model human-human interactions with strong priors for the model. Treuille *et. al.* [62] use continuum dynamics, Antonini *et. al.* [2] propose a Discrete Choice framework and Wang *et. al.* [69], Tay *et. al.* [59] use Gaussian processes. Such functions have alse been used to study stationary groups [74, 48]. These works target smooth motion paths and do not handle the problems associated with discretization.

Another line of work uses well-engineered features and attributes to improve tracking and forecasting. Alahi *et. al.* [1] presented a social affinity feature by learning from human trajectories in crowd their relative positions, while Yu *et. al.* [74] proposed the use of human-attributes to improve forecasting in dense crowds. They also use an agent-based model similar to [6]. Rodriguez et al. [54] analyze videos with high-density crowds to track and count people.

Most of these models provide hand-crafted energy potentials based on relative distances and rules for specific scenes. In contrast, we propose a method to learn human-human interactions in a more generic data-driven fashion.

Activity forecasting   Activity forecasting models try to predict the motion and/or action to be carried out by people in a video. A large body of work learns motion patterns through clustering trajectories [26, 30, 46, 77]. More approaches can be found in [45, 52, 34, 3, 16, 33]. Kitani *et. al.* in [32] use *Inverse Reinforcement Learning* to predict human paths in static scenes. They infer walkable paths in a scene by modeling human-space interactions. Walker *et al.* in [68] predict the behavior of generic agents (*e.g.*, a vehicle) in a visual scene given a large collection of videos. Ziebart et al. [78, 23] presented a planning based approach. Turek *et al.* [63, 40] used a similar idea to identify the functional map of a scene. Other approaches like [27, 19, 42, 36] showed the use of scene semantics to predict goals and paths for human navigation. Scene semantics has also been used to predict multiple object dynamics [17, 36, 34, 28]. These works are mostly restricted to the use of static scene information to predict human motion or activity. In our work, we focus on modeling dynamic crowd interactions for path prediction.

More recent works have also attempted to predict future human actions. In particular, Ryoo *et. al.* [55, 8, 71, 67, 44, 58] forecast actions in streaming videos. More relevant to our work, is the idea of using a RNN mdoel to predict future events in videos [53, 57, 66, 56, 31]. Along similar lines, we predict future trajectories in scenes.

RNN models for sequence prediction   Recently Recurrent Neural Networks (RNN) and their variants including Long Short Term Memory (LSTM) [25] and Gated Recurrent Units [12] have proven to be very successful for sequence prediction tasks: speech recognition [21, 11, 13], caption generation [64, 29, 75, 15, 72], machine translation [4], image/video classification [7, 22, 70, 47], human dynamics [18] to name a few. RNN models have also proven to be effective for tasks with densely connected data such as semantic segmentation [76], scene parsing [51] and even as an alternative to Convolutional Neural Networks [65]. These works show that RNN models are capable of learning the dependencies between spatially correlated data such as image pixels. This motivates us to extend the sequence generation model from Graves et al. [20] to our setting. In particular, Graves et al. [20] predict isolated handwriting

sequences; while in our work we jointly predict multiple correlated sequences corresponding to human trajectories.

3. Our model

Humans moving in crowded scenes adapt their motion based on the behaviour of other people in their vicinity. For instance, a person could completely alter his/her path or stop momentarily to accommodate a group of people moving towards him. Such deviation in trajectory cannot be predicted by observing the person in isolation. Neither, can it be predicted with simple "repulsion" or "attraction" functions (the traditional *social forces* models [24, 43, 73, 50])

This motivates us to build a model which can account for the behavior of other people within a large neighborhood, while predicting a person's path. In this section, we describe our pooling based LSTM model (Fig. 2) which jointly predicts the trajectories of all the people in a scene. We refer to this as the "Social" LSTM model.

Problem formulation We assume that each scene is first preprocessed to obtain the spatial coordinates of the all people at different time-instants. Previous work follow this convention as well [41, 1]. At any time-instant t, the ith person in the scene is represented by his/her xy-coordinates (x_t^i, y_t^i). We observe the positions of all the people from time 1 to T_{obs}, and predict their positions for time instants T_{obs+1} to T_{pred}. This task can also be viewed as a sequence generation problem [20], where the input sequence corresponds to the observed positions of a person and we are interested in generating an output sequence denoting his/her future positions at different time-instants.

3.1. Social LSTM

Every person has a different motion pattern: they move with different velocities, acceleration and have different gaits. We need a model which can understand and learn such person-specific motion properties from a limited set of initial observations corresponding to the person.

Long Short-Term Memory (LSTM) networks have been shown to successfully learn and generalize the properties of isolated sequences like handwriting [20] and speech [21]. Inspired by this, we develop a LSTM based model for our trajectory prediction problem as well. In particular, we have one LSTM for each person in a scene. This LSTM learns the state of the person and predicts their future positions as shown in Fig. 2. The LSTM weights are shared across all the sequences.

However, the naive use of one LSTM model per person does not capture the interaction of people in a neighborhood. The vanilla LSTM is agnostic to the behaviour of other sequences. We address this limitation by connecting neighboring LSTMs through a new pooling strategy visualized in Fig. 3,2.
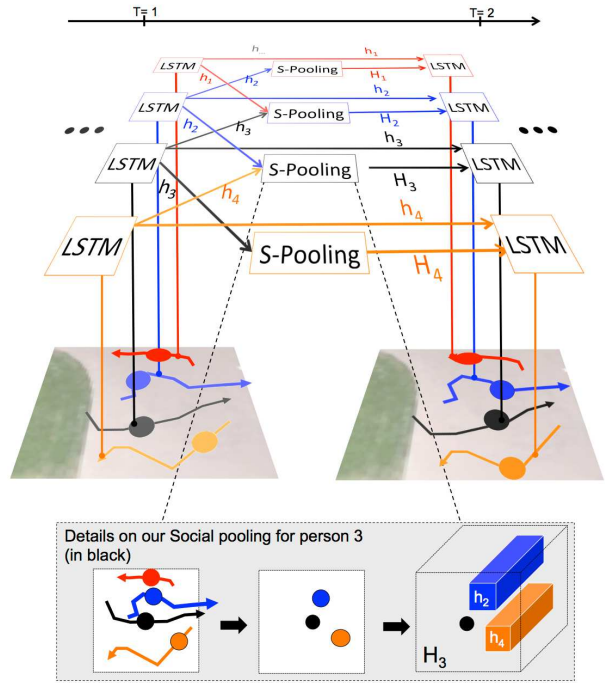

Figure 2. Overview of our Social-LSTM method. We use a separate LSTM network for each trajectory in a scene. The LSTMs are then connected to each other through a Social pooling (S-pooling) layer. Unlike the traditional LSTM, this pooling layer allows spatially proximal LSTMs to share information with each other. The variables in the figure are explained in Eq. 2. The bottom row shows the S-pooling for one person in the scene. The hidden-states of all LSTMs within a certain radius are pooled together and used as an input at the next time-step.

Social pooling of hidden states Individuals adjust their paths by implicitly reasoning about the motion of neighboring people. These neighbors in-turn are influenced by others in their immediate surroundings and could alter their behaviour over time. We expect the hidden states of an LSTM to capture these time varying motion-properties. In order to jointly reason across multiple people, we share the states between neighboring LSTMS. This introduces a new challenge: every person has a different number of neighbors and in very dense crowds [1], this number could be prohibitively high.

Hence, we need a compact representation which combines the information from all neighboring states. We handle this by introducing "Social" pooling layers as shown in Fig. 2. At every time-step, the LSTM cell receives pooled hidden-state information from the LSTM cells of neighbors.

While pooling the information, we try to preserve the spatial information through grid based pooling as explained below.

The hidden state h_i^t of the LSTM at time t captures the latent representation of the ith person in the scene at that instant. We share this representation with neighbors by building a "Social" hidden-state tensor H_t^i. Given a hidden-state dimension D, and neighborhood size N_o, we construct a $N_o \times N_o \times D$ tensor H_t^i for the ith trajectory:

$$H_t^i(m, n, :) = \sum_{j \in \mathcal{N}_i} \mathbf{1}_{mn}[x_t^j - x_t^i, y_t^j - y_t^i] h_{t-1}^j, \qquad (1)$$

where h_{t-1}^j is the hidden state of the LSTM corresponding to the jth person at $t-1$, $\mathbf{1}_{mn}[x, y]$ is an indicator function to check if (x, y) is in the (m, n) cell of the grid, and $\mathcal{N}_i$ is the set of neighbors corresponding to person i. This pooling operation is visualized in Fig. 3.

We embed the pooled Social hidden-state tensor into a vector a_i^t and the co-ordinates into e_i^t. These embeddings are concatenated and used as the input to the LSTM cell of the corresponding trajectory at time t. This introduces the following recurrence:

$$
\begin{aligned}
e_t^i &= \phi(x_t^i, y_t^i; W_e) \qquad &(2)\\
a_i^t &= \phi(H_t^i; W_a),\\
h_i^t &= \text{LSTM}\left(h_i^{t-1}, e_t^i, a_t^i; W_l\right)
\end{aligned}
$$

where $\phi(.)$ is an embedding function with ReLU non-linearlity, W_e and W_a are embedding weights. The LSTM weights are denoted by W_l.

Position estimation The hidden-state at time t is used to predict the distribution of the trajectory position $(\hat{x}, \hat{y})_{t+1}^i$ at the next time-step $t+1$. Similar to Graves et al. [20], we assume a bivariate Gaussian distribution parametrized by the mean $\mu_{t+1}^i = (\mu_x, \mu_y)_{t+1}^i$, standard deviation $\sigma_{t+1}^i = (\sigma_x, \sigma_y)_{t+1}^i$ and correlation coefficient ρ_{t+1}^i. These parameters are predicted by a linear layer with a $5 \times D$ weight matrix W_p. The predicted coordinates $(\hat{x}_t^i, \hat{y}_t^i)$ at time t are given by

$$(\hat{x}, \hat{y})_t^i \sim \mathcal{N}(\mu_t^i, \sigma_t^i, \rho_t^i) \qquad (3)$$

The parameters of the LSTM model are learned by minimizing the negative log-Likelihood loss (L^i for the ith trajectory):

$$
\begin{aligned}
\left[\mu_t^i, \sigma_t^i, \rho_t^i\right] &= W_p h_i^{t-1} \qquad (4)\\
L^i(W_e, W_l, W_p) &= -\sum_{t=T_{obs}+1}^{T_{pred}} \log\left(\mathbb{P}(x_t^i, y_t^i | \sigma_t^i, \mu_t^i, \rho_t^i)\right),
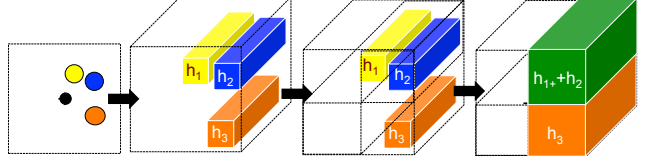\end{aligned}
$$

Figure 3. We show the Social pooling for the person represented by a black-dot. We pool the hidden states of the neighbors (shown in yellow, blue and orange) within a certain spatial distance. The pooling partially preserves the spatial information of neighbors as shown in the last two steps.

We train the model by minimizing this loss for all the trajectories in a training dataset. Note that our "Social" pooling layer does not introduce any additional parameters.

An important distinction from the traditional LSTM is that the hidden states of multiple LSTMs are coupled by our "Social" pooling layer and we jointly back-propagate through multiple LSTMs in a scene at every time-step.

Occupancy map pooling The "Social" LSTM model can be used to pool any set of features from neighboring trajectories. As a simplification, we also experiment with a model which only pools the co-ordinates of the neighbors (referred to as O-LSTM in the experiments Sect. 4). This is a reduction of the original model and does not require joint back-propagation across all trajectories during training. This model can still learn to reposition a trajectory to avoid immediate collision with neighbors. However, in the absence of more information from neighboring people, this model would be unable to smoothly change paths to avoid future collisions.

For a person i, we modify the definition of the tensor H_t^i, as a $N_o \times N_o$ matrix at time t centered at the person's position, and call it the occupancy map O_t^i. The positions of all the neighbors are pooled in this map. The m, n element of the map is simply given by:

$$O_t^i(m, n) = \sum_{j \in \mathcal{N}_i} \mathbf{1}_{mn}[x_t^j - x_t^i, y_t^j - y_t^i], \qquad (5)$$

where $\mathbf{1}_{mn}[.]$ is an indicator function as defined previously. This can also be viewed as a simplification of the social tensor in Eq. 1 where the hidden state vector is replaced by a constant value indicating the presence or absence of neighbors in the corresponding cell.

The vectorized occupancy map is used in place of H_t^i in Eq. 2 while learning this simpler model.

Inference for path prediction During test time, we use the trained Social-LSTM models to predict the future position $(\hat{x}_t^i, \hat{y}_t^i)$ of the ith person. From time T_{obs+1} to T_{pred},

964

we use the predicted position $(\hat{x}_t^i, \hat{y}_t^i)$ from the previous Social-LSTM cell in place of the true coordinates (x_t^i, y_t^i) in Eq. 2. The predicted positions are also used to replace the actual coordinates while constructing the Social hidden-state tensor H_t^i in Eq. 1 or the occupancy map O_t^i in Eq. 5.

3.2. Implementation details

We use an embedding dimension of 64 for the spatial co-ordinates before using them as input to the LSTM. We set the spatial pooling size N_o to be 32 and use a 8x8 sum pooling window size without overlaps. We used a fixed hidden state dimension of 128 for all the LSTM models. Additionally, we also use an embedding layer with ReLU (rectified Linear Units) non-linearity on top of the pooled hidden-state features, before using them for calculating the hidden state tensor H_t^i. The hyper-parameters were chosen based on cross-validation on a synthetic dataset. This synthetic was generated using a simulation that implemented the social forces model. This synthetic data contained trajectories for hundreds of scenes with an average crowd density of 30 per frame. We used a learning rate of 0.003 and RMS-prop [14] for training the model. The Social-LSTM model was trained on a single GPU with a Theano [5] implementation.

4. Experiments

In this section, we present experiments on two publicly available human-trajectory datasets: ETH [49] and UCY [39]. The ETH dataset contains two scenes each with 750 different pedestrians and is split into two sets (*ETH* and *Hotel*). The UCY dataset contains two scenes with 786 people. This dataset has 3-components: *ZARA-01*, *ZARA-02* and *UCY*. In total, we evaluate our model on 5 sets of data. These datasets represent real world crowded settings with thousands of non-linear trajectories. As shown in [49], these datasets also cover challenging group behaviours such as couples walking together, groups crossing each other and groups forming and dispersing in some scenes.

We report the prediction error with three different metrics. Similar to Pellegrini et al. [49] we use:

1. *Average displacement error* - The mean square error (MSE) over all estimated points of a trajectory and the true points. This was introduced in Pellegirini et al. [49].

2. *Final displacement error* - The distance between the predicted final destination and the true final destination at end of the prediction period T_{pred}.

3. *Average non-linear displacement error* - The is the MSE at the non-linear regions of a trajectory. Since most errors in trajectory-prediction occur during non-linear turns arising from human-human interactions,

we explicitly evaluate the errors around these regions. We set a heuristic threshold on the norm of the second derivative to identify non-linear regions.

In order to make full use of the datasets while training our models, we use a leave-one-out approach. We train and validate our model on 4 sets and test on the remaining set. We repeat this for all the 5 sets. We also use the same training and testing procedure for other baseline methods used for comparison.

During test time, we observe a trajectory for $3.2secs$ and predict their paths for the next $4.8secs$. At a frame rate of 0.4, this corresponds to observing 8 frames and predicting for the next 12 frames. This is similar to the setting used by [49, 39]. In Tab. 4, we compare the performance of our model with state-of-the-art methods as well as multiple control settings:

- *Linear model (**Lin.**)* We use an off-the-shelf Kalman filter to extrapolate trajectories with assumption of linear acceleration.

- *Collision avoidance (**LTA**).* We report the results of a simplified version of the Social Force [73] model which only uses the collision avoidance energy, commonly referred to as linear trajectory avoidance.

- *Social force (**SF**).* We use the implementation of the Social Force model from [73] where several factors such as group affinity and predicted destinations have been modeled.

- *Iterative Gaussian Process (**IGP**).* We use the implementation of the IGP from [61]. Unlike the other baselines, IGP also uses additional information about the final destination of a person.

- *Our Vanilla LSTM (**LSTM**).* This is a simplified setting of our model where we remove the "Social" pooling layers and treat all the trajectories to be independent of each other.

- *Our LSTM with occupancy maps (**O-LSTM**).* We show the performance of a simplified version of our model (presented in Sec. 3.1). As a reminder, the model only pools the coordinates of the neighbors at every time-instance.

The naive linear model produces high prediction errors, which are more pronounced around non-linear regions as seen from the average non-linear displacement error. The vanilla LSTM outperforms this linear baseline since it can extrapolate non-linear curves as shown in Graves et al. [20]. However, this simple LSTM is noticeably worse than the Social Force and IGP models which explicitly model

Metric	Methods	Lin	LTA	SF [73]	IGP* [60]	LSTM	our O-LSTM	our Social-LSTM
Avg. disp. error	ETH [49]	0.80	0.54	0.41	**0.20**	0.60	0.49	0.50
	HOTEL [49]	0.39	0.38	0.25	0.24	0.15	**0.09**	0.11
	ZARA 1 [39]	0.47	0.37	0.40	0.39	0.43	**0.22**	**0.22**
	ZARA 2 [39]	0.45	0.40	0.40	0.41	0.51	0.28	**0.25**
	UCY [39]	0.57	0.51	0.48	0.61	0.52	0.35	**0.27**
	Average	0.53	0.44	0.39	0.37	0.44	0.28	**0.27**
Avg. non-linear disp. error	ETH [49]	0.95	0.70	0.49	0.39	0.28	**0.24**	0.25
	HOTEL [49]	0.55	0.49	0.38	0.34	0.09	**0.06**	0.07
	ZARA 1 [39]	0.56	0.39	0.41	0.54	0.24	**0.13**	**0.13**
	ZARA 2 [39]	0.44	0.41	0.39	0.43	0.30	0.20	**0.16**
	UCY [39]	0.62	0.57	0.54	0.62	0.31	0.20	**0.16**
	Average	0.62	0.51	0.44	0.46	0.24	0.17	**0.15**
Final disp. error	ETH [49]	1.31	0.77	0.59	**0.43**	1.31	1.06	1.07
	HOTEL [49]	0.55	0.64	0.37	0.37	0.33	**0.20**	0.23
	ZARA 1 [39]	0.89	0.66	0.60	0.39	0.93	**0.46**	0.48
	ZARA 2 [39]	0.91	0.72	0.68	0.42	1.09	0.58	**0.50**
	UCY [39]	1.14	0.95	0.78	1.82	1.25	0.90	**0.77**
	Average	0.97	0.74	**0.60**	0.69	0.98	0.64	0.61

Table 1. Quantitative results of all the methods on all the datasets. We present the performance metrics as follows: First 6 rows are the Average displacement error, row 7 to 12 are the Average displacement error for non-linear regions, and the final 6 rows are the Final displacement error. All methods forecast trajectories for a fixed period of 4.8 seconds. (*) Note that IGP uses the intended ground truth destination of a person during test time unlike other methods.

human-human interactions. This shows the need to account for such interactions.

Our Social pooling based LSTM and O-LSTM outperform the heavily engineered Social Force and IGP models in almost all datasets. In particular, the error reduction is more significant in the case of the UCY datasets as compared to ETH. This can be explained by the different crowd densities in the two datasets: UCY contains more crowded regions with a total of $32K$ non-linearities as opposed to the more sparsely populated ETH scenes with only $15K$ non-linear regions.

In the more crowded UCY scenes, the deviation from linear paths is more dominated by human-human interactions. Hence, our model which captures neighborhood interactions achieves a higher gain in UCY datasets. The pedestrians' intention to reach a certain destination plays a more dominant role in the ETH datasets. Consequently, the IGP model which knows the true final destination during testing achieves lower errors in parts of this dataset.

In the case of ETH, we also observe that the occupancy and Social LSTM errors are at par with each other and in general better than the Social force model. Again, our Social-LSTM outperforms O-LSTM in the more crowded UCY datasets. This shows the advantage of pooling the entire hidden state to capture complex interactions in dense crowds.

4.1. Analyzing the predicted paths

Our quantitative evaluation in the Sec. 4 shows that the learned Social-LSTM model outperforms state-of-the-art methods on standard datasets. In this section, we try to gain more insights on the actual behaviour of our model in different crowd settings. We qualitatively study the performance of our Social-LSTM method on social scenes where individuals interact with each others in a specific pattern.

We present an example scene occupied by four individuals in Figure 4. We visualize the distribution of the paths predicted by our model at different time-instants. The first and third rows in Figure 4 show the current position of each person as well as their true trajectory (solid line for the future path and dashed line for the past). The second and fourth rows show our Social-LSTM prediction for the next 12.4 secs. In these scenes, we observe three people(2,3,4) walking close to each other and a fourth person(1) walking farther away from them.

Our model predicts a linear path for person(1) at all times. The distribution for person (1) is similar across time indicating that the speed of the person is constant.

We can observe more interesting patterns in the predicted trajectories for the 3-person group. In particular, our model makes intelligent route choices to yield for others and pre-empt future collisions. For instance, at time-steps 2, 4, and 5 our model predicts a deviation from the linear paths for person(3) and person(4), even before the start of the actual turn.
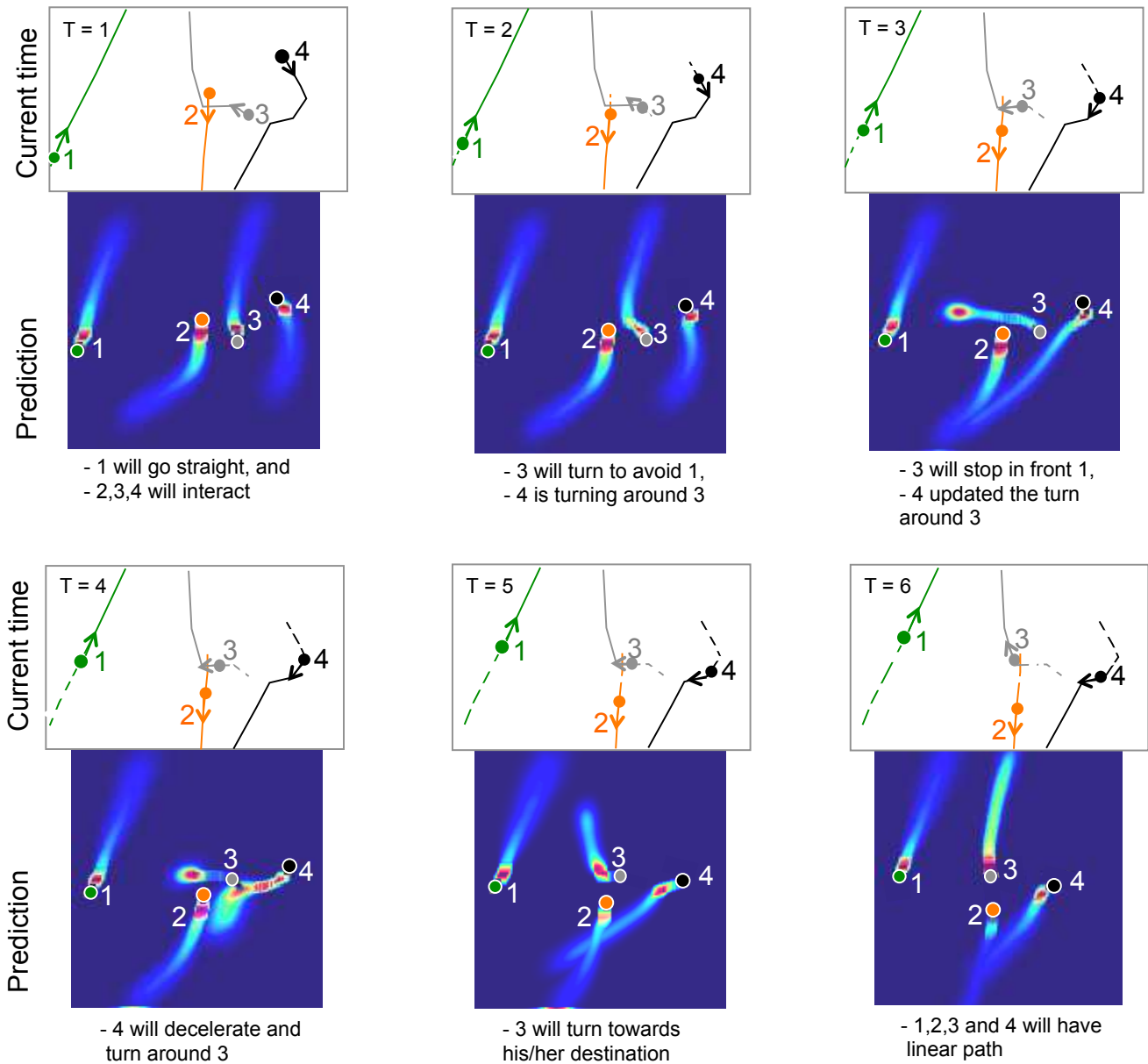
Figure 4. We visualize the probability distribution of the predicted paths for 4 people moving in a scene across 6 time steps. The sub-caption describes what our model is predicting. At each time-step: the solid lines in rows 1,3 represents the ground-truth future trajectories, the dashed lines refer to the observed positions till that time-step and the dots denote the position at that time-step. We notice that our model often correctly predicts the future paths in challenging settings with non-linear motions. We analyze these figures in more details in Sec. 4.1. Note that T stands for time and the id (1 to 4) denote person ids. *More examples are provided in the supplementary material.*

At time-step 3 and 4, we notice that the Social-LSTM predicts a "halt" for person(3) in order to yield for person(1). Interestingly at time-step 4, the location of the haling point is updated to match the true turning-point in the path. At the next time-step, with more observations, the model is able to correctly predict the full turn anchored at that point.

In Figure 5, we illustrate the prediction results of our Social-LSTM, the SF model [49] and the linear baseline on

one of the ETH datasets. When people walk in a group or as *e.g.* a couple, our model is able to jointly predict their trajectories. It is interesting to note that unlike Social Forces[73] we do not explicitly model group behavior. However, our model is better at predicting grouped trajectories in a holistic fashion. In the last row of Figure 5, we show some failure cases, *i.e.,* when our predictions are worse than previous works. We either predict a a linear path (2nd column) or de-
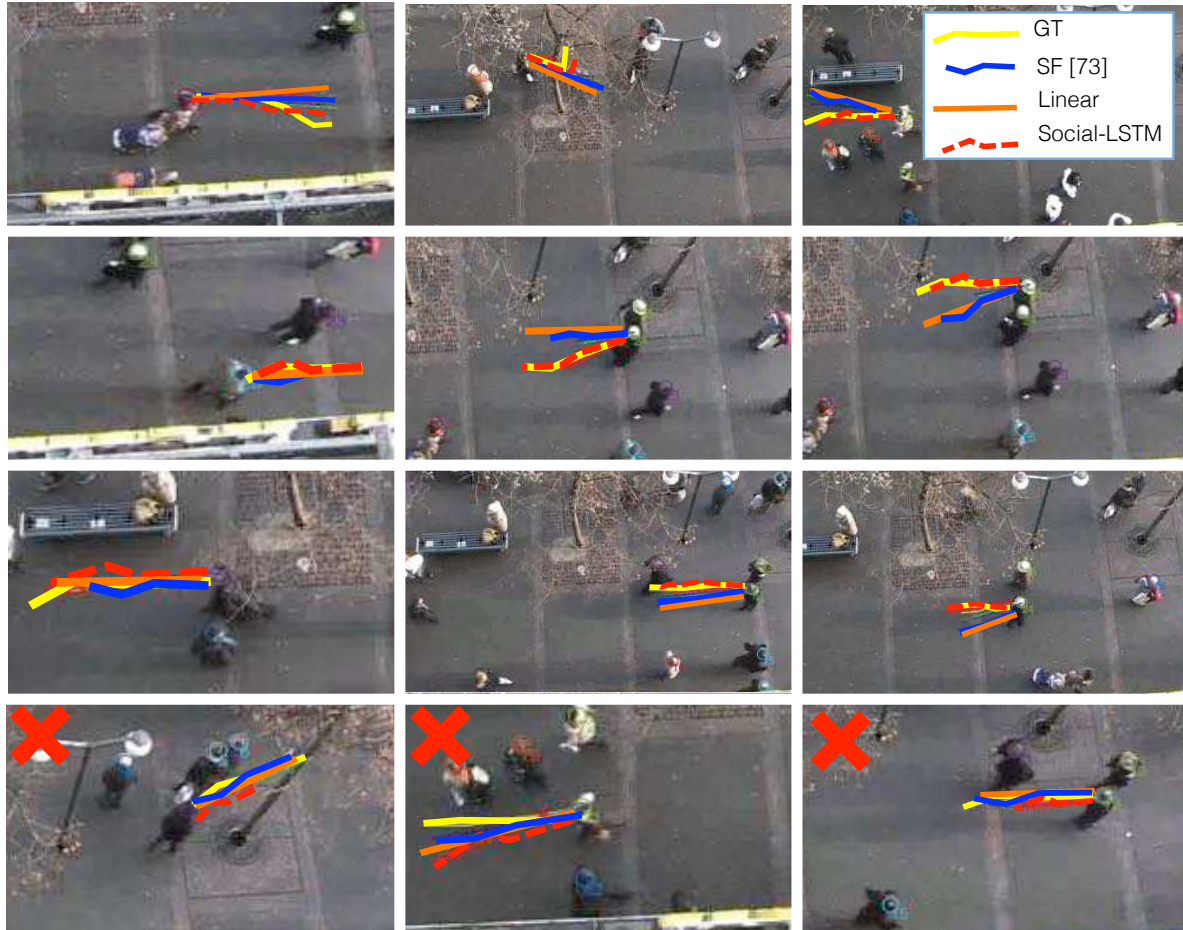
Figure 5. Illustration of our Social-LSTM method predicting trajectories. On the first 3 rows, we show examples where our model successfully predicts the trajectories with small errors (in terms of position and speed). We also show other methods such as Social Forces [73] and linear method. The last row represents failure cases, e.g., person slowed down or took a linear path. Nevertheless, our Social-LSTM method predicts a plausible path. The results are shown on ETH dataset [49].

celerate earlier (1st and 3rd column) than needed. Although the trajectories do not match the ground-truth in these cases, our Social-LSTM still outputs "plausible" trajectories, *i.e.* trajectories that humans could have taken. For instance, in the first and third columns, our model slows down to avoid a potential collision with the person ahead.

5. Conclusions

We have presented a LSTM-based model that can jointly reason across multiple individuals to predict human trajectories in a scene. We use one LSTM for each trajectory and share the information between the LSTMs through the introduction of a new Social pooling layer. We refer to the resulting model as the "Social" LSTM. Our proposed method outperforms state-of-the-art methods on two publicly available datasets. In addition, we qualitatively show that our Social-LSTM successfully predicts various non-linear be-

haviors arising from social interactions, such as a group of individuals moving together. Future work will extend our model to multi-class settings where several objects such as bicycles, skateboards, carts, and pedestrians share the same space. Each object will have its own label in the occupancy map. In addition, human-space interaction can be modeled in our framework by including the local static-scene image as an additional input to the LSTM. This could allow jointly modeling of human-human and human-space interactions in the same framework.

6. Acknowledgement

References

[1] A. Alahi, V. Ramanathan, and L. Fei-Fei. Socially-aware large-scale crowd forecasting. In *CVPR*, 2014. 2, 3

[2] G. Antonini, M. Bierlaire, and M. Weber. Discrete choice models of pedestrian walking behavior. *Transportation Research Part B: Methodological*, 40(8):667–687, 2006. 2

[3] J. Azorin-Lopez, M. Saval-Calvo, A. Fuster-Guillo, and A. Oliver-Albert. A predictive model for recognizing human behaviour based on trajectory representation. In *Neural Networks (IJCNN), 2014 International Joint Conference on*, pages 1494–1501. IEEE, 2014. 2

[4] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014. 2

[5] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio. Theano: A cpu and gpu math compiler in python. 5

[6] E. Bonabeau. Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences*, 99(suppl 3):7280–7287, 2002. 2

[7] C. Cao, X. Liu, Y. Yang, Y. Yu, J. Wang, Z. Wang, Y. Huang, L. Wang, C. Huang, W. Xu, et al. Look and think twice: Capturing top-down visual attention with feedback convolutional neural networks. *ICCV*, 2015. 2

[8] Y. Cao, D. Barrett, A. Barbu, S. Narayanaswamy, H. Yu, A. Michaux, Y. Lin, S. Dickinson, J. M. Siskind, and S. Wang. Recognize human activities from partially observed videos. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2658–2665. IEEE, 2013. 2

[9] W. Choi and S. Savarese. A unified framework for multi-target tracking and collective activity recognition. In *Computer Vision–ECCV 2012*, pages 215–230. Springer, 2012. 2

[10] W. Choi and S. Savarese. Understanding collective activitiesof people from videos. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(6):1242–1257, 2014. 2

[11] J. Chorowski, D. Bahdanau, K. Cho, and Y. Bengio. End-to-end continuous speech recognition using attention-based recurrent nn: First results. *arXiv preprint arXiv:1412.1602*, 2014. 2

[12] 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

[13] J. Chung, K. Kastner, L. Dinh, K. Goel, A. C. Courville, and Y. Bengio. A recurrent latent variable model for sequential data. *CoRR*, abs/1506.02216, 2015. 2

[14] Y. N. Dauphin, H. de Vries, J. Chung, and Y. Bengio. Rmsprop and equilibrated adaptive learning rates for non-convex optimization. *CoRR*, abs/1502.04390, 2015. 5

[15] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. *arXiv preprint arXiv:1411.4389*, 2014. 2

[16] J. Elfring, R. Van De Molengraft, and M. Steinbuch. Learning intentions for improved human motion prediction. *Robotics and Autonomous Systems*, 62(4):591–602, 2014. 2

[17] D. F. Fouhey and C. L. Zitnick. Predicting object dynamics in scenes. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 2027–2034. IEEE, 2014. 2

[18] K. Fragkiadaki, S. Levine, P. Felsen, and J. Malik. Recurrent network models for human dynamics. 2

[19] H. Gong, J. Sim, M. Likhachev, and J. Shi. Multi-hypothesis motion planning for visual object tracking. In *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, pages 619–626, Washington, DC, USA, 2011. IEEE Computer Society. 2

[20] A. Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013. 2, 3, 4, 5

[21] A. Graves and N. Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1764–1772, 2014. 2, 3

[22] K. Gregor, I. Danihelka, A. Graves, and D. Wierstra. Draw: A recurrent neural network for image generation. *arXiv preprint arXiv:1502.04623*, 2015. 2

[23] K. P. Hawkins, N. Vo, S. Bansal, and A. F. Bobick. Probabilistic human action prediction and wait-sensitive planning for responsive human-robot collaboration. In *Humanoid Robots (Humanoids), 2013 13th IEEE-RAS International Conference on*, pages 499–506. IEEE, 2013. 2

[24] D. Helbing and P. Molnar. Social force model for pedestrian dynamics. *Physical review E*, 51(5):4282, 1995. 1, 2, 3

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

[26] W. Hu, D. Xie, Z. Fu, W. Zeng, and S. Maybank. Semantic-based surveillance video retrieval. *Image Processing, IEEE Transactions on*, 16(4):1168–1181, 2007. 2

[27] C. Huang, B. Wu, and R. Nevatia. Robust object tracking by hierarchical association of detection responses. In *ECCV*, 2008. 2

[28] D.-A. Huang and K. M. Kitani. Action-reaction: Forecasting the dynamics of human interaction. In *Computer Vision–ECCV 2014*, pages 489–504. Springer, 2014. 2

[29] A. Karpathy et al. Deep fragment embeddings for bidirectional image sentence mapping. In *NIPS*, 2014. 2

[30] K. Kim, D. Lee, and I. Essa. Gaussian process regression flow for analysis of motion trajectories. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1164–1171. IEEE, 2011. 2

[31] K. Kitani, T. Okabe, Y. Sato, and A. Sugimoto. Fast unsupervised ego-action learning for first-person sports videos. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3241–3248, June 2011. 2

[32] K. M. Kitani, B. D. Ziebart, J. A. Bagnell, and M. Hebert. Activity forecasting. In *Computer Vision–ECCV 2012*, pages 201–214. Springer, 2012. 1, 2

[33] Y. Kong, D. Kit, and Y. Fu. A discriminative model with multiple temporal scales for action prediction. In *Computer Vision–ECCV 2014*, pages 596–611. Springer, 2014. 2

[34] J. F. P. Kooij, N. Schneider, F. Flohr, and D. M. Gavrila. Context-based pedestrian path prediction. In *Computer Vision–ECCV 2014*, pages 618–633. Springer, 2014. 2

[35] H. Koppula and A. Saxena. Anticipating human activities using object affordances for reactive robotic response. 2013. 1

[36] H. Kretzschmar, M. Kuderer, and W. Burgard. Learning to predict trajectories of cooperatively navigating agents. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 4015–4020. IEEE, 2014. 2

[37] L. Leal-Taixé, M. Fenzi, A. Kuznetsova, B. Rosenhahn, and S. Savarese. Learning an image-based motion context for multiple people tracking. In *CVPR*, pages 3542–3549. IEEE, 2014. 2

[38] L. Leal-Taixe, G. Pons-Moll, and B. Rosenhahn. Everybody needs somebody: Modeling social and grouping behavior on a linear programming multiple people tracker. In *ICCV Workshops*, 2011. 2

[39] A. Lerner, Y. Chrysanthou, and D. Lischinski. Crowds by example. In *Computer Graphics Forum*, volume 26, pages 655–664. Wiley Online Library, 2007. 2, 5, 6

[40] K. Li and Y. Fu. Prediction of human activity by discovering temporal sequence patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(8):1644–1657, 2014. 2

[41] M. Luber, J. A. Stork, G. D. Tipaldi, and K. O. Arras. People tracking with human motion predictions from social forces. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 464–469. IEEE, 2010. 1, 2, 3

[42] D. Makris and T. Ellis. Learning semantic scene models from observing activity in visual surveillance. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 35(3):397–408, 2005. 2

[43] R. Mehran, A. Oyama, and M. Shah. Abnormal crowd behavior detection using social force model. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 935–942. IEEE, 2009. 1, 2, 3

[44] B. Minor, J. R. Doppa, and D. J. Cook. Data-driven activity prediction: Algorithms, evaluation methodology, and applications. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 805–814. ACM, 2015. 2

[45] B. T. Morris and M. M. Trivedi. A survey of vision-based trajectory learning and analysis for surveillance. *Circuits and Systems for Video Technology, IEEE Transactions on*, 18(8):1114–1127, 2008. 2

[46] B. T. Morris and M. M. Trivedi. Trajectory learning for activity understanding: Unsupervised, multilevel, and long-term adaptive approach. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(11):2287–2301, 2011. 2

[47] J. Y.-H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. *arXiv preprint arXiv:1503.08909*, 2015. 2

[48] H. S. Park and J. Shi. Social saliency prediction. 2

[49] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 261–268. IEEE, 2009. 2, 5, 6, 7, 8

[50] S. Pellegrini, A. Ess, and L. Van Gool. Improving data association by joint modeling of pedestrian trajectories and groupings. In *Computer Vision–ECCV 2010*, pages 452–465. Springer, 2010. 1, 2, 3

[51] P. H. Pinheiro and R. Collobert. Recurrent convolutional neural networks for scene parsing. *arXiv preprint arXiv:1306.2795*, 2013. 2

[52] H. Pirsiavash, C. Vondrick, and A. Torralba. Inferring the why in images. *arXiv preprint arXiv:1406.5472*, 2014. 2

[53] M. Ranzato et al. Video (language) modeling: a baseline for generative models of natural videos. *arXiv:1412.6604*, 2014. 2

[54] M. Rodriguez, J. Sivic, I. Laptev, and J.-Y. Audibert. Data-driven crowd analysis in videos. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1235–1242. IEEE, 2011. 2

[55] M. Ryoo. Human activity prediction: Early recognition of ongoing activities from streaming videos. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1036–1043. IEEE, 2011. 2

[56] M. Ryoo, T. J. Fuchs, L. Xia, J. Aggarwal, and L. Matthies. Early recognition of human activities from first-person videos using onset representations. *arXiv preprint arXiv:1406.5309*, 2014. 2

[57] N. Srivastava, E. Mansimov, and R. Salakhutdinov. Unsupervised learning of video representations using lstms. *arXiv:1502.04681*, 2015. 2

[58] A. Surana and K. Srivastava. Bayesian nonparametric inverse reinforcement learning for switched markov decision processes. In *Machine Learning and Applications (ICMLA), 2014 13th International Conference on*, pages 47–54. IEEE, 2014. 2

[59] M. K. C. Tay and C. Laugier. Modelling smooth paths using gaussian processes. In *Field and Service Robotics*, pages 381–390. Springer, 2008. 2

[60] P. Trautman and A. Krause. Unfreezing the robot: Navigation in dense, interacting crowds. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 797–803. IEEE, 2010. 6

[61] P. Trautman, J. Ma, R. M. Murray, and A. Krause. Robot navigation in dense human crowds: the case for cooperation. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 2153–2160. IEEE, 2013. 5

[62] A. Treuille, S. Cooper, and Z. Popović. Continuum crowds. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 1160–1168. ACM, 2006. 2

[63] M. W. Turek, A. Hoogs, and R. Collins. Unsupervised learning of functional categories in video scenes. In *ECCV*, 2010. 2

[64] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. *arXiv preprint arXiv:1411.4555*, 2014. 2

[65] F. Visin, K. Kastner, K. Cho, M. Matteucci, A. Courville, and Y. Bengio. Renet: A recurrent neural network based alternative to convolutional networks. *arXiv preprint arXiv:1505.00393*, 2015. 2

[66] C. Vondrick, H. Pirsiavash, and A. Torralba. Anticipating the future by watching unlabeled video. *arXiv preprint arXiv:1504.08023*, 2015. 2

[67] T.-H. Vu, C. Olsson, I. Laptev, A. Oliva, and J. Sivic. Predicting actions from static scenes. In *Computer Vision–ECCV 2014*, pages 421–436. Springer, 2014. 2

[68] J. Walker, A. Gupta, and M. Hebert. Patch to the future: Unsupervised visual prediction. In *CVPR*, 2014. 2

[69] J. M. Wang, D. J. Fleet, and A. Hertzmann. Gaussian process dynamical models for human motion. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(2):283–298, 2008. 2

[70] T. Xiao, Y. Xu, K. Yang, J. Zhang, Y. Peng, and Z. Zhang. The application of two-level attention models in deep convolutional neural network for fine-grained image classification. *arXiv preprint arXiv:1411.6447*, 2014. 2

[71] D. Xie, S. Todorovic, and S.-C. Zhu. Inferring" dark matter" and" dark energy" from videos. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 2224–2231. IEEE, 2013. 2

[72] K. Xu, J. Ba, R. Kiros, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. *arXiv preprint arXiv:1502.03044*, 2015. 2

[73] K. Yamaguchi, A. C. Berg, L. E. Ortiz, and T. L. Berg. Who are you with and where are you going? In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1345–1352. IEEE, 2011. 2, 3, 5, 6, 7, 8

[74] S. Yi, H. Li, and X. Wang. Understanding pedestrian behaviors from stationary crowd groups. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3488–3496, 2015. 2

[75] D. Yoo, S. Park, J.-Y. Lee, A. Paek, and I. S. Kweon. Attentionnet: Aggregating weak directions for accurate object detection. *arXiv preprint arXiv:1506.07704*, 2015. 2

[76] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. Torr. Conditional random fields as recurrent neural networks. *arXiv preprint arXiv:1502.03240*, 2015. 2

[77] B. Zhou, X. Wang, and X. Tang. Random field topic model for semantic region analysis in crowded scenes from tracklets. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3441–3448. IEEE, 2011. 2

[78] B. D. Ziebart, N. Ratliff, G. Gallagher, C. Mertz, K. Peterson, J. A. Bagnell, M. Hebert, A. K. Dey, and S. Srinivasa. Planning-based prediction for pedestrians. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 3931–3936. IEEE, 2009. 2