Peeking into the Future: Predicting Future Person Activities and Locations in Videos

Junwei Liang[1]* Lu Jiang[2] Juan Carlos Niebles[3,2] Alexander Hauptmann[1] Li Fei-Fei[3,2]
[1]Carnegie Mellon University [2]Google AI [3]Stanford University
{junweil,alex}@cs.cmu.edu, lujiang@google.com, {lifeifei,jniebles}@cs.stanford.edu

Abstract

Deciphering human behaviors to predict their future paths/trajectories and what they would do from videos is important in many applications. Motivated by this idea, this paper studies predicting a pedestrian's future path jointly with future activities. We propose an end-to-end, multi-task learning system utilizing rich visual features about the human behavioral information and interaction with their surroundings. To facilitate the training, the network is learned with two auxiliary tasks of predicting future activities and the location in which the activity will happen. Experimental results demonstrate our state-of-the-art performance over two public benchmarks on future trajectory prediction. Moreover, our method is able to produce meaningful future activity prediction in addition to the path. The result provides the first empirical evidence that a joint modeling of paths and activities benefits future path prediction.

1. Introduction

With the advancement in deep learning, systems now are able to analyze an unprecedented amount of rich visual information from videos to enable applications such as accident avoidance and smart personal assistance. An important analysis is forecasting the future path of pedestrians, called future person path/trajectory prediction. This problem has received increasing attention in the computer vision community [12, 1, 6]. It is regarded as an essential building block in video understanding because looking at the visual information from the past to predict the future is useful in many applications like self-driving cars, socially-aware robots [18], *etc*.

Humans navigate through public spaces often with specific purposes in mind, ranging from simple ones like entering a room to more complicated ones like putting things into a car. Such intention, however, is mostly neglected in existing work. Consider the example in Fig. 1, the person (at the top-right corner) might take different paths depending on their intention, *e.g*., they might take the green path to *transfer object* or the yellow path to *load object into the car*. Inspired by this, this paper is interested in modeling the future path jointly with such intention in videos. We model the intention in terms of a predefined set of 30 activities provided by the NIST such as "loading", "object transfer", *etc*. See Table 4 for the full list.

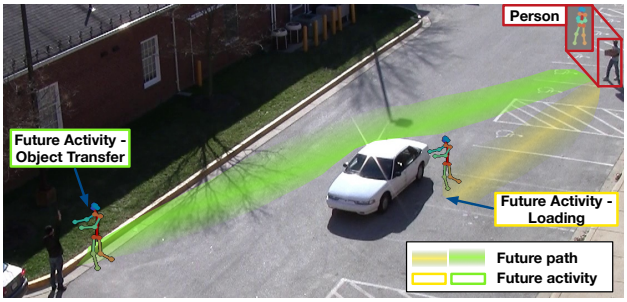

Figure 1. Our goal is to jointly predict a person's future path and activity. The green and yellow line show two possible future trajectories and two possible activities are shown in the green and yellow boxes. Depending on the future activity, the person (top right) may take different paths, *e.g*. the yellow path for "loading" and the green path for "object transfer".

The joint prediction model can have two benefits. First, learning the activity together with the path may benefit the future path prediction. Intuitively, humans are able to read from others' body language to anticipate whether they are going to cross the street or continue walking along the sidewalk. After understanding these behaviors, humans can make better predictions. In the example of Fig. 1, the person is carrying a box, and the man at the bottom left corner is waving at the person. Based on common sense, we may agree that the person will take the green path instead of the yellow path. Second, the joint model advances the capability of understanding not only the future path but also the future activity by taking into account the rich semantic context in videos. This increases in the capabilities of automated video analytics for social good such as real-time accident alerting, self-driving cars, and smart robot assistance. For example. it may also have safety applications

*Work partially done during a part-time research program at Google.

such as anticipating pedestrian movement at traffic intersections or a road robot helping humans transport goods to the trunk of a car. Note that our techniques focus on predicting a few seconds into the future, and should not be useful for non-routine activities.

To this end, we propose a multi-task learning model called *Next* which has prediction modules for learning future paths and future activities simultaneously. As predicting future activity is challenging, we introduce two new techniques to address the issue. First, unlike most of the existing work [12, 1, 6, 25, 20, 29] which oversimplifies a person as a point in space, we encode a person through rich semantic features about visual appearance, body movement and interaction with the surroundings, motivated by the fact that humans derive such predictions by relying on similar visual cues. Second, to facilitate the training, we introduce two auxiliary tasks for future activity prediction, *i.e*. activity label classification and activity location prediction. In the latter task, we design a discretized grid which we call the Manhattan Grid as location prediction target for the system. Experiments show that these auxiliary tasks improve the accuracy of future path prediction.

To the best of our knowledge, our work is the first on joint future path and activity prediction in streaming videos, and more importantly the first to demonstrate such joint modeling can considerably improve the future path prediction. We empirically validate our model on two benchmarks: ETH & UCY [22, 15], and ActEV/VIRAT [21, 3]. Experimental results show that our method outperforms state-of-the-art baselines, achieving the best-published result on two common benchmarks and producing additional prediction about the future activity. To summarize, the contributions of this paper are threefold: **(i)** We conduct a pilot study on joint future path and activity prediction in videos. We are the first to empirically demonstrate the benefit of such joint learning. **(ii)** We propose a multi-task learning framework with new techniques to tackle the challenge of joint future path and activity prediction. **(iii)** Our model achieves the best-published performance on two public benchmarks. Ablation studies are conducted to verify the contribution of the proposed sub-modules.

2. Related Work

Person-person models for trajectory prediction. Person trajectory prediction models try to predict the future path of people, mostly pedestrians. A large body of work learns to predict person path by considering human social interactions and behaviors in crowded scene [30, 32]. Zou *et al*. in [34] learned human behaviors in crowds by imitating a decision-making process. Social-LSTM [1] added social pooling to model nearby pedestrian trajectory patterns. Social-GAN [6] added adversarial training on Social-LSTM to improve performance. Different from these previous work, we represent a person by rich visual features instead of simply considering a person as points in the scene. Meanwhile we use *geometric relation* to explicitly model the person-scene interaction and the person-object relations, which have not been used in previous work.

Person-scene models for trajectory prediction. A number of works focused on learning the effects of the physical scene, *e.g*., people tend to walk on the sidewalk instead of grass. Kitani *et al*. in [12] used Inverse Reinforcement Learning to forecast human trajectory. Xie *et al*. in [29] considered pedestrian as "particles" whose motion dynamics are modeled within the framework of Lagrangian Mechanics. Scene-LSTM [20] divided the static scene into Manhattan Grid and predict pedestrian's location using LSTM. CAR-Net [11] proposed an attention network on top of scene semantic CNN to predict person trajectory. SoPhie [25] combined deep neural network features from scene semantic segmentation model and generative adversarial network (GAN) using attention to model person trajectory. A disparity to [25] is that we explicitly pool scene semantic features around each person at each time instant so that the model can directly learn from such interactions.

Person visual features for trajectory prediction. Some recent works have attempted to predict person path by utilizing individual's visual features instead of considering them as points in the scene. Kooij *et al*. in [13] looked at pedestrian's faces to model their awareness to predict whether they will cross the road using a Dynamic Bayesian Network in dash-cam videos. Yagi *et al*. in [31] used person keypoint features with a convolutional neural network to predict future path in first-person videos. Different from these works, we consider rich visual semantics for future prediction that includes both the person behavior and their interactions with soundings .

Activity prediction/early recognition. Many works have been proposed to anticipate future human actions using Recurrent Neural Network (RNN). [19] and [2] proposed different losses to encourage LSTM to recognize actions early in internet videos. Srivastava *et al*. in [28] utilized unsupervised learning with LSTM to reconstruct and predict video representations. Another line of works is anticipating human activities in robotic vision [14, 9]. Our work differs in that both person behavior and person interaction modeling are used for joint activity and trajectory prediction.

Multiple cues for tracking/group activity recognition. There are previous works that take into account multiple cues in videos for tracking [10, 24] and group activity recognition [5, 27, 26]. Our work differs in that rich visual features and focal attention are used for joint person path and activity prediction. Meanwhile, our work utilizes novel activity location prediction (see Section 3.5) to bridge the two tasks.

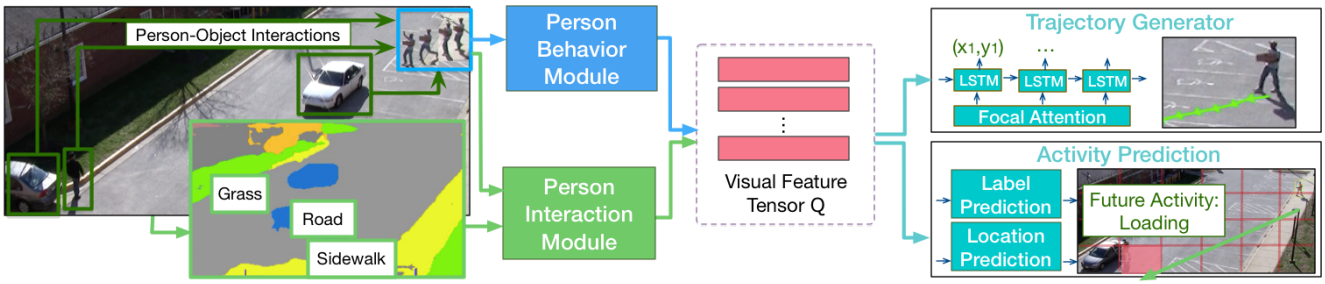

Figure 2. Overview of our model. Given a sequence of frames containing the person for prediction, our model utilizes person behavior module and person interaction module to encode rich visual semantics into a feature tensor. We propose novel person interaction module that takes into account both person-scene and person-object relations for joint activities and locations prediction.

3. Approach

Humans navigate through spaces often with specific purposes in mind. Such purposes may considerably orient the future trajectory/path. This motive us to study the future path prediction jointly with the intention. In this paper, we model the intention in terms of a predefined set of future activities such as "walk", "open_door", "talk", *etc*.

Problem Formulation: Following [1, 6, 25], we assume each scene is first processed to obtain the spatial coordinates of all people at different time instants. Based on the coordinates, we can automatically extract their bounding boxes. Our system observes the bounding box of all the people from time 1 to T_{obs}, and objects if there are any, and predicts their positions (in terms of xy-coordinates) for time T_{obs+1} to T_{pred}, meanwhile estimating the possibilities of future activity labels at time T_{pred}.

3.1. Network Architecture

Fig. 2 shows the overall network architecture of our *Next* model. Unlike most of the existing work [12, 1, 6, 25, 20, 29] which oversimplifies a person as a point in space, our model employs two modules to encode rich visual information about each person's behavior and interaction with the surroundings. In summary, it has the following key components:

Person behavior module extracts visual information from the behavioral sequence of the person.

Person interaction module looks at the interaction between a person and their surroundings.

Trajectory generator summarizes the encoded visual features and predicts the future trajectory by the LSTM decoder with focal attention [16].

Activity prediction utilizes rich visual semantics to predict the future activity label for the person. In addition, we divide the scene into a discretized grid of multiple scales, which we call the Manhattan Grid, to compute classification and regression for robust activity location prediction.

In the rest of this section, we will introduce the above modules and the learning objective in details.

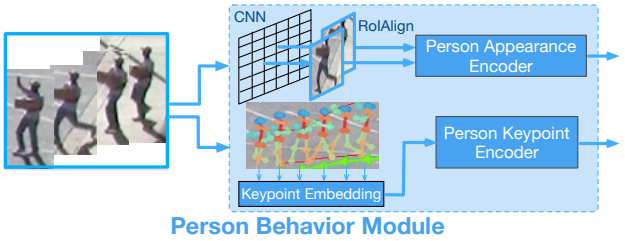

Figure 3. We show the person behavior module given a sequence of person frames. We extract person appearance features and pose features to model the changes of a person's behavior. See Section 3.2.

3.2. Person Behavior Module

This module encodes the visual information about every individual in a scene. As opposed to oversimplifying a person as a point in space, we model the person's the appearance and body movement. To model appearance changes of a person, we utilize a pre-trained object detection model with "RoIAlign" [7] to extract fixed size CNN features for each person bounding box. See Fig. 3. For every person in the scene, we average the feature along the spatial dimensions and feed them into an LSTM encoder. Finally, we obtain a feature representation of $T_{obs} \times d$, where d is the hidden size of the LSTM. These appearance and movement features are commonly used in a wide variety of studies and thus do not introduce new concern on machine learning fairness.

To capture the body movement, we utilize a person keypoint detection model trained on MSCOCO dataset [7] to extract person keypoint information. We apply the linear transformation to embed the keypoint coordinates before feeding into the LSTM encoder. The shape of the encoded feature has the shape of $T_{obs} \times d$.

3.3. Person Interaction Module

This module looks at the interaction between a person and their surroundings, *i.e*. person-scene and person-objects interactions.

Person-scene. To encode the nearby scene of a person, we first use a pre-trained scene segmentation model [4] to extract pixel-level scene semantic classes for each frame. We

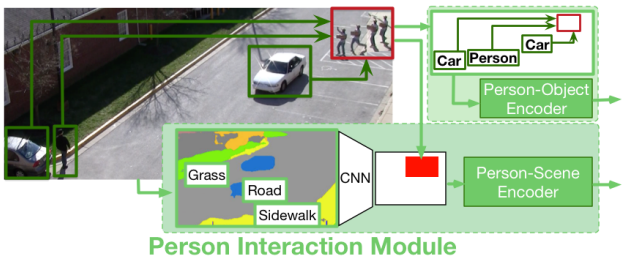

Figure 4. We show the person interaction module which includes person-scene and person-objects modeling. For person-objects modeling, given the person sequence as the red box in the video frame, we extract the spatial relations between the person and other objects at each time instant. For person-scene modeling, surrounding scene semantic features are pooled around the person into the encoder. See Section 3.3.

use totally $N_s = 10$ common scene classes, such as roads, sidewalks, *etc*. The scene semantic features are integers (class indexes) of the size $T_{obs} \times h \times w$, where h, w are the spatial resolution. We first transform the integer tensor into N_s binary masks (one mask for each class), and average along the temporal dimension. This results in N_s real-valued masks, each of the size of $h \times w$. We apply two convolutional layers on the mask feature with a stride of 2 to get the *scene CNN features* in two scales.

Given a person's xy-coordinate, we pool the scene features at the person's current location from the convolution feature map. As the example shown at the bottom of Fig. 4, the red part of the convolution feature is the discretized location of the person at the current time instant. The receptive field of the feature at each time instant, *i.e*. the size of the spatial window around the person which the model looks at, depends on which scale is being pooled from and the convolution kernel size. In our experiments, we set the scale to 1 and the kernel size to 3, which means our model looks at the 3-by-3 surrounding area of the person at each time instant. The person-scene representation for a person is in $\mathbb{R}^{T_{obs} \times C}$, where C is the number of channels in the convolution layer. We feed this into a LSTM encoder in order to capture the temporal information and get the final person-scene features in $\mathbb{R}^{T_{obs} \times d}$.

Person-objects. Unlike previous work [1, 6] which relies on LSTM hidden states to model nearby people, our module explicitly models the *geometric relation* and the *object type* of all the objects/persons in the scene. At any time instant, given the observed box of a person (x_b, y_b, w_b, h_b) and K other objects/persons in the scene $\{(x_k, y_k, w_k, h_k)|k \in [1, K]\}$, we encode the geometric relation into $\mathcal{G} \in \mathbb{R}^{K \times 4}$, the k-th row of which equals to:

$$\mathcal{G}_k = [\log(\frac{|x_b - x_k|}{w_b}), \log(\frac{|y_b - y_k|}{h_b}), \log(\frac{w_k}{w_b}), \log(\frac{h_k}{h_b})] \quad (1)$$

This encoding computes the geometric relation in terms of the geometric distance and the fraction box size. We use a logarithmic function to reflect our observation that human trajectories are more likely to be affected by close-by objects or people. This encoding has been proven effective in object detection [8].

For the object type, we simply use one-hot encoding to get the feature in $\mathbb{R}^{K \times N_o}$, where N_o is the total number of object classes. We then embed the geometric features and the object type features at the current time into d_e-dimensional vectors and feed the embedded features into an LSTM encoder to obtain the final feature of the shape $T_{obs} \times d$.

As shown in the example from Fig. 4, the person-objects feature can capture how far away the person is to the other person and the cars (with respect to their own height). The person-scene feature can capture whether the person is near the sidewalk or grass. We feed this information to the model with the hope of learning things like a person walks more often on the sidewalk than the grass and tends to avoid bumping into cars.

3.4. Trajectory Generation with Focal Attention

As discussed, the above four types of visual features, *i.e*. appearance, body movement, person-scene, and person-objects, are encoded by separate LSTM encoders into the same dimension. Besides, given a person's trajectory output from the last time instant, we extract the trajectory embedding by

$$e_{t-1} = \tanh\{W_e[x_{t-1}, y_{t-1}]\} + b_e \in \mathbb{R}^d, \quad (2)$$

where $[x_{t-1}, y_{t-1}]$ is the trajectory prediction of time $t-1$ and W_e, b_e are learnable parameters. We then feed the embedding e_{t-1} into another LSTM encoder for the trajectory. The hidden states of all encoders are packed into a tensor named $Q \in \mathbb{R}^{M \times T_{obs} \times d}$, where $M = 5$ denotes the total number of features and d is the hidden size of the LSTM.

Following [6], we use an LSTM decoder to directly predict the future trajectory in the xy-coordinate. The hidden state of this decoder is initialized using the last state of the person's trajectory LSTM encoder. At each time instant, the xy-coordinate will be computed from the decoder state $h_t = \text{LSTM}(h_{t-1}, [e_{t-1}, \tilde{q}_t])$ and by a fully connected layer. $\tilde{q}_t$ is an important attended feature vector which summarizes salient cues in the input features Q. We employ an effective focal attention [16] to this end. It was originally proposed to carry out multimodal inference over a sequence of images for visual question answering. The key idea is to project multiple features into a space of correlation, where discriminative features can be easier to capture by the attention mechanism.

To do so, we compute a correlation matrix $S^t \in \mathbb{R}^{M \times T_{obs}}$ at every time instant t, where each entry $S^t_{ij} = h^\top_{t-1} \cdot Q_{ij:}$ is measured using the dot product similarity and $:$ is a slicing operator that extracts all elements from that

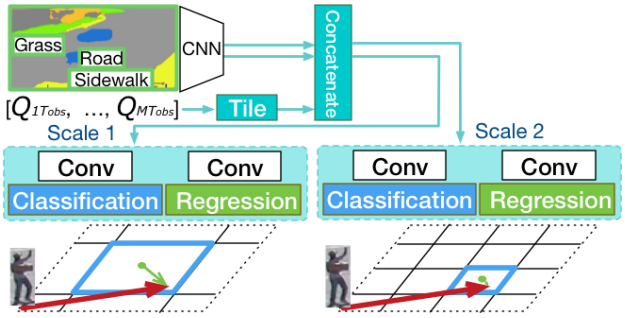

Figure 5. Activity location prediction with classification and regression on the multi-scale Manhattan Grid. See Section 3.5.

dimension. Then we compute two focal attention matrices:

$$A^t = \text{softmax}(\max_{i=1}^{M} S^t_{i:}) \in \mathbb{R}^M \quad (3)$$

$$B^t = [\text{softmax}(S^t_{1:}), \cdots, \text{softmax}(S^t_{M:})] \in \mathbb{R}^{M \times T_{obs}} \quad (4)$$

Then the attended feature vector is given by:

$$\tilde{q}_t = \sum_{j=1}^{M} A^t_j \sum_{k=1}^{T_{obs}} B^t_{jk} Q_{jk:} \in \mathbb{R}^d \quad (5)$$

As shown, the focal attention models the correlation among different features and summarizes them into a low-dimensional attended vector. Section 4 show its benefit in our experiments.

3.5. Activity Prediction

Since the trajectory generation module outputs one location at a time, errors may accumulate across time and the final destination would deviate from the actual location. Using the wrong location for activity prediction may lead to bad accuracy. To counter this disadvantage, we introduce an auxiliary task, *i.e.* activity location prediction, in addition to predicting the future activity label of the person. We describe the two prediction modules in the following.

Activity location prediction with the Manhattan Grid. To bridge the gap between trajectory generation and activity label prediction, we propose an activity location prediction module to predict the final location of where the person will engage in the future activity. The activity location prediction includes two tasks, *location classification* and *location regression*. As illustrated in Fig. 5, we first divide a video frame into a discretized $h \times w$ grid, namely *Manhattan Grid*, and learn to classify the correct grid block and at the same time to regress from the center of that grid block to the actual location. Specifically, the aim for the classification task is to predict the correct grid block in which the final location coordinates reside. After classifying the grid block, the aim for the regression task is to predict the deviation of the grid block center (Blue Dot in the figure) to the final location coordinate (the end of Red Arrow). The reason for adding the regression task are: (i) it will provide more precise locations than just a grid block area; (ii) it is complementary to the trajectory prediction which requires xy-coordinates localization. We repeat this process on the Manhattan Grid of different scales and use separate prediction heads to model them. These prediction heads are trained end-to-end with the rest of the model. Our idea is partially inspired by the region proposal network [23] and our intuition is that similar to object detection problem, we need accurate localization using multi-scale features in a cost-efficient way.

As shown in Fig. 5, we first concatenate the scene CNN features (see Section 3.3) with the last hidden state of the encoders (see Section 3.4). For compatibility, we tile the hidden state $Q_{:T_{obs}:}$ along the height and width dimension resulting in a tensor of the size $M \times d \times w \cdot h$, where $w \cdot h$ is the total number of the grid blocks. The hidden state contains rich information from all encoders and allow gradients flow smoothly through from prediction to feature encoders.

The concatenated features are fed into two separate convolution layers for classification and regression. The convolution output for grid classification $\mathsf{cls}_{grid} \in \mathbb{R}^{w \cdot h \times 1}$ indicates the probability of each grid block being the correct destination. In comparison, the convolution output for grid regression $\mathsf{rg}_{grid} \in \mathbb{R}^{w \cdot h \times 2}$ denotes the deviation, in the xy-coordinates, between the final destination and every grid block center. A row of rg_{grid} represents the difference to a grid block, calculated from $[x_t - x_{ci}, y_t - y_{ci}]$ where (x_t, y_t) denotes the predicted location and (x_{ci}, y_{ci}) is the center of the i-th grid block. The ground truth for the grid regression can be computed in a similar way. During training, only the correct grid block receives gradients for regression. Recent work [20] also incorporates the grid for location prediction. Our model differs in that we link grid locations to scene semantics, and use a classification layer and a regression layer together to make more robust predictions.

Activity label prediction. Given the encoded visual observation sequence, the activity label prediction module predicts the future activity at time instant T_{pred}. We compute the future N_a activity probabilities using the concatenated last hidden states of the encoders:

$$\mathsf{cls}_{act} = \text{softmax}(W_a \cdot [Q_{1T_{obs}:}, \cdots, Q_{MT_{obs}:}]) \quad (6)$$

where W_a is a learnable weight. The future activity of a person could be multi-class, *e.g.* a person could be "walking" and "carrying" at the same time.

3.6. Training

The entire network is trained end-to-end by minimizing a multi-task objective. The primary loss is the common L_2 loss between the predicted future trajectories and the ground-truth trajectories [20, 6, 25]. The loss is summed into L_{xy} over all persons from T_{obs+1} to T_{pred}.

The second category of loss is the activiy location classification and regression loss discussed in Section 3.5. We have $L_{grid\_cls} = \sum_{i=1}^{N} \text{ce}(\mathsf{cls}^i_{grid}, \mathsf{cls}^{*i}_{grid})$, where cls^{*i}_{grid} is the ground-truth final location grid block ID for the i^{th} training trajectory. Likewise $L_{grid\_reg} =$

$\sum_{i=1}^{N} \text{smooth}_{L_1}(\text{rg}^i_{grid}, \text{rg}^{*i}_{grid})$ and rg^{*i}_{grid} is the ground-truth difference to the correct grid block center. This loss is designed to bridge the gap between the trajectory generation task and activity label prediction task.

The third loss is for activity label prediction. We employ the cross-entropy loss: $L_{act} = \sum_{i=1}^{N} \text{ce}(\text{cls}^i_{act}, \text{cls}^{*i}_{act})$. The final loss is then calculated from:

$$L = L_{xy} + \lambda(L_{grid\_cls} + L_{grid\_reg}) + L_{act} \quad (7)$$

We use a balance controller $\lambda = 0.1$ for location destination prediction to offset their higher loss values during training.

4. Experiments

We evaluate the proposed *Next* model on two common benchmarks for future path prediction: ETH [22] and UCY [15], and ActEV/VIRAT [3, 21]. We demonstrate that our model performs favorably against the state-of-the-art models on this challenging task. The source code and models will be made available to the public.

4.1. ActEV/VIRAT

Dataset & Setups. ActEV/VIRAT [3] is a public dataset released by NIST in 2018 for activity detection research in streaming video (https://actev.nist.gov/). This dataset is an improved version of VIRAT [21], with more videos and annotations. It includes 455 videos at 30 fps from 12 scenes, more than 12 hours of recordings. Most of the videos have a high resolution of 1920x1080. We use the official training set for training and the official validation set for testing.

Following [1, 6, 25], the models observe 3.2 seconds (8 frames) of every person and predict the future 4.8 seconds (12 frames) of person trajectory, we downsample the videos to 2.5 fps and extract person trajectories using the code released in [6]. Since we do not have the homographic matrix, we use the pixel values for the trajectory coordinates as it is done in [31].

Evaluation Metrics. Following prior work [1, 6, 25], we use two error metrics for person trajectory prediction:

i) *Average Displacement Error* (ADE): The average Euclidean distance between the ground truth coordinates and the prediction coordinates over all time instants,

$$\text{ADE} = \frac{\sum_{i=1}^{N} \sum_{t=1}^{T_{pred}} \|\tilde{Y}^i_t - Y^i_t\|_2}{N * T_{pred}} \quad (8)$$

ii) *Final Displacement Error* (FDE): The euclidean distance between the predicted points and the ground truth point at the final prediction time instant T_{pred},

$$\text{FDE} = \frac{\sum_{i=1}^{N} \|\tilde{Y}^i_{T_{pred}} - Y^i_{T_{pred}}\|_2}{N} \quad (9)$$

The errors are measured in the pixel space on ActEV/VIRAT whereas in meters on ETH and UCY. For future activity prediction, we use mean average precision (mAP).

	Method	ADE	FDE	move_ADE	move_FDE
Single Model	Linear	32.19	60.92	42.82	80.18
	LSTM	23.98	44.97	30.55	56.25
	Social LSTM	23.10	44.27	28.59	53.75
	SGAN-PV	30.51	60.90	37.65	73.01
	SGAN-V	30.48	62.17	35.41	68.77
	Ours	**17.99**	**37.24**	**20.34**	**42.54**
20 Outputs	SGAN-PV-20	23.11	41.81	29.80	53.04
	SGAN-V-20	21.16	38.05	26.97	47.57
	Ours-20	**16.00**	**32.99**	**17.97**	**37.28**

Table 1. Comparison to baseline methods on the ActEV/VIRAT validation set. Top uses the single model output. Bottom uses 20 outputs. Numbers denote errors thus lower are better.

Baseline methods. We compare our method with the two simple baselines and two recent methods: ***Linear*** is a single layer model that predicts the next coordinates using a linear regressor based on the previous input point. ***LSTM*** is a simple LSTM encoder-decoder model with coordinates input only. ***Social LSTM*** [1]: We train the social LSTM model to directly predict trajectory coordinates instead of Gaussian parameters. ***SGAN*** [6]: We train two model variants (PV & V) detailed in the paper using the released code from Social-GAN [6] (https://github.com/agrimgupta92/sgan/).

Aside from using a single model at test time, Gupta *et al*. [6] also used 20 model outputs per frame and selected the best prediction to count towards the final performance. Following the practice, we train 20 identical models using random initializations and report the same evaluation results, which are marked "20 outputs" in Table 6.

Implementation Details. We use LSTM cell for both the encoder and decoder. The embedding size d_e is set to 128, and the hidden sizes d of encoder and decoder are both 256. Ground truth bounding boxes of persons and objects are used during the observation period (from time 1 to T_{obs}). For person keypoint features, we utilize the pre-trained pose estimator from [7] to extract 17 joints for each ground truth person box. For person appearance feature, we utilize the pre-trained object detection model FPN [17] to extract appearance features from person bounding boxes. The scene semantic segmentation features are resized to (64, 36) and the scene convolution layers are set to have a kernel size of 3, a stride of 2 and the channel dimension is 64. We resize all videos to 1920x1080 and utilize two grid scales, 32x18 and 16x9. The activation function is $tanh$ if not stated otherwise and we do not use any normalization. For training, we use Adadelta optimizer [33] with an initial learning rate of 0.1 and the dropout value is 0.3. We use gradient clipping of 10 and weight decay of 0.0001. For Social LSTM, the neighbor is set to 256 pixels as in [31]. All baselines use the same embedding size and hidden size as our model, therefore all encoder-decoder models have about the same numbers of parameters. Other hyper-parameters we use for the baselines follow the ones in [6].

Main Results. Table 6 lists the testing error, where the top

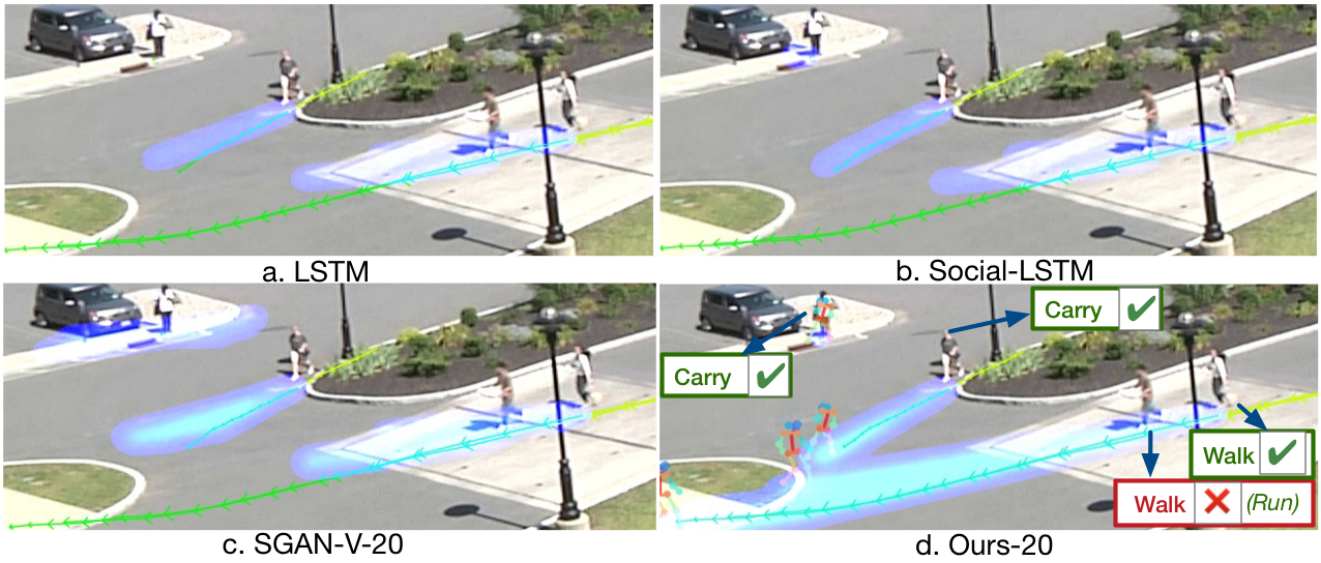

Figure 6. (Better viewed in color.) Qualitative comparison between our method and the baselines. Yellow path is the observable trajectory and Green path is the ground truth trajectory during the prediction period. Predictions are shown as Blue heatmaps. Our model also predicts the future activity, which is shown in the text and with the person pose template.

part is the error of a single model output and the bottom shows the best result of 20 model outputs. The "ADE" and "FDE" columns summarize the error over all trajectories, and the last two columns further detail the subset trajectories of moving activities ("walk", "run", and "ride_bike"). We report the mean performance of 20 runs of our single model at Row 7. The standard deviation on "ADE" metric is 0.043. Full numbers can be found in supplemental material. As we see, our method performs favorably against other methods, especially in predicting the trajectories of moving activities. For example, our model outperforms Social-LSTM and Social-GAN by a large margin of 10 points in terms of the "move_FDE" metric. The results demonstrate the efficacy of the proposed model and its state-of-the-art performance on future trajectory prediction.

Qualitative analysis. We visualize and compare our model outputs and the baselines in Fig. 6. In each graph the yellow trajectories are the observable sequences of each person and the green trajectories are the ground truth future trajectories. The predicted trajectories are shown in the blue heatmap. To better visualize the predicted future activities of our method, we plot the person keypoint template for each predicted activity at the end of the predicted trajectory. As we see, our method outputs more accurate trajectories for each person, especially for the two persons on the right that were about to accelerate their movement. Our method is also able to predict most of the activities correct except one (walk versus run). Our model successfully predicts the activity "carry" and the static trajectory of the person near the car, while in Fig 6(c), SGAN predicts several moving trajectories in different directions.

Method	ADE ↓	FDE ↓	Act mAP ↑
Our full model	17.91	37.11	0.192
No p-behavior	18.99	39.82	0.139
No p-interaction	18.83	39.35	0.163
No focal attention	19.93	42.08	0.144
No act label loss	19.48	41.45	-
No act location loss	19.07	39.91	0.152
No multi-task	20.37	42.79	-

Table 2. Multi-task performance & ablation experiments.

We further provide a qualitative analysis of our model predictions. (i) Successful cases: In Fig 8(a) and 8(b), both the trajectory prediction and future activity prediction are correct. (ii) Imperfect case: In Fig 8(c), although the trajectory prediction is mostly correct, our model predicts that the person is going to open the door of the car, given the observation that he is walking towards the side of the car. (iii) Failed case: In Fig 8(d), our model fails to capture the subtle interactions between the two persons and predicts that they will go separate ways, while in fact they are going to stop and talk to each other.

4.2. Ablation Model

In Table 2, we systematically evaluate our method through a series of ablation experiments, where "ADE" and "FDE" denotes the errors thus lower are better. "Act" is the mean Average Precision (mAP) of the activity label prediction over 30 activities and higher are better.

Efficacy of rich visual features. We investigate the feature contribution of person behavior and person interactions by separately ablating them. As shown in the first three rows in Table 2, both features are important to trajectory prediction while person behavior features are more essential for activity prediction.

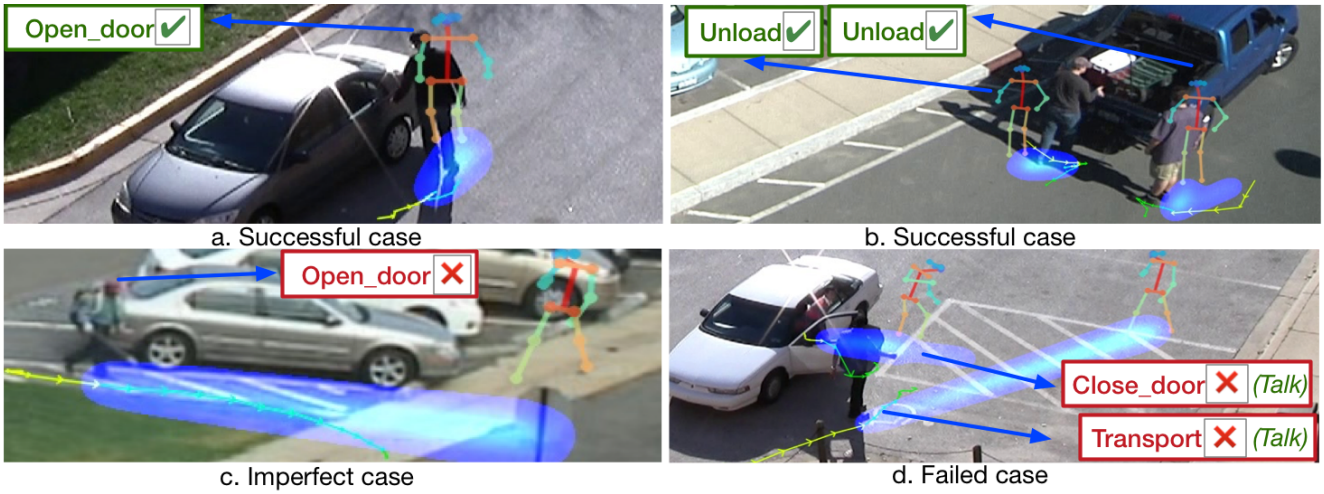

Figure 7. (Better viewed in color.) Qualitative analysis of our model. Please refer to Fig. 6 for legends.

	Method	ETH	HOTEL	UNIV *	ZARA1	ZARA2	AVG
Single Model	Linear	1.33 / 2.94	0.39 / 0.72	0.82 / 1.59	0.62 / 1.21	0.77 / 1.48	0.79 / 1.59
	LSTM	1.09 / 2.41	0.86 / 1.91	**0.61 / 1.31**	**0.41 / 0.88**	0.52 / 1.11	0.70 / 1.52
	Alahi *et al.* [1]	1.09 / 2.35	0.79 / 1.76	0.67 / 1.40	0.47 / 1.00	0.56 / 1.17	0.72 / 1.54
	Ours-single-model	**0.88 / 1.98**	**0.36 / 0.74**	0.62 / 1.32	0.42 / 0.90	**0.34 / 0.75**	**0.52 / 1.14**
20 Outputs	Gupta *et al.* [6](V)	0.81 / 1.52	0.72 / 1.61	0.60 / 1.26	0.34 / 0.69	0.42 / 0.84	0.58 / 1.18
	Gupta *et al.* [6](PV)	0.87 / 1.62	0.67 / 1.37	0.76 / 1.52	0.35 / 0.68	0.42 / 0.84	0.61 / 1.21
	Sadeghian *et al.* [25]	**0.70 / 1.43**	0.76 / 1.67	**0.54 / 1.24**	**0.30 / 0.63**	0.38 / 0.78	0.54 / 1.15
	Ours-20	0.73 / 1.65	**0.30 / 0.59**	0.60 / 1.27	0.38 / 0.81	**0.31 / 0.68**	**0.46 / 1.00**

Table 3. Comparison of different methods on ETH (Column 3 and 4) and UCY datasets (Column 5-7). * We use a smaller test set on UNIV since 1 video is unable to download.

Effect of focal attention. In the fourth row of Table 2, we replace focal attention in Eq. (5) with a simple average of the last hidden states from all encoders. Both trajectory and activity prediction hurt as a result.

Impact of multi-task learning. In the last three rows of Table 2, we remove the additional tasks of predicting the activity label or the activity location or both to see the impact of multi-task learning. Results show the benefit of our multi-task learning method.

Using observed activities as predictions. Since we are predicting activities in the not so distant future, a system may perform well enough if it just outputs the current activity labels as the future prediction. We train an identical model to detect the activity labels at time T_{obs} as the future prediction outputs, which results in a performance of 0.155 mAP for activity prediction and 18.27 ADE for trajectory prediction. Such a significant performance drop (0.192 vs. 0.155) suggests that activity prediction even for 4.8 seconds into the future is not a trivial task.

4.3. ETH & UCY

Dataset. ETH [22] and UCY [15] are common datasets for person trajectory prediction benchmark [1, 6, 20, 25]. Same as previous work [1, 6, 20, 25], we report performance by averaging over both datasets. We use the same data processing method and settings detailed in [6]. This benchmark includes videos from five scenes: ETH, HOTEL, UNIV, ZARA1 and ZARA2. Leave-one-scene-out data split is used and we evaluate our model on 5 sets of data. We follow the same testing scenario and baselines as in the previous section. We have also cited the latest state-of-the-art results from [25]. Due to 1 video cannot be downloaded, we use a smaller test set for UNIV and a smaller training set across all splits. The other 4 test sub-datasets are the same as in [6] so the numbers are comparable.

Since there is no activity annotation, we do not use activity label prediction module in our model. Since the annotation is only a point for each person and the human scale in each video doesn't change much, we apply a fixed size expansion from points for each video to get the person bounding box annotation for feature pooling. We do not use any other bounding box. We don't use any additional annotation compared to baselines to ensure a fair comparison.

Implementation Details. We do not use person keypoint feature. Final location loss and trajectory L2 loss are used. Unlike [25], we don't utilize any data augmentation. We train our model for 40 epochs with the adadelta optimizer. Other hyper-parameters are the same as in Section 4.1.

Results & Analysis. Experiments are shown in Table 3. Our model outperforms other methods in both evaluations,

where we obtain the best-published single model on ETH and best average performance on the ETH & UCY benchmark. As shown in the table, our model performs much better on HOTEL and ZARA2. The average movement at each time-instant in these two scenes are 0.18 and 0.22, respectively, much lower than others: 0.389 (ZARA1), 0.460 (ETH), 0.258 (UNIV). Recall that the leave-one-scene-out data split is used in training. The results suggest other methods are more likely to overfit to the trajectories of large movements, e.g. Social-GAN [6] often "over-shoot" when predicting the future trajectories. In comparison, our method uses attention to find the "right" visual signal and show better performance for trajectories of small movements on HOTEL and ZARA2 while still being competitive for trajectories of large movements.

5. Conclusion

In this paper, we have presented a new neural network model for predicting human trajectory and future activity simultaneously. We first encode a person through rich visual features capturing human behaviors and interactions with their surroundings. Then we add two auxiliary tasks of predicting the activity labels as well as the activity locations to facilitate the joint training process. We refer to the resulting model as *Next*. We showed the efficacy of our model on both popular and recent large-scale video benchmarks on person trajectory prediction. In addition, we quantitatively and qualitatively demonstrated that our *Next* model successfully predicts meaningful future activities.

Our research goal is to promote human safety in applications such as robotics or autonomous driving. We experiment on the public benchmark ActEV, the primary driver of which is to support public safety and traffic monitoring and management by automatic activity detection in streaming video[1]. Our approach works on a predefined set of 30 activities provided by the NIST, such as "loading", "object transfer". See Table 4 for the full list. Our system may not work beyond these predefined activities.

Future research into activity and path prediction may implicate ethical issues around privacy, safety and fairness and ought to be considered carefully before being used in real-world applications. Our method for predicting trajectory and activity has not been tested for different populations of people. As such, it is important to further evaluate these issues before employing the model in situations that may differentially impact people.

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*, 2016. 1, 2, 3, 4, 6, 8

[2] M. S. Aliakbarian, F. Saleh, M. Salzmann, B. Fernando, L. Petersson, and L. Andersson. Encouraging lstms to anticipate actions very early. 2017. 2

[3] G. Awad, A. Butt, K. Curtis, J. Fiscus, A. Godil, A. F. Smeaton, Y. Graham, W. Kraaij, G. Qunot, J. Magalhaes, D. Semedo, and S. Blasi. Trecvid 2018: Benchmarking video activity detection, video captioning and matching, video storytelling linking and video search. In *TRECVID*, 2018. 2, 6

[4] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018. 3

[5] W. Choi and S. Savarese. Understanding collective activitiesof people from videos. *IEEE transactions on pattern analysis and machine intelligence*, 36(6):1242–1257, 2014. 2

[6] A. Gupta, J. Johnson, S. Savarese, Li Fei-Fei, and A. Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *CVPR*, 2018. 1, 2, 3, 4, 5, 6, 8, 9, 12

[7] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *ICCV*, 2017. 3, 6

[8] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei. Relation networks for object detection. In *CVPR*, 2018. 4

[9] A. Jain, H. S. Koppula, B. Raghavan, S. Soh, and A. Saxena. Car that knows before you do: Anticipating maneuvers via learning temporal driving models. In *CVPR*, 2015. 2

[10] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena. Structural-rnn: Deep learning on spatio-temporal graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5308–5317, 2016. 2

[11] N. Jaipuria, G. Habibi, and J. P. How. A transferable pedestrian motion prediction model for intersections with different geometries. *arXiv preprint arXiv:1806.09444*, 2018. 2

[12] K. M. Kitani, B. D. Ziebart, J. A. Bagnell, and M. Hebert. Activity forecasting. In *ECCV*, 2012. 1, 2, 3

[13] J. F. P. Kooij, N. Schneider, F. Flohr, and D. M. Gavrila. Context-based pedestrian path prediction. In *ECCV*, 2014. 2

[14] H. S. Koppula and A. Saxena. Anticipating human activities using object affordances for reactive robotic response. *IEEE transactions on pattern analysis and machine intelligence*, 38(1):14–29, 2016. 2

[1] https://actev.nist.gov/1B-Evaluation

[15] A. Lerner, Y. Chrysanthou, and D. Lischinski. Crowds by example. In *Computer Graphics Forum*, pages 655–664. Wiley Online Library, 2007. 2, 6, 8

[16] J. Liang, L. Jiang, L. Cao, L.-J. Li, and A. Hauptmann. Focal visual-text attention for visual question answering. In *CVPR*, 2018. 3, 4

[17] T.-Y. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 6

[18] M. Luber, J. A. Stork, G. D. Tipaldi, and K. O. Arras. People tracking with human motion predictions from social forces. In *ICRA*, 2010. 1

[19] S. Ma, L. Sigal, and S. Sclaroff. Learning activity progression in lstms for activity detection and early detection. In *CVPR*, 2016. 2

[20] H. Manh and G. Alaghband. Scene-lstm: A model for human trajectory prediction. *arXiv preprint arXiv:1808.04018*, 2018. 2, 3, 5, 8

[21] S. Oh, A. Hoogs, A. Perera, N. Cuntoor, C.-C. Chen, J. T. Lee, S. Mukherjee, J. Aggarwal, H. Lee, L. Davis, et al. A large-scale benchmark dataset for event recognition in surveillance video. In *CVPR*, 2011. 2, 6

[22] S. Pellegrini, A. Ess, and L. Van Gool. Improving data association by joint modeling of pedestrian trajectories and groupings. In *ECCV*, 2012. 2, 6, 8

[23] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 5

[24] A. Sadeghian, A. Alahi, and S. Savarese. Tracking the untrackable: Learning to track multiple cues with long-term dependencies. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 300–311, 2017. 2

[25] 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. 2, 3, 5, 6, 8, 12

[26] T. Shu, S. Todorovic, and S.-C. Zhu. Cern: confidence-energy recurrent network for group activity recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, 2017. 2

[27] T. Shu, D. Xie, B. Rothrock, S. Todorovic, and S. Chun Zhu. Joint inference of groups, events and human roles in aerial videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4576–4584, 2015. 2

[28] N. Srivastava, E. Mansimov, and R. Salakhudinov. Unsupervised learning of video representations using lstms. In *ICML*, 2015. 2

[29] D. Xie, T. Shu, S. Todorovic, and S.-C. Zhu. Learning and inferring dark matter and predicting human intents and trajectories in videos. *IEEE transactions on pattern analysis and machine intelligence*, 40(7):1639–1652, 2018. 2, 3

[30] Y. Xu, Z. Piao, and S. Gao. Encoding crowd interaction with deep neural network for pedestrian trajectory prediction. In *CVPR*, 2018. 2

[31] T. Yagi, K. Mangalam, R. Yonetani, and Y. Sato. Future person localization in first-person videos. In *CVPR*, 2018. 2, 6

[32] S. Yi, H. Li, and X. Wang. Pedestrian behavior understanding and prediction with deep neural networks. In *ECCV*, 2016. 2

[33] M. D. Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012. 6

[34] H. Zou, H. Su, S. Song, and J. Zhu. Understanding human behaviors in crowds by imitating the decision-making process. *arXiv preprint arXiv:1801.08391*, 2018. 2

Appendix

In this appendix, we present more details and analysis for our experiments on the ActEV/VIRAT and ETH & UCY Benchmarks. We also provide statistical comparisons of the two datasets.

5.1. ActEV/VIRAT Details

5.1.1 Object & Activity Class

We show the object classes we used for our person interaction module and the activity classes for our activity prediction module in Table 4. Detailed class definition can be found on https://actev.nist.gov/.

	Classes
Object	Bike, Construction_Barrier, Construction_Vehicle, Door, Dumpster, Parking_Meter, Person, Prop, Push_Pulled_Object, Vehicle
Activity	Carry, Close_Door, Close_Trunk, Crouch, Enter, Exit, Gesture, Interaction, Load, Object_Transfer, Open_Door, Open_Trunk, PickUp, PickUp_Person, Pull, Push, Ride_Bike, Run, SetDown, Sit, Stand, Talk, Talk_phone, Texting, Touch, Transport, Unload, Use_tool, Walk

Table 4. Object & Activity Classes.

5.1.2 Trajectory Type

In ActEV/VIRAT dataset, there are two distinctive types of trajectory: relatively static and the moving ones. We label the person trajectory as moving if at time T_{obs} there is an activity label of one of the following: "Walk", "Run", "Ride_Bike", otherwise we label it as static trajectory. Table 5.1.4 shows the mean displacement in pixels between the last observed point and the prediction trajectory points. As we see, there is a large difference between the two types of trajectory.

	move_traj	static_traj
Average Displacement (train)	69.18	7.57
Final Displacement (train)	124.79	14.63
num% (train)	48.8%	51.2%
Average Displacement (test)	75.78	12.01
Final Displacement (test)	137.21	23.11
num% (test)	61.9%	38.1%

Table 5. Trajectory statistics for different trajectory class in ActEV dataset (on the training set).

5.1.3 Nearest Neighbor Experiment

Since the ActEV/VIRAT experiment is not camera-independent, we conduct a nearest neighbor experiment. Specifically, for each observed sequence in the test set, we use the nearest sequence in the training set as future predictions. As shown in Table 6, it is non-trivial to predict human trajectory as people navigate differently even in the same scene. Please refer to the paper for evaluation metrics.

5.1.4 Single Model Experiment

We train 20 identical *Next* models with different initialization for the single output experiment. We show the mean and standard deviation numbers in Table 6.

5.1.5 Single Feature Ablation Experiments

We experiment with ablating person-object, person-scene, person keypoint and person appearance feature, as shown in Table 7.

5.1.6 More Qualitative Analysis

We show more qualitative analysis in Fig. 8. In each graph the yellow trajectories are the observable sequences of each person and the green trajectories are the ground truth future trajectories. The predicted trajectories are shown in the blue heatmap. To better visualize the predicted future activities of our method, we plot the person keypoint template for each predicted activity at the end of the predicted trajectory.

Successful cases: In Fig 8(a), Fig 8(b), Fig 8(c) and Fig 8(d), both the trajectory prediction and future activity prediction are correct. In Fig 8(d), our model successfully predicts the two persons at the bottom is going to walk past the car and also one of them is going to gesture at the other people by the trunk of the car.

Imperfect cases: In Fig 8(e) and Fig 8(f), although the activity predictions are correct, our model predicts the wrong trajectories. In Fig 8(e), our model fails to predict that the person is going to the other direction. In Fig 8(e), our model fails to predict that the person near the car is going to open the front door instead of the back door.

Failed cases: In Fig 8(g) and Fig 8(h), our model fails to predict both trajectories and activities. In Fig 8(h), the person on the bike is going to turn to avoid the incoming car while our model predicts a straight direction.

5.1.7 Comparing ActEV/VIRAT to ETH & UCY Benchmark

We compare the ActEV/VIRAT dataset and the ETH & UCY trajectory benchmark in Table 5.1.7. As we see, the ActEV/VIRAT dataset is much larger compared to the other benchmark. Also, the ActEV/VIRAT includes bounding

Metric	Nearest Neighbor	Our-Single-Model
ADE	40.04	17.99±0.043
FDE	73.69	37.24±0.102
move_ADE	39.52	20.34±0.059
move_FDE	72.67	42.54±0.146

Table 6. Our single model experiment on the ActEV/VIRAT benchmark.

Method	ADE ↓	FDE ↓	Act mAP ↑
Our full model	17.91	37.11	0.192
No p-object	18.17	37.13	0.198
No p-scene	18.18	37.75	0.206
No p-keypoint	18.25	37.96	0.190
No p-appearance	18.20	37.79	0.154

Table 7. Single feature ablation experiments on the ActEV/VIRAT benchmark.

	ActEV	ETH, UCY
#Scene	5	4
Dataset Length	4 hours 22 minutes	38 minutes
Resolutions	1920x1080, 1280x720	640x480, 720x576
FPS	30	25
Annotation FPS	30	2.5
#Traj	84600	19359, (10039 in Univ)
Annotations	Person+object bounding boxes, activities	Person coordinates

Table 8. Comparison to commonly used person trajectory benchmark datasets.

box and activity annotations that could be used for multi-task learning. The ActEV/VIRAT is inherently different from the crow dataset since it includes diverse annotation of human activities rather than just passers-by, which makes trajectory prediction more purpose-oriented. We show the trajectory numbers after processing based on the setting of eight-second-length sequences. Note that in the public benchmark it is unbalanced since there is one crowded scene called "University" that contains over half of the trajectories in 4 scenes.

5.2. ETH & UCY Details

5.2.1 Dataset Difference Compared to SGAN

The dataset we use is slightly different from the one in [6], as some original videos are unavailable even though their trajectory annotations are provided. Specifically, two videos from UNIV scene, "students001", "uni_examples", and one video from ZARA3, "crowds_zara03", which is used in training for all corresponding splits in [6], cannot be downloaded from the dataset website. Therefore, the test set for UNIV we use is smaller than previous methods [6, 25] while the training set we use is about 34% smaller. Test sets for other 4 splits are the same therefore the numbers are comparable.

5.2.2 Pre-Processing Details

Since the annotation is only a point for each person and the human scale in each video doesn't change much, we apply a fixed size expansion from the annotated points for each video to get the person bounding box annotation for appearance and person-scene feature pooling. Specifically, we use a bounding box size of 50 pixels by 80 pixels with the original annotation point putting at the center of the bottom line. All videos are resized to 720x576. The spatial dimension of the scene semantic segmentation feature is (64, 51) and two grid scales are used: (32, 26), (16, 13).

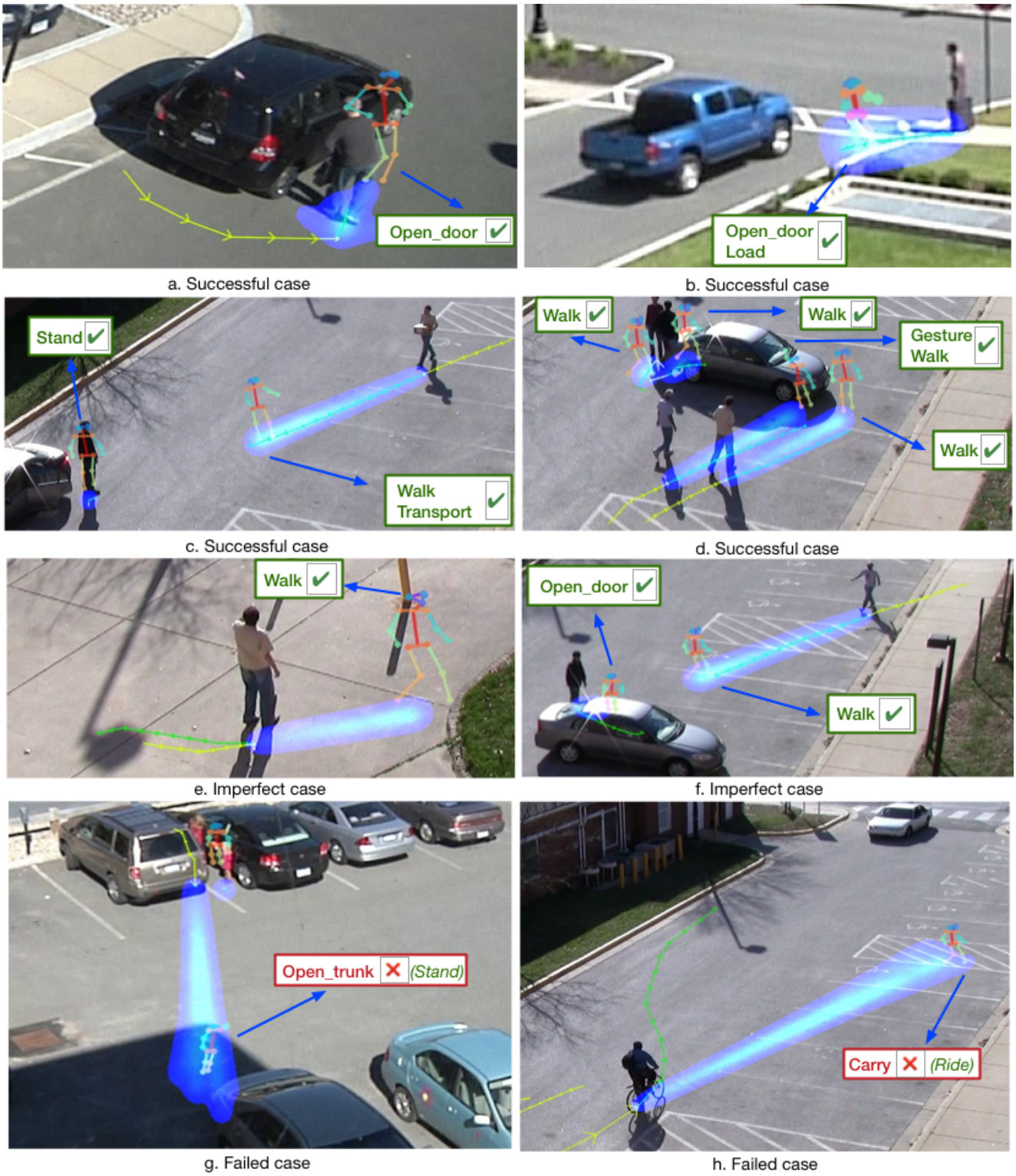

Figure 8. (Better viewed in color.) Qualitative analysis of our model.