Group Split and Merge Prediction with 3D Convolutional Networks

Allan Wang[1], and Aaron Steinfeld[1]

***Abstract*—Mobile robots in crowds often have limited navigation capability due to insufficient evaluation of pedestrian behavior. We strengthen this capability by predicting splits and merges in multi-person groups. Successful predictions should lead to more efficient planning while also increasing human acceptance of robot behavior. We take a novel approach by formulating this as a video prediction problem, where group splits or merges are predicted given a history of geometric social group shape transformations. We take inspiration from the success of 3D convolution models for video-related tasks. By treating the temporal dimension as a spatial dimension, a modified C3D model successfully captures the temporal features required to perform the prediction task. We demonstrate performance on several datasets and analyze transfer ability to other settings. While current approaches for tracking human motion are not explicitly designed for this task, our approach performs significantly better at predicting the occurrence of splits and merges. We also draw human interpretations from the model's learned features.**

***Index Terms*—Human-Centered Robotics, Human Detection and Tracking**

I. Introduction

ONE goal of human-robot interaction (HRI) is to enable trust and acceptance of robots in public settings. A key capability in support of this goal is *social navigation* when a robot is maneuvering among pedestrians. Traditionally, mobile robots have poor navigation skills in crowded areas, which can lead to the *freezing robot problem* [38] or result in displays of confusion or unpredictability. Humans may perceive these behavior as rude or dangerous [20], [24].

In this work, we try to predict social group splits and merges. Our analysis of social groups is inspired by a human perceptual process known as Gestalt [7], where humans mentally group individuals moving together at a similar direction and pace into a single unit. This has been used successfully for robot perception of human crowds [3]. Likewise, past work suggests that individuals within such a group may have social ties among them [23] making it inappropriate for a robot to plan a path that cuts through a group. Predicting splits and merges is important also because it may lead to more efficient robot navigation. Knowing when and where a split or merge will occur allows path planning towards a split point or preemptive avoidance of merging groups. This planning consideration can form more natural navigation paths, increasing trust and acceptance from humans.

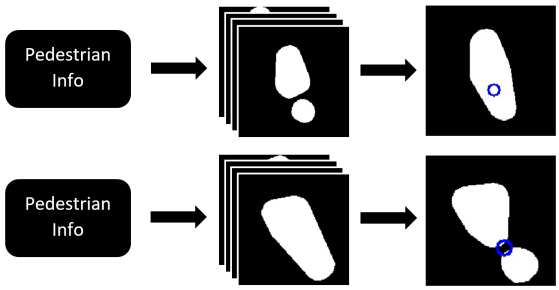

Fig. 1. Predicting social group splits and merges offer navigation benefits for mobile robots. We first generate social group shape sequences from available pedestrian information. Then we use our model to predict splits and merges. The blue circles represent the locations of splits and merges.

We formulate our group dynamics analysis as a video event prediction and localization problem. As shown in Figure 1, we attempt to predict if a split or merge will happen given a history of social group shape images. If an event occurs, we also attempt to predict where it will happen. Deep learning techniques have shown great potential recently for many real-world tasks. Acting as complex function approximators [18], deep learning models can approximate implicit functions not yet sufficiently studied. Predicting splits and merges based on temporal and spatial features within social group shapes is one such implicit function.

Our problem is different from many other video analysis tasks. First, the inputs to our model are textureless binary image videos, as shown in Figure 1. The model needs to rely on the subtle temporal features in the transformations of group shapes because it is impossible to predict splits and merges from a single image, unlike action recognition tasks [13], [14], [31]. Second, our model uses social group shapes, because group shapes can be generated from noisy, raw sensor inputs such as point clouds and, depending on the formulation of group shapes, can be used to account for occluded pedestrians [3]. This makes our task different from tasks that require precise tracking of pedestrians such as the trajectory prediction tasks [1], [9], [28], [40]. Third, our network tries to predict an event in the future. In other words, the defining features that signal the event are not available to the network, as opposed to tracking tasks [12], [27], [46].

In summary, we utilize a 3D convolutional network to predict the occurrence and location of group splits and merges, given a sequence of video frames representing the evolution

Manuscript received: September, 10, 2019; Revised November, 30, 2019; Accepted January, 3, 2020.

This paper was recommended for publication by Editor Tamim Asfour upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by grant (IIS-1734361) from the National Science Foundation.

[1]A. Wang and A. Steinfeld are with the Robotics Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA 15213. {allanwan@cs., steinfeld@}cmu.edu

Digital Object Identifier (DOI): see top of this page.

of social group shapes. Our contributions include:

1) Definition of a new formulation for the split-merge problem, which includes example pipelines to generate social group shape images;
2) A slightly modified C3D architecture [37] with demonstrated effectiveness;
3) Concrete evidence that shows the effectiveness of 3D Convolutions due to our task uniqueness;
4) Discussions of human behavioral implications from our model's learned features.

II. Related Work

A. Social Interaction Between Pedestrians

Traditionally, researchers have tried to understand pedestrian social behavior from a definitive standpoint by drawing inspirations from topics in physics, such as fluid dynamics [10] and potential energy [4]. Helbing and Molnár's Social Force model approach [11] employs attractive and repulsive forces to model pedestrian interactions. While inspiring, these rule-based models are too simple to account for the complexities that arise within human interactions.

Due to its superior performance, machine learning has played an active role in analyzing other types of pedestrian behavior. A popular topic in the field is pedestrian trajectory prediction. [1] uses the SocialLSTM model to account for neighboring pedestrian's actions. [40] uses Social Attention to bypass the local neighborhood assumption. [9] uses generative adversarial models with a global social pooling layer. More sophisticated models developed by [29] employ a multi-modal approach by taking the local image patches around pedestrians as extra inputs. These models generate predicted trajectories that simulate realistic pedestrian interaction behavior. However, these models require precise tracking of pedestrian locations, which is often infeasible for real-world robot platforms.

There have also been studies on the grouping of pedestrians. Previously, [34] proposed using structural support vector machine-based learning to model social groups, [22] modeled the grouping process as a sequential Monte Carlo process. Additionally, the dynamics of grouping has been actively incorporated in pedestrian tracking problems. [27] and [46] believe social groups can be used to enhance the tracking of pedestrians, including social group merges and splits. [12] incorporated group splits and merges in an extended maximum a posteriori problem. [21] utilized Bayesian multiple-hypothesis tracking. However, due to the nature of the tracking task, pedestrian information during merging or splitting is available to these models. In some cases, even future pedestrian information is available to the models. In contrast, our task is a prediction task and is excluded from observing any information about the pedestrians from the moment when the split or merge happens.

A similar category of tasks that has potential application in pedestrian behavior analysis is group action recognition. [13] uses hierarchical recurrent networks to identify joint actions performed by athletes during a sporting event. [31] and [32] utilize a similar concept on pedestrians to predict group activities such as queuing or crossing the street. Resembling trajectory prediction tasks, these approaches depend on tracking individual pedestrians. In addition, these approaches only model within-group activities, whereas splits and merges involve multiple groups.

B. Deep Learning of Videos

The Long-Short Term Memory network (LSTM), a kind of Recurrent Neural Network (RNN), has also shown recent success in sequential data analysis tasks. Some models use LSTMs to process the features produced by CNNs [5], [41]. [30] and [25] proposed Convolutional LSTM (ConvLSTM) for video prediction tasks. However, [39] suggested that any form of RNN breaks the aforementioned video patches into short clips, likely leading to sub-optimal performance. Our initial attempts on RNN-based architectures also resulted in unsatisfactory performance.

In recent years, Convolutional Neural Networks (CNN) have had great success in tackling image processing challenges because of their ability to encode useful spatial features from huge numbers of images [44]. [14] proposed a 3D CNN that was similar to a traditional CNN, but used 3D kernels to jointly encode spatial-temporal features. Shortly after, [37] created a C3D network to classify videos successfully. Since then, many networks based on 3D convolutions [19], [36], [39] have shown great performance on tasks such as action recognition. We believe C3D's success lies in its ability to combine video frames together as video patches.

III. APPROACH

Group shapes can be generated using arbitrary algorithms, assuming that the resulting group shapes reflect temporal patterns as the pedestrians progress. We first formulate two group shape generation algorithms. One from pedestrian information to compare our approach with trajectory prediction models. And another from simulated 2D laser scan points to demonstrate the flexibility of group shape generation algorithms that our model can handle.

A. Group Shapes from Pedestrians

Social group definition. In our definition, a social group is formed when a number of people who are in close proximity of each other share largely similar motion characteristics. Therefore, when a group of fast-walking people make a small split to pass a slow-walking pedestrian, the fast group is maintained, but they do not include the slow-walking pedestrian in their group due to the different walking speeds. Suppose there are n pedestrians in frame V_i. The motion characteristics we use to define grouping are their positions $P_i = \{(x_{i1}, y_{i1}), (x_{i2}, y_{i2}), \ldots, (x_{in}, y_{in})\}$, their velocity directions $\Theta_i = \{\theta_{i1}, \theta_{i2}, \ldots, \theta_{in}\}$, and their velocity magnitudes $S_i = \{s_{i1}, s_{i2}, \ldots, s_{in}\}$.

Grouping algorithm. For each image frame V_i, we apply the approach from [3], the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm [6]. We apply DBSCAN three times, each time within the clusters

Fig. 2. Left: A sample individual social space. Right: A sample group space from simulated laser scans. Note that the lower-right pedestrian is occluded from the "robot" (blue circle) and is not included in the group.

from the previous DBSCAN. We performed DBSCAN in the order of Θ_i, S_i, P_i, and obtain the group membership labels $L_i = \{l_{i1}, l_{i2}, \ldots, l_{in}\}$. During each DBSCAN pass, we use a threshold value to determine the clustering boundary. These threshold values are determined by observing the grouping outcomes on the dataset, and group membership assignments can be changed by adjusting these values.

Individual social space. Once we have the group membership labels, we can define how to generate a social group space. We first define the social space of a single pedestrian $f_s(x, y, \theta, s)$ as a 2D asymmetric Gaussian distribution similar to [16], shown in Figure 2. Given s, we first construct four axes corresponding to the front, the two sides and the rear of the pedestrian, with the pedestrian location (x, y) as the origin. Each axis has a variance value:

$$\sigma_f = \max(2s, 0.5), \sigma_s = \frac{2}{3}\sigma_f, \sigma_r = \frac{1}{2}\sigma_f \tag{1}$$

Then, according to [16], each individual's social space can be defined by the following equations:

$$L_e(\varphi) = \sqrt{\frac{C}{\cos^2\gamma/(2\sigma_1) + \sin^2\gamma/(2\sigma_2)}} \tag{2}$$

$$f_s(x, y, \theta, s) = \begin{pmatrix} x + \cos(\varphi + \theta)L_e(\varphi) \\ y + \sin(\varphi + \theta)L_e(\varphi) \end{pmatrix}, \quad \text{for } 0 < \varphi \leq 2\pi \tag{3}$$

where we define $C = 0.35$, $\gamma = \mod(\varphi, \pi/2)$ and σ_1, σ_2 are the variances of two axis that are closest to angle φ.

Group Social Space. After defining individual social space, for each group membership j in image frame V_i, we first obtain all the pedestrians belonging to this group $G_{ij} = \{k|l_{ik} = j\}$. Then, we construct individual social spaces for each of them $\mathcal{S}_{ij} = \{f_s(x_{ik}, y_{ik}, \theta_{ik}, s_{ik})|k \in G_{ij}\}$. Next, we construct a convex hull around the set of these social spaces $H_{ij} = \text{convexhull}(\mathcal{S}_{ij})$. This convex hull H_{ij} is the group social space for group label j in image frame V_i. Some example social group shapes are shown in Figure 1.

Splits and merges. Group splits and merges occur when group memberships of the pedestrians change from frame V_i to frame V_{i+1}. As shown in Figure 1, if pedestrians who had the same group membership G_{ij} now have two memberships among them $G_{(i+1)j_1}, G_{(i+1)j_2}$, then a split occurs. Similarly, if pedestrians who had different group memberships G_{ij_1}, G_{ij_2} now have the same membership $G_{(i+1)j}$, a merge occurs. Note that $j \neq j_1 \neq j_2$.

B. Group Shapes from Simulated Laser Scan Points

Most robots will not have data representing overhead views where the full perimeter of convex hulls are visible. Likewise, many robots may only be equipped with a laser scanner and not have access to full video scenes. Therefore, it is important to also examine the performance of our approach when a robot is limited to a single laser scan plane.

Simulated Laser Scans. For each video frame V_i, we place a "robot" at a random location in the scene unoccupied by pedestrians. The robot is a point and emits rays of lines around itself with 0.1 degree resolution. We assume each pedestrian to be a circle with a diameter of 0.5 meters. A scan point is defined when one of the robot's rays touches the perimeter of a pedestrian's circle. We further add a standard Gaussian noise truncated at ± 5 cm to the coordinates of each scan point. Each scan point shares the same orientation and speed as its corresponding pedestrian. Similar to definitions in Section III-A, we now have laser scan point information that can also be represented as $P_i = \{(x_{i1}, y_{i1}), (x_{i2}, y_{i2}), \ldots, (x_{in}, y_{in})\}$, $\Theta_i = \{\theta_{i1}, \theta_{i2}, \ldots, \theta_{in}\}$, and $S_i = \{s_{i1}, s_{i2}, \ldots, s_{in}\}$.

Grouping algorithm. Similar to Section III-A, we apply DBSCAN on Θ_i, S_i, P_i to obtain the group membership labels $L_i = \{l_{i1}, l_{i2}, \ldots, l_{in}\}$.

Group Social Space. We no longer assume known pedestrian locations as the "robot" only sees the laser scan points (Figure 2, right), so we generate social spaces directly from laser scan points. As would occur with a real laser scanner, the "robot" sometimes fails to observe occluded pedestrians. We obtain the group space H_{ij} of group label j by constructing a convex hull around the group of laser scan points $G_{ij} = \{k|l_{ik} = j\}$ in video frame V_i: $H_{ij} = \text{convexhull}(\{(x_{ik}, y_{ik})|k \in G_{ij}\})$. The splits and merges follow the same formulation as in Section III-A.

C. 3D Convolution Neural Network

The Third Spatial Dimension. Evidence for why 3D convolution works has been vague. Both [14] and [37] proposed 3D Convolutions based on mere intuition that 3D kernels connect spatial features and temporal features together. Although [37] provided examples of learned features, it is hard to distinguish whether these features belong to the spatial dimension or the temporal dimension, as even one frame in the video can signal the entire action. In our case, identifying whether a split or merge takes place from a single image is impossible, so we supplement [14] and [37]'s intuition by providing more concrete evidence.

As shown in Figure 3, splits and merges have unique volumetric features [15] in the temporal dimension. They both represent shapes similar to tree branches. In our task, the decisive branching moments are not available to our model. Thus, an alternative interpretation of our model's goal is to perform 3D object classification by analyzing voxel-grid-represented 3D objects and predict whether these 3D objects will evolve into tree branches.

The architecture. We largely leverage the C3D architecture [37], shown in Figure 4, because it has demonstrated strong performance in action recognition tasks. A difference between

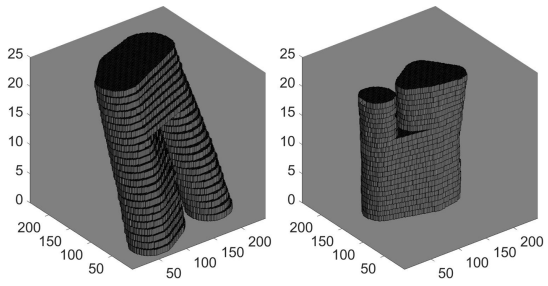

Fig. 3. Volumetric features of a merge (left) and a split (right). Only features before the branch (the 17th layer) are visible to our model.

our network and C3D occurs after the `pool5` layer where our network progresses into two branches with two fully connected hidden layers of 4096 units. The first branch outputs a three-class prediction score $p = (p_0, p_1, p_2)$ with the classes arranged in the order of no-action, a merge, and a split. The second branch predicts a 2D pixel coordinate $r = (r_x, r_y)$, indicating a possible split or merge location regardless of what the other branch predicts.

The location of group splits and merges is a vague concept. When asked where exactly a split or merge takes place, few people can agree on a fixed pixel location. In our problem, we define the ground truth location to be the midpoint of the shortest line that connects the involved two social group spaces as shown in Figure 1. In practice, extreme pursuit of the split and merge location accuracy is meaningless.

Two-task loss function. We have two output layers, the ground truth one-hot class prediction score $p_t = (p_{t0}, p_{t1}, p_{t2})$ and the previously defined ground truth event location $r_t = (r_{tx}, r_{ty})$, so we combine the two loss functions together (similar to [8]) as follows:

$$\mathcal{L}(p, r, p_t, r_t) = \mathcal{L}_{cls}(p, p_t) + \lambda \delta(p_t) \mathcal{L}_{loc}(r, r_t) \tag{4}$$

$\mathcal{L}_{cls}$ is the softmax cross-entropy loss function representing the class prediction loss,

$$\mathcal{L}_{cls}(p, p_t) = -\sum_{i=0}^{2} p_{ti} \log \left(\frac{e^{p_i}}{\sum_{j=0}^{2} e^{p_j}} \right) \tag{5}$$

$\mathcal{L}_{loc}$ is the L2 loss function representing the location prediction loss,

$$\mathcal{L}_{loc}(r, r_t) = (r_{tx} - r_x)^2 + (r_{ty} - r_y)^2 \tag{6}$$

$\delta(p_t)$ is to ensure that the location loss will only be incorporated if the ground truth event is a merge or split. Given that both p and p_t have their class indexes in the order of no-action, merge and split, $\delta(p_t)$ is of the form:

$$\delta(p_t) = \begin{cases} 1, & \text{if } \arg\max_i(p_{ti}) \geq 1 \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

The hyper-parameter λ in Eq. (4) controls how much the location loss influences the overall loss.

IV. Experiments

A. Setup

Our default experiment setting is very similar to [37]. With input videos of size $16 \times 224 \times 224$, the network has 8 3D convolution layers and 5 3D max-pooling layers. Each convolution layer has a kernel size of $3 \times 3 \times 3$ and strides $1 \times 1 \times 1$. This configuration was found to be the most effective [37], similar to 2D CNNs [33]. Each max-pooling layer has a kernel size of $2 \times 2 \times 2$ and strides $2 \times 2 \times 2$, except for `pool1`, which has strides $1 \times 2 \times 2$ to accommodate for our 16-frame inputs.

During training for all of the experiments, we used $\lambda = 0.005$. Using an Adam optimizer, the initial learning rate was set to $1\text{e}{-5}$ with a batch size of 1. Details on the inputs to the network are presented in the following section.

B. Datasets

Our raw dataset was a mixture of the publicly available ETH dataset [26] and UCY dataset [17]. These two datasets have been commonly used in pedestrian trajectory prediction problems [1], [28], [40], [43]. Both datasets contain complex group interaction behaviors [26] and have their videos recorded at 25 FPS. The ETH dataset [26] contains two sets of data from two scenes (ETH and HOTEL). The UCY dataset [17] contains three sets of data from another two scenes (ZARA1, ZARA2, UCY1).

To generate training data for a merge instance, suppose the merge happens at time $i + 1$ and we want to predict this event n number of frames beforehand. Also, suppose the merging groups have labels j_1, j_2. From this, we can first obtain convex hulls $H_{j_1} = (H_{(i-n-15)j_1}, \ldots, H_{(i-n)j_1})$ and similarly convex hulls H_{j_2} by following the procedures in Section III. Next, we pasted these convex hulls into blank images, frame by frame, to generate a binary social space video $W = (W_{i-n-15}, \ldots, W_{i-n})$. Because we want to filter out noisy groups, W would be invalid training data if j_1 or j_2 is missing in these 16 frames. Generating training data for a split follows a similar method, the only difference being that there is now only one convex hull blob in the input video instead of two convex hull blobs.

To generate training data for no-action cases, we randomly sampled a time step $i + 1$ and a group label j in L_i. Then, we constructed $H_j = (H_{(i-15)j}, \ldots, H_{ij})$ and generated the training data W. We also need to construct no-action data with two convex hull blobs to distinguish between merge and no-action. In this case, the second convex hull sequence $H_{j'}$ was defined such that the centroid of $H_{ij'}$ was the nearest neighbor to the centroid of H_{ij} among all the convex hull centroids at time i.

Unfortunately, group splits and merges are infrequent events. Training on inputs that are 1 frame ahead of the event only gives us a total of 477 splits and 367 merges across all 5 sets of data. To improve training, we first performed scale normalization for each W to limit the size range of all convex hull blobs. This was done by cropping the empty space around the input volumetric feature. Then, we performed translation normalization so that the geometric center of the group shapes in the last input frame is at the center. We then performed data augmentations to randomly flip the video images or rotate them by an arbitrary angle. Also due to the scarcity of the events, we did not perform evaluations on time horizons longer

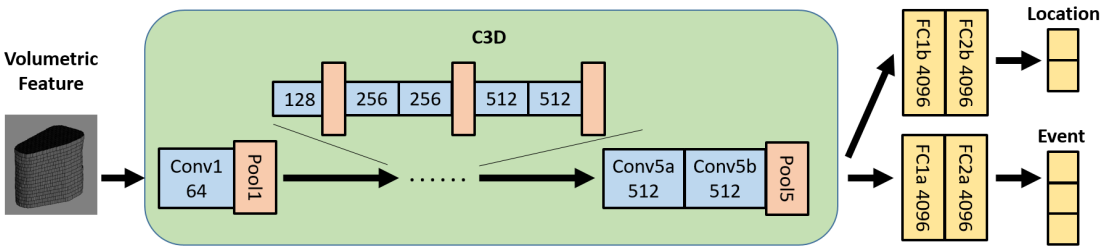

Fig. 4. Our 3D convolutional neural network architecture. Each blue block represents a 3D convolution block with the number indicating the amount of output channels. Each orange block represents a pooling block. Each yellow block represents a fully connected layer. The portion of our architecture within the green block is the same as C3D [37] before the fully connected layers. Sharing the `pool5` layer, two branches of fully connected layers predict split and merge occurrences and locations respectively.

TABLE I
COMPARISON OF OUR APPROACH WITH SOCIAL-LSTM, SOCIAL-GAN AND SR-LSTM MODELS ON PREDICTION ACCURACY (F1 SCORE)

Method		S-LSTM [1]		S-GAN [9]		SR-LSTM [45]		Ours		Ours: Laser Scans	
Prediction Time		1 frame	0.5 sec	1 frame	0.5 sec	1 frame	0.5 sec	1 frame	0.5 sec	1 frame	0.5 sec
ETH	N.A.	58.4	56.3	66.2	53.6	**68.9**	59.0	60.5	**64.2**	49.5	41.7
	Merge	36.6	16.7	59.7	40.0	58.0	39.0	**76.7**	**80.0**	76.9	45.8
	Split	53.3	27.9	63.2	29.3	54.8	11.8	**86.5**	**87.3**	71.7	46.7
	AVG	49.4	33.6	63.0	41.0	60.6	36.6	**74.6**	**77.1**	66.0	44.7
HOTEL	N.A.	57.2	56.8	67.3	54.1	**69.3**	**58.1**	58.5	53.1	65.0	53.2
	Merge	44.5	14.8	68.9	22.2	67.7	12.9	**78.5**	**63.6**	79.4	74.8
	Split	56.3	53.1	55.3	27.8	68.7	18.2	**88.4**	**86.3**	82.9	56.7
	AVG	52.7	41.5	63.8	34.7	68.6	29.7	**75.1**	**67.7**	75.8	61.6
ZARA1	N.A.	57.9	51.4	60.8	53.9	57.9	54.1	**62.5**	**60.0**	58.5	50.0
	Merge	57.1	24.0	**78.1**	25.0	57.1	48.3	76.0	**62.5**	78.4	80.0
	Split	40.9	30.8	47.6	18.8	50.0	19.4	**90.6**	**92.0**	83.0	86.2
	AVG	52.0	35.4	62.2	32.5	55.0	40.6	**76.4**	**71.5**	73.3	72.1
ZARA2	N.A.	55.7	50.0	**58.2**	**51.7**	55.8	51.5	49.5	50.5	56.1	55.6
	Merge	34.5	20.4	43.4	16.7	40.4	30.8	**81.2**	**70.6**	69.1	30.8
	Split	35.3	20.9	34.5	22.6	34.6	22.2	**78.7**	**75.5**	77.4	75.9
	AVG	41.8	30.4	45.4	30.3	43.6	31.5	**69.8**	**65.5**	67.5	54.1
UCY1	N.A.	51.5	50.2	50.8	51.4	52.2	52.4	**59.6**	**59.7**	51.4	57.2
	Merge	31.6	34.7	34.9	26.3	32.3	26.9	**54.8**	**52.9**	58.0	48.8
	Split	34.4	41.2	36.2	29.3	24.5	22.1	**82.6**	**84.5**	77.8	83.3
	AVG	39.2	42.1	40.6	35.7	36.3	33.8	**65.6**	**65.7**	62.4	63.1
Total Average		47.0	36.6	55.0	34.8	52.7	34.4	**72.3**	**69.5**	69.0	00.0

than 0.5 seconds. At 0.5 seconds, there are only a total of 313 splits and 214 merges. And at 1 second, there are only a total of 229 splits and 153 merges.

C. Evaluation

Due to the size of dataset, we performed leave-one-out cross-validation. This evaluation approach is also adopted by [1], [9], [45]. We trained our model on 4 sets of data and evaluated it on the remaining set.

Comparisons with trajectory models. A logical approach is to generate group shapes from trajectory prediction models. Because local overhead image patches around pedestrians are infeasible to a ground-based, real-world robot, we did not compare with models that take local image patches as inputs (e.g., [29], [42]). Therefore, we used Social-LSTM [1], Social-GAN [9] as baselines and SR-LSTM [45] as the state-of-the-art model for comparisons.

Social-LSTM, Social-GAN and SR-LSTM use input sequences of 8 frames, so we changed the `pool2` layer of our model to have strides $1 \times 2 \times 2$ similar to `pool1`. For these models, we applied our grouping algorithm on the first frame to determine the pedestrians' original group memberships. We then fed the pedestrians' trajectories into these models to obtain the predicted future trajectories. Next, we applied our grouping algorithm to the predicted future trajectories to obtain their new group memberships. The change in memberships allowed us to determine whether these models can predict splits, merges, or no-actions. We evaluated the models on all of the merges and splits and an equal number of no-action sequences on the test dataset. Then, we used the leave-one-out approach following a similar evaluation methodology as [1], [9], [45].

To allow location prediction accuracy comparisons, we also modified the trajectory based models by applying our group shape generation pipeline on the predicted trajectories. Once we obtained the group shapes, we applied our definition to estimate the split and merge location.

Since this is a categorization, all models were evaluated on the usual classification metrics of precision, recall, and F1 score for the three ground truth events (no action, merge, and split). We only report F1 scores in Table I, but saw our model regularly outperform the others for all three metrics.

As shown in Table I, our method was generally better than these techniques. Although for a different task, Social-GAN and SR-LSTM models still outperformed the baseline Social-LSTM models. However, our approach outperformed

TABLE II
COMPARISON OF OUR APPROACH WITH SOCIAL-LSTM, SOCIAL-GAN AND SR-LSTM MODELS ON LOCATION PREDICTION ERROR (IN PIXELS)

Dataset	ETH	HOTEL	ZARA1	ZARA2	UCY1
S-LSTM	9.21	11.20	5.56	19.41	27.58
S-GAN	7.73	9.46	5.88	7.98	14.14
SR-LSTM	**6.55**	**8.03**	**5.53**	**7.22**	**8.91**
Ours	17.66	13.89	15.11	15.55	21.86
Ours: Laser Scans	19.56	18.78	15.55	20.45	26.86

all other models on all three metrics, especially for splits and merges. Social-LSTM, Social-GAN and SR-LSTM models performed better in predicting no-actions, but they were weak at rejecting false negatives, resulting in an overall F1 score that was on par with our model. We also observed that all three models showed a strong bias towards predicting no-actions. This demonstrates that individual pedestrian-based models are unable to accurately capture complex pedestrian interactions, such as grouping, even when modified for social behaviors.

When making predictions 0.5 seconds ahead, the trajectory models incur a significant performance downgrade while our model's performance drops moderately. This indicates that our model successfully captures the temporal clues within the inputs to predict temporally distant splits and merges. In contrast, the trajectory models are lackluster when predicting splits and merges in the far future. Note that the trajectory models' performances lowered to near random guess levels and are likely to downgrade further for longer time horizons.

Our model's performance on predicting no-actions was generally worse than for splits and merges. This is because too much data variation exists in the no-action class during training. Our model was trained on equal numbers of each class instance. In reality, no-action instances vastly dominate pedestrian interactions and map to far more possible social group shape transformations. However, feeding too much training data from the no-action class leads to the class imbalance problem [2]. The issue of large data variety with limited amounts of data is an important area for future work.

In Table II, we can see that the state-of-the-art baseline models performed ~ 10 pixels better than our approach. This is expected because our definition of split and merge location is dependent on pedestrian positions, which are unknown to our model. Due the arbitrary nature of split and merge locations, 10 pixels is not a huge practical advantage. We also believe that being able to predict whether a split or merge will occur is more important than pinpointing the location.

Accuracy across parameters. Recall from Section III that we use DBSCAN as our grouping algorithm with three threshold values. These threshold values were determined subjectively, so a sensitivity analysis was performed on models trained with inputs from the social generation method in Section III-A. Varying these values also simulates how pedestrian behavior in social groups can vary greatly across cultures (e.g., [35]). Therefore, this analysis shows how well our model transfers when one of the grouping parameters changes. We selected two models evaluated on ETH and UCY1 to represent a normal and a difficult scenario, because our model performed moderately on ETH and the worst on UCY1 as shown in Table I. Then, we adjusted each grouping parameter to examine our model's adaptability.

TABLE III
SENSITIVITY ANALYSIS ON DIFFERENT SETS OF GROUPING PARAMETERS (F1, LOCATION ACCURACY IN PIXELS)

Threshold Values	ETH	UCY1	Threshold Values	ETH	UCY1
$\downarrow P = 1.5$ $O = 30$ $V = 1.0$	48.7 (17.05)	70.1 (15.79)	$\uparrow P = 2.5$ $O = 30$ $V = 1.0$	63.7 (16.69)	60.9 (22.64)
$P = 2.0$ $\downarrow O = 15$ $V = 1.0$	54.6 (17.05)	64.0 (15.15)	$P = 2.0$ $\uparrow O = 45$ $V = 1.0$	70.1 (14.75)	64.7 (18.67)
$P = 2.0$ $O = 30$ $\downarrow V = 0.5$	63.2 (16.75)	66.2 (18.58)	$P = 2.0$ $O = 30$ $\uparrow V = 1.5$	64.6 (16.25)	66.3 (19.02)
$P = 2.0$ $O = 30$ $V = 1.0$	65.9 (16.54)	67.2 (18.98)			

We evaluated performance using event prediction accuracy and average event location prediction error. The latter was measured in the normalized images, as mentioned at the end of Section IV-B. This error can be significantly less when the prediction is projected back to the raw image.

In Table III, P is the position distance threshold in meters; O is the velocity orientation threshold in degrees; and V is the velocity magnitude threshold in meters per second. Arrows indicate an increase or decrease of the corresponding parameter from the value used for the single frame prediction in Table I. The values of average location errors are in parenthesis. From the table, we can infer that our approach demonstrates excellent model transfer abilities across different parameter settings both in terms of prediction accuracy and location prediction error.

Comparison to simulated laser scans. As mentioned in Section III, we approximated a simulated laser scan for use in our model. This was to demonstrate that our model is compatible with various types of inputs. Input video sequences generated from simulated laser scans are noisier and can neglect occluded pedestrians when compared to those generated from basic pedestrian information. Training for this analysis was stopped once performance had reached levels comparable to our regular approach. As shown in Table I and Table II, applying our model on simulated laser scans results in similar prediction accuracies, but location predictions are less accurate due to inputs being more challenging.

Future work should examine performance from real laser scan data, but this analysis hints that our approach will be effective with laser scans.

Group interaction signals and evidence on 3D CNN. A key concern is whether the process is understandable by humans. We took the feature map of `conv5b` and followed the deconvolution pipeline [44] to trace the feature map layer back to the input image space. We selected the highest activation value in the `conv5b` feature map for four example cases. We then inspected the corresponding input image space projections to examine what parts of the input image contribute to these activation values (Figure 5). The results suggest that

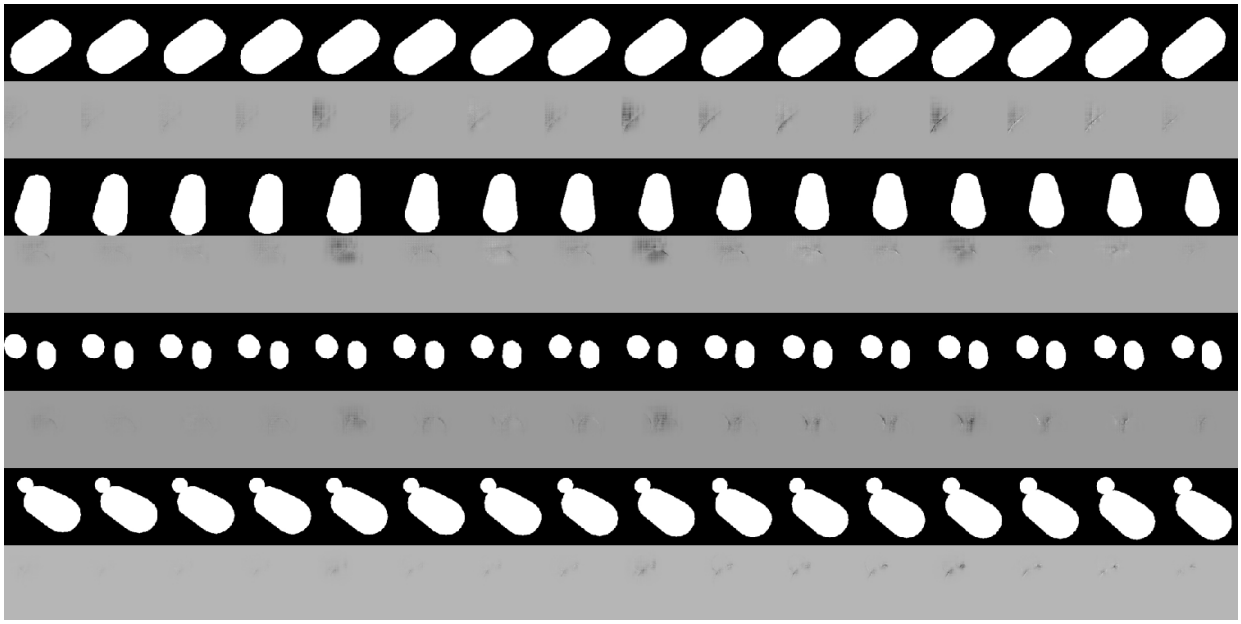

Fig. 5. Visualization of our model using the method from [44]. The first four rows are two inputs corresponding to two cases of splits and their learned contributions to the highest activation value in `conv5b`. The last four rows correspond to two cases of merges.

the features captured by our model can also be interpreted from a behavioral perspective:

1) In the first split example, our network focused on the portion of the leading edge close to the bottom-left individual. As the bottom-left person moved farther away from the other person, our network captured the increase in length of that portion of the leading edge. From a human perspective, a leading edge increase reflects a gap increase within the group.
2) In the second split example, our network focused on the round edge above the person at the top. The projected features show that as time progresses, the network rotated its attention along the group space boundary counter-clockwise. From a human perspective, this means that a subgroup within the group starts to show signs of changing movement direction.
3) In the first merge example, our network focused on the two edges of the two groups that are closest to each other. Over time, our network captured the trend that these two edges are approaching each other, indicating a merge. This is also consistent with how humans predict merges by observing diminishing gaps.
4) In the second merge example, as a group of people walked past an individual, the individual sped up and joined the group. As a result, the social space around this individual grew larger. Our network captured this by noticing the shrinking of two "cracks" around their combined social space. From a human perspective, this can be interpreted as an individual's behavior evolving to conform with a group's behavior.

Based on the the learned features shown in Figure 5, we can infer that our model captures temporal features of our input data. If we concatenate these learned temporal features frame by frame, we can then interpret the results as spatial features of our volumetric features. Examples include an enlarging surface, a slightly twisted round curve, two surfaces that are about to intersect, and the narrowing of two dents, etc.

V. Conclusion

To enhance robot navigation efficiency and social navigation near groups of moving pedestrians, we present a 3D CNN model for predicting social group splits and merges. We first developed a pipeline that transformed pedestrian information into social group spaces. Then, we utilized a modified C3D network [37] since volumetric features [15] can transform the temporal dimension into a spatial dimension and 3D CNNs excel at encoding 3D spatial features. We showed that our approach was (a) on par with, or better than, the state-of-the-art pedestrian trajectory prediction models for predicting the occurrence of splits and merges and (b) transferred well across different prediction times and cultural settings. However, our approach does require a diverse training dataset.

This work included secondary results valuable towards future research efforts. We provide examples demonstrating that our model learns features that can be interpreted from a human perspective. We also showed, using an approximation of laser scan data, that our approach has potential for robot deployments that lack access to overhead views. Finally, our model's success also provides evidence that 3D Convolution learns temporal features in videos.

This work is only a partial step towards giving robots the ability to predict group events and navigate in an appropriate manner. Future work will involve solving the data variation imbalance problem for the no-action class. We also need more data to strengthen the model's adaptability for different scenes and enable prediction of group event time. Finally, we need to modify our social group space generation pipeline to enable real-world robots to fully take advantage of this

approach. This will allow future systems to skip the expensive step of detecting individual pedestrians, which is needed for trajectory-based approaches. This will also confirm that laser scanners and robot camera views that occlude group members can be supported with our approach.

ACKNOWLEDGMENT

This work was funded by grant (IIS-1734361) from the National Science Foundation.

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 *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, June 2016, pp. 961–971.

[2] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Networks*, vol. 106, pp. 249 – 259, 2018.

[3] I. Chatterjee and A. Steinfeld, "Performance of a low-cost, human-inspired perception approach for dense moving crowd navigation," in *Proc. IEEE Int. Symp. Robot and Human Interact. Commun.*, Aug 2016, pp. 578–585.

[4] X. Cui, Q. Liu, M. Gao, and D. N. Metaxas, "Abnormal detection using interaction energy potentials," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, June 2011, pp. 3161–3167.

[5] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, June 2015, pp. 2625–2634.

[6] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. Int. Conf. Knowl. Discovery and Data Mining*, 1996, pp. 226–231.

[7] I. Gadol, "Beyond the hot seat: Gestalt approaches to group," *Int. J. of Group Psychotherapy*, vol. 31, no. 2, pp. 262–264, 1981.

[8] R. Girshick, "Fast r-cnn," in *Proc. IEEE Int. Conf. Comput. Vis.*, December 2015, pp. 1440–1448.

[9] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social gan: Socially acceptable trajectories with generative adversarial networks," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, June 2018, pp. 2255–2264.

[10] D. Helbing, "A Fluid Dynamic Model for the Movement of Pedestrians," *arXiv:cond-mat/9805213*, May 1998.

[11] D. Helbing and P. Molnár, "Social force model for pedestrian dynamics," *Phys. Rev. E*, vol. 51, pp. 4282–4286, May 1995.

[12] J. F. Henriques, R. Caseiro, and J. Batista, "Globally optimal solution to multi-object tracking with merged measurements," in *Proc. IEEE Int. Conf. on Comput. Vis.*, Nov 2011, pp. 2470–2477.

[13] M. Ibrahim, S. Muralidharan, Z. Deng, A. Vahdat, and G. Mori, "A Hierarchical Deep Temporal Model for Group Activity Recognition," *arXiv:1511.06040*, Nov 2015.

[14] S. Ji, W. Xu, M. Yang, and K. Yu, "3d convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, Jan 2013.

[15] Y. Ke, R. Sukthankar, and M. Hebert, "Efficient visual event detection using volumetric features," in *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 1, Oct 2005, pp. 166–173.

[16] R. Kirby, "Social robot navigation," Ph.D. dissertation, Carnegie Mellon University, Pittsburgh, PA, May 2010.

[17] A. Lerner, Y. Chrysanthou, and D. Lischinski, "Crowds by example," *Comput. Graph. Forum*, vol. 26, no. 3, pp. 655–664, 2007.

[18] S. Liang and R. Srikant, "Why Deep Neural Networks for Function Approximation?" *arXiv:1610.04161*, Oct. 2016.

[19] Z. Liu, C. Zhang, and Y. Tian, "3d-based deep convolutional neural network for action recognition with depth sequences," *Image and Vis. Comput.*, vol. 55, pp. 93 – 100, 2016.

[20] S. Ljungblad, J. Kotrbova, M. Jacobsson, H. Cramer, and K. Niechwiadowicz, "Hospital robot at work: Something alien or an intelligent colleague?" in *Proc. ACM Conf. on Comput. Supported Cooperative Work*, 2012, pp. 177–186.

[21] A. Makris and C. Prieur, "Bayesian multiple-hypothesis tracking of merging and splitting targets," *IEEE Trans. Geosci. and Remote Sens.*, vol. 52, no. 12, pp. 7684–7694, Dec 2014.

[22] L. Mihaylova, A. Y. Carmi, F. Septier, A. Gning, S. K. Pang, and S. Godsill, "Overview of bayesian sequential monte carlo methods for group and extended object tracking," *Digit. Signal Process.*, vol. 25, pp. 1 – 16, 2014.

[23] M. Moussaïd, N. Perozo, S. Garnier, D. Helbing, and G. Theraulaz, "The walking behaviour of pedestrian social groups and its impact on crowd dynamics," *PLOS ONE*, vol. 5, pp. 1–7, 04 2010.

[24] B. Mutlu and J. Forlizzi, "Robots in organizations: The role of workflow, social, and environmental factors in human-robot interaction," in *Proc. ACM/IEEE Int. Conf. Human Robot Interact.*, 2008, pp. 287–294.

[25] V. Patraucean, A. Handa, and R. Cipolla, "Spatio-temporal video autoencoder with differentiable memory," *arXiv:1511.06309*, Nov. 2015.

[26] S. Pellegrini, A. Ess, K. Schindler, and L. van Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, Sept 2009, pp. 261–268.

[27] A. G. A. Perera, C. Srinivas, A. Hoogs, and G. B. and, "Multi-object tracking through simultaneous long occlusions and split-merge conditions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 1, June 2006, pp. 666–673.

[28] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese, "Learning social etiquette: Human trajectory understanding in crowded scenes," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 549–565.

[29] A. Sadeghian, V. Kosaraju, A. Sadeghian, N. Hirose, S. H. Rezatofighi, and S. Savarese, "SoPhie: An Attentive GAN for Predicting Paths Compliant to Social and Physical Constraints," *arXiv:1806.01482*, Jun 2018.

[30] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-k. Wong, and W.-c. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in *Adv. Neural Inf. Process. Syst.*, 2015, pp. 802–810.

[31] T. Shu, S. Todorovic, and S.-C. Zhu, "CERN: Confidence-Energy Recurrent Network for Group Activity Recognition," *arXiv:1704.03058*, Apr 2017.

[32] T. Shu, D. Xie, B. Rothrock, S. Todorovic, and S.-C. Zhu, "Joint Inference of Groups, Events and Human Roles in Aerial Videos," *arXiv:1505.05957*, May 2015.

[33] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv:1409.1556*, Sept. 2014.

[34] F. Solera, S. Calderara, E. Ristani, C. Tomasi, and R. Cucchiara, "Tracking social groups within and across cameras," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 3, pp. 441–453, March 2017.

[35] A. Sorokowska, P. Sorokowski, P. Hilpert, K. Cantarero, T. Frackowiak, and et al., "Preferred interpersonal distances: A global comparison," *J. Cross-Cultural Psychol.*, vol. 48, no. 4, pp. 577–592, 2017.

[36] L. Sun, K. Jia, D.-Y. Yeung, and B. E. Shi, "Human action recognition using factorized spatio-temporal convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, December 2015.

[37] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, December 2015.

[38] P. Trautman and A. Krause, "Unfreezing the robot: Navigation in dense, interacting crowds," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, Oct 2010, pp. 797–803.

[39] G. Varol, I. Laptev, and C. Schmid, "Long-term temporal convolutions for action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1510–1517, June 2018.

[40] A. Vemula, K. Muelling, and J. Oh, "Social attention: Modeling attention in human crowds," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2018, pp. 1–7.

[41] Z. Wu, X. Wang, Y.-G. Jiang, H. Ye, and X. Xue, "Modeling spatial-temporal clues in a hybrid deep learning framework for video classification," in *Proc. ACM Int. Conf. Multimedia*, 2015, pp. 461–470.

[42] H. Xue, D. Q. Huynh, and M. Reynolds, "Ss-lstm: A hierarchical lstm model for pedestrian trajectory prediction," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, vol. 00, Mar 2018, pp. 1186–1194.

[43] K. Yamaguchi, A. C. Berg, L. E. Ortiz, and T. L. Berg, "Who are you with and where are you going?" in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, June 2011, pp. 1345–1352.

[44] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 818–833.

[45] P. Zhang, W. Ouyang, P. Zhang, J. Xue, and N. Zheng, "Sr-lstm: State refinement for lstm towards pedestrian trajectory prediction," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, June 2019, pp. 12 085–12 094.

[46] F. Zhu, X. Wang, and N. Yu, "Crowd tracking with dynamic evolution of group structures," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 139–154.