SOCIAL NAVIGATION WITH PEDESTRIAN GROUPS

Allan Wang

PHD THESIS

December 15, 2023

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Robotics.

CMU-RI-TR-23-88

The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee: Aaron Steinfeld, *Chair*
Jean Oh
Katia Sycara
Takayuki Kanda, *Kyoto University*

ABSTRACT

Autonomous navigation in human crowds (i.e., social navigation) presents several challenges: The robot often needs to rely on its noisy sensors to identify and localize pedestrians in human crowds; the robot needs to plan efficient paths to reach its goals; and the robot needs to do so in a safe and socially appropriate manner. Recent work has proposed model-based methods with an emphasis on modeling specific interaction scenarios and learning-based methods to tackle the navigation problem end-to-end. Model-based methods lack adaptation in complex crowded environments, while learning-based methods do not have access to large, complex datasets and can only be trained in unrealistic simulators.

In this thesis, we focus on the novel angle of leveraging pedestrian groups to address the social navigation problem. We first introduce the concept of social group space via group split and merge predictions and formulate a model for group state predictions. We further show that split and merge predictions on group-based representations are more accurate than predictions made on individual-based representations. Second, we integrate our group-based representations and prediction models into a Model Predictive Control (MPC) framework. We show that compared to individual-based representations in the same MPC framework, our framework produces safer and more socially appropriate motions. This demonstrates the benefit of model-based methods when coupled with a learning-based state predictor. Third, we propose a simplified representation of the social group space based on the visible edges of the groups. We show that the simplified representation can replace our original representation in an MPC framework by maintaining similar performance levels while significantly reducing computation time.

In parallel to these contributions, we address the need for real-world large-scale pedestrian

datasets in training learning-based methods for social navigation. We also identify a similar need to capture greater varieties of group-based pedestrian interactions. In response to these needs, we introduce our own scalable data collection efforts and dataset: the TBD Pedestrian Dataset. Our data collection pipeline enables efficient collection and labeling of large quantities of data. Our publicly available dataset contains both top-down and ego-centric view sensor data and is much larger than similar prior datasets. This contribution will dramatically advance the work on social robot navigation.

ACKNOWLEDGMENTS

First and foremost, I want to thank my advisor Aaron Steinfeld. Thank you for all the support during all these years. Thank you for the valuable input to my research projects and for inspiring me to pursue many interesting ideas. Thank you for encouraging me when I feel nervous about my progress. Thank you for your life advice. Thank you for connecting me to many other colleagues, many of whom I had the pleasure to work with. Thank you for bringing me into the wonderful world of human-robot interaction. Having you as my advisor is one of the best things that has happened in my life.

I would also like to offer my special thanks to Christoforos Mavrogiannis. Your mentorship on the group-based navigation project opened my mind to many things we can do with pedestrian groups and helped me obtain an incredible oral presentation at CoRL. I would also like to thank Daisuke Sato. Thank you for letting me use the incredible suitcase robot, Cabot, and for supporting me in our data collection effort wholeheartedly. I also want to thank Abhijat Biswas. We had plenty of collaboration during the early years of my PhD and we built many good systems useful for social navigation together. Next, I want to thank Yasser Corzo. Among all the students that I have mentored, you contributed the most to my research projects. Thank you for your great work in laying down the foundation work for the label correction app.

Next, I would like to thank the rest of my committee members Jean Oh, Katia Sycara, and Takayuki Kanda. Thank you all for your valuable feedback on my thesis and for brainstorming inspiring ideas to pursue on social navigation. Thank you, Takayuki for inviting me to visit your lab in 2022. It was an incredible experience.

I have also had the chance to collaborate with many incredible people: Nathan Tsoi, Zakia

Hammal, Changdo Song, Marynel Vázquez, Raj Mehta, Xinyi Lu, Claire Chen, Sonya Simkin, Dapeng Zhao, Francesca Baldini, Pete Trautman, Gustavo Silvera, Henny Admoni, Ben Stoler, Phani Teja Singamaneni, Rohan Chandra, Alhanof Alolyan, Ishani Chatterjee, Lynn Urbina, Aditi Kulkarni, Bernardine Dias, Nicholas Tiwari, Ada Taylor, and many others. Thank you all for your valuable input and contributions to my research, and for collaborating on projects and event organization.

Next, I want to thank my Transportation, Bots, and Disability (TBD) labmates: Xiang Zhi Tan, Liz Carter, Samantha Reig, Katherine Shih, Lynn Kirabo, Jirachaya (Fern) Limprayoon, Oscar Romero, Ceci Morales, Mary Hatfalvi, Sarthak Ahuja, Tesca Fitzgerald, Alex Haig, Prithu Pareek, Amal Nanavati, Byung Cheol Min, Suresh Kumaar Jayaraman, Joe Connolly, Sikai Chen, Huy Quyen Ngo, Rayna Hata, Abena Boadi-Agyemang and other current and former TBD Lab members. It is now my 9th year in the lab since I joined as an undergraduate student. I cherished every moment with every one of you and will forever remember you in my heart. Especially Zhi, we joined the lab at around the same time. Thank you for your guidance and mentorship during all the years when we were together. Thank you Kate for helping me proofread this thesis.

I would also like to thank the many friends I had the pleasure to make within the Robotics Institute and the Human-Robot Interaction sub-community. We had many intellectual conversations and did many fun activities together. You all made Robotics Institute feel like home to me. Especially to Chia Dai, Martin Li, and Zirui Wang. Thank you for your support during the entirety of my PhD. I also want to extend my special thanks to my long-time officemate, Ben Newman. We had such a similar life trajectory, but you were always one step ahead of me. Thank you for sharing your experience and your wisdom with me.

Apart from my research collaborators and peers, I also want to thank the Robotics Institute administrative staff members Suzanne Muth, Peggy Martin, and Rachel Burcin. Thank you for your suggestions and support. Thank you for handling the many logistical challenges for me.

Most importantly, I want to thank my wife Tongtong Jiang. Thank you for taking care of me during the darkest of times. Without you, it would be much more challenging to endure my most difficult times. Without you, my PhD career would not be possible.

Finally, I would like to thank my funding sources: the National Science Foundation (IIS-1734361), National Institute on Disability, Independent Living, and Rehabilitation Research (NIDILRR, 90DPGE0003), and Office of Naval Research (ONR N00014-18-1-2503). I also want to thank Jie Tan and Lewis Chiang from Google for your support and feedback on my research.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Part I

Introduction

CHAPTER 1

INTRODUCTION

Figure 1.1: A food delivery robot developed by Starship Technologies.

Over the last three decades, as mobile robots' navigation capabilities have increased, we have seen growing interest in mobile robots navigating in pedestrian environments. Early deployments of this type have focused on autonomous mobile robots that serve as tour guide robots in museums, such as RHINO [Burgard et al., 1998] and Minerva [Thrun et al., 1999]. More recently, mobile robots have taken on the tasks of cleaning and delivery in crowded public areas, leading to documented problems with social behavior [Mutlu and Forlizzi, 2008]. In particular, the ability to navigate among humans is a necessity for autonomous service robots (an example of which is shown in Figure 1.1 [1]). This ability, often referred to as *social navigation*, requires the robot to perform socially acceptable navigation actions that cause minimal disturbance to pedestrians while preserving efficiency in reaching task goals. However, many commercial service robots today lack full social navigation capacity. They often run on predefined paths and employ a stop-and-go strategy when a pedestrian approaches. These overly constrained navigation settings and strategies result in low efficiency for the robots in achieving their tasks, and the predefined paths severely

[1] "16a.Robot.Postmates.WDC.25October2017" by Elvert Barnes is licensed under CC BY-SA 2.0

limit the robots' possible navigation goals.

The primary reason a stop-and-go strategy has low efficiency is that service robots often run into dense human crowds. In a dense population, the robot frequently detects nearby human presence and waits. Even if the robot is equipped with a planner, a conservative estimate of the pedestrian dynamics could result in no free space for the robot to navigate. This is referred to by Trautman and Krause [2010] as *the freezing robot problem*.

To address this issue, researchers have concluded that modeling human-human and human-robot navigation interactions is the key to improving efficiency while maintaining safety. This is explored using both model-based interaction modeling methods [van den Berg et al., 2011, Trautman et al., 2015, Singamaneni et al., 2021] and learning-based methods [Chen et al., 2017, Chen et al., 2019, Liu et al., 2021, Kästner et al., 2020]. A survey paper by [Mavrogiannis et al., 2021] provides more in-depth coverage of work in both directions. Model-based methods offer the benefits of good safety control and interpretability of robot behavior, but they adapt poorly to complex, crowded real-world scenarios where pedestrian behavior is too diverse to be covered comprehensively by the proposed models. Learning-based models have the ability to abstract pedestrian behavior, but current methods are trained only in simulation due to a lack of large-scale datasets. The simulators used in training these models are often unrealistic in terms of how pedestrian movements are simulated. If pedestrians are realistically simulated in simulation, we can copy their behavior model as the social navigation solution to a real-world robot.

In this thesis, we propose leveraging pedestrian groups to address navigation in dense human crowds. Humans subconsciously behave and perceive others as groups when in crowds [Koffka, 1935]. Therefore, using group-based representations inherently respects the social norm of grouping and reduces the likelihood of intruding into pedestrian groups. In addition, dense crowds pose perception challenges for a mobile robot as the robot is unable to detect everyone in the environment due to occlusions. By abstracting sensor detection into groups, we can potentially bypass the challenge of identifying individual pedestrians in a computationally efficient manner. Pedestrian group formulation is a level of model-based abstraction to simplify inter-group interactions. We further integrate our group-based representations into a Model Predictive Control framework for better safety control. However, to model group behavior and inter-group interactions, we use learning-based models to capture and predict future states of the groups.

Parallel to the work on group-based navigation, we also address the lack of large-scale labeled pedestrian datasets in the learning-based community. We also identify the need for large-scale datasets to better model group-based behavior and interactions. We propose a data collection infrastructure that allows efficient labeling of large quantities of pedestrian data. With it, we also

present a larger-scale dataset, the TBD Pedestrian Dataset. Our dataset contains many unique characteristics and is much larger than prior similar datasets.

1.1 Challenges

First, we explain in greater detail the challenges that hinder current mobile robots' ability to achieve full social navigation in pedestrian environments, particularly in crowded scenarios. This thesis contributes work addressing the following observed challenges.

1.1.1 Social Challenges of Social Navigation

Many social challenges exist in this domain. To support the top-level goal of abiding by social norms while maintaining safety and efficiency, mobile robots need to reason about and perform actions that minimize disturbances to pedestrian behavior. However, there is limited knowledge on systematic modeling of pedestrian behavior.

In this thesis, we focus on the challenge of understanding and preserving inter-pedestrian relationships. In a crowded environment, pedestrians form varying levels of relationships with other pedestrians, ranging from strangers who temporarily share common navigation goals to intimate partners. At the proxemics level, these relationships inform the social spaces among pedestrians. Mobile robots that intrude on such social spaces are often perceived as rude and unsocial. On a side note, mobile robots can also form implicit navigational relationships with nearby pedestrians to better communicate intentions and produce predictable navigation actions.

1.1.2 Practical Challenges of Social Navigation

From a practical perspective, we observe two complexities associated with real-world mobile robot implementations.

The first complexity involves limitations in perception. Mobile robots take inputs from noisy onboard sensors such as sonars, LiDARs, and RGBD cameras to perceive humans [Chatterjee and Steinfeld, 2016]. These sensors, despite the potential of state-of-the-art deep learning models [Wu et al., 2019, Redmon and Farhadi, 2018], often fail to produce identification and localization of pedestrians with reliability guarantees. Even if the sensors are noise-free, occlusions are commonplace in crowds from the perspective of a ground-level mobile robot. Additionally, unlike autonomous driving vehicles, mobile service robots are often smaller in profile, and their operating

environment often demands closer interaction with pedestrians. As a result, their perception of pedestrians can be partial, observing only the leg and waist areas of nearby pedestrians.

The second complexity originates from current limitations in planning. In crowded environments, conservative estimates of pedestrian dynamics and future state predictions sometimes block off all available navigation paths between the robot and its goal. With no available paths to proceed in its planning space, the robot is unable to make a progressive navigation decision, i.e. it encounters the *freezing robot problem*. A natural solution is to consider navigation strategies that coordinate with pedestrians. However, explicit forms of communication (e.g. verbal communication) with nearby pedestrians are undesirable because they distract pedestrians, may induce discomfort, and do not scale when considering large groups of robots and people. They are also inefficient because mobile robots make navigation decisions frequently. Therefore, in this thesis, we solely focus on the robot's motions with humans, or implicit interactions.

1.1.3 Background Challenges of Social Navigation

Learning-based methods have gained great popularity in the social navigation community. Many works focus on end-to-end learning, through reinforcement learning [Chen et al., 2017, Chen et al., 2019, Liu et al., 2021, Kästner et al., 2020] or inverse reinforcement learning [Tai et al., 2018, Okal and Arras, 2016]. Other works focus on the use of learning-based models to address perception challenges, such as pedestrian trajectory forecasting [Alahi et al., 2016, Gupta et al., 2018], group detection [Taylor et al., 2020] or eye gaze [Belkada et al., 2021]. For a dataset to be useful for social navigation, it needs to be collected in pedestrian-rich environments and contain grounded labels in the metric space. So far, datasets that satisfy these two criteria are small, with ETH [Pellegrini et al., 2009] and UCY [Lerner et al., 2007] still the mainstream datasets used in social navigation research. Some datasets such as ATC [Brščić et al., 2013] and SCAND [Karnan et al., 2022] use automated methods to label pedestrians or do not label pedestrians. Although these datasets are valuable large-scale datasets, the social navigation research community largely utilizes simulation or computer vision task-related datasets as alternatives.

Social navigation simulation tools have also developed rapidly. Notable recent social navigation simulators include [Biswas et al., 2021, Tsoi et al., 2020, Kästner et al., 2022]. However, the problem of modeling pedestrian behavior realistically in simulators is a paradox: if the pedestrians behave realistically in simulators, then the same behavior model can be carried over to the real world, and social navigation will be solved.

1.2 Contributions

In a step towards addressing these challenges, this thesis proposes leveraging pedestrian groups in support of robot social navigation and showcases a data collection system for large-scale pedestrian datasets.

The use of pedestrian groups is psychologically intuitive. From a perception perspective, we argue that human perceptions of crowds are short-term and do not scale with crowd size. Instead of tracking and predicting pedestrians individually, humans group other pedestrians with similar motion characteristics together. This observation is in line with the psychological process known as Gestalt [Koffka, 1935], where organisms observe the formation of entities rather than individual components. From a behavioral perspective, we similarly argue that humans do not expend significant energy in navigation planning in crowds. Instead of carefully zigzagging through gaps among pedestrians, humans often conform to the behavior of neighboring pedestrians with similar short-term goals and form pedestrian groups.

On the perception level, this thesis proposes a novel group-based obstacle representation of pedestrians. Leveraging pedestrian groups in this way directly offers the benefit of preventing the robot from intruding into social spaces formed by inter-pedestrian relationships. Pedestrians often navigate in groups when they share social ties with each other. Intrusion into such pedestrian groups is often perceived as rude and unsocial. Even without social ties, cutting through pedestrian groups can cause disturbances to the pedestrians navigating within the groups.

From a practical standpoint, our group-based obstacle representation can also be generated from noisy sensor inputs, such as point clouds, instead of outputs from pedestrian detection and localization models. In addition, by focusing on the edges of the groups that are closest to the robot, group-based representations can potentially cover occluded pedestrians [Chatterjee and Steinfeld, 2016]. We also show that group-based representations, when integrated with a traditional planner on a mobile robot, result in fewer group intrusions. This may lead to safer and more socially appropriate navigation behavior for the robot.

On the navigation level, we observe that pedestrian groups, as an intermediate representation, not only help to prevent social group intrusions, but also close off the tiny gaps that exist among pedestrians. It is dangerous for the robot to navigate using these tiny gaps because it leaves little margin for error in controls or uncertainty in perception. As shown in chapter 4, the robot's overall behavior becomes more polite and safe.

In parallel to the proposed group-based navigation system, we address the background challenge in social navigation by designing a scalable pedestrian data collection system. The system

consists of portable hardware powered by batteries, so that data can be collected at any location theoretically. The semi-autonomous labeling pipeline coupled with our error-checking tools can produce large amounts of grounded, human-verified labels quickly. With this system, we collect a large-scale pedestrian dataset that supports social navigation research. Our dataset also contains some unique characteristics such as a combination of top-down views and ego-centric views and the use of a suitcase robot that is pulled by humans to capture naturalistic pedestrian interactions with the robot.

1.3 Outline

An outline of the thesis projects is as follows.

- **Chapter 3:** We introduce our initial pipeline to generate group-based representations. We show that by observing the transformation of social group shapes generated in this way, we can successfully predict social group splits and merges. We also show that such a task may be successfully performed on social group shapes generated both from pedestrians and from simulated laser scan points.
- **Chapter 4:** We incorporate the group-based representations generated by the pipeline in Chapter 3 into a model predictive control (MPC)-based planner. We show that by leveraging group-based representations and future state predictions, the mobile robot produces safer and more social behavior in simulation.
- **Chapter 5:** We propose to modify our group generation pipeline by focusing on *the visible edge* of the groups. This modification greatly decreases the computation time to generate group-based representations and inference prediction models while maintaining similar performance.
- **Chapter 7:** Due to the small scales of the pedestrian datasets available, we do not see large quantities of pedestrian groups present in these datasets. We describe our first set of data collection efforts, which include a portable hardware setup and a semi-autonomous labeling system to facilitate the data collection process.
- **Chapter 8:** We introduce our second set of data collection efforts, which include major improvements on hardware, the labeling pipeline, and the addition of an error checking tool that enables fast label production. Using the updated system, we create a much larger dataset.

CHAPTER 2

PROBLEM DEFINITION

For the purposes of this work, we use the problem definition from [Mavrogiannis et al., 2021]. Navigation is the task of following a collision-free, efficient route from an initial location to a destination. In general, navigation has been studied as a hierarchical planning problem composed of *global planning* and *local planning*. Given a static environment, a global planner is designed to find a sequence of waypoints to reach a destination. Given a route or a set of waypoints found by a global planner, a local planner aims to navigate safely to the next waypoint. In this thesis, we focus on the local planning level.

We define social navigation as a navigation task in a dynamic human environment where each agent has both hidden and visible objectives. In this thesis, we do not factor in visible objectives that can be captured via explicit pedestrian communications such as verbal communication or gestures. Instead, we only focus on capturing implicit pedestrian objectives via their motion trajectories. The uncertainty induced by the lack of knowledge of explicit parameters introduces the need for the incorporation of prediction mechanisms over the behaviors of pedestrians but also the need for frequent re-planning to ensure adaptation to the dynamic environment.

Definition 2.0.1 (Social Navigation)**.** *Consider a robot navigating in a workspace $\mathcal{W} \subseteq \mathbb{R}^2$ amongst n other dynamic agents. Denote by $s \in \mathcal{W}$ the state of the robot and by $s^i \in \mathcal{W}$ the state of agent $i \in \mathcal{N} = \{1, \dots, n\}$. The robot navigates from a state s_0 towards a destination s_T by executing a policy $\pi : \mathcal{W}^{n+1} \times \mathcal{U} \to \mathcal{U}$ that maps the world state $\boldsymbol{S} = s \cup_{i=1:n} s^i$ to a control action $u \in \mathcal{U}$, drawn from a space of controls $\mathcal{U} \subseteq \mathbb{R}^2$. We assume that the robot is not aware of agents' destinations s_T^i or policies $\pi_i : \mathcal{W}^{n+1} \times \mathcal{U}^i \to \mathcal{U}^i$, $i \in \mathcal{N}$. Our goal is to design a policy π that enables the robot to navigate from s_0 to s_T safely and socially.*

In case of unknown agent states due to imperfect perceptions, $s^i \in \mathcal{W}$ represents the states of sensor input (instead of agents) from n observations $i \in \mathcal{N} = \{1, \dots, n\}$.

Part II

Group Based Representations and Predictions

CHAPTER 3

GROUP SPLIT AND MERGE

This chapter describes the work in Wang and Steinfeld [2020].

3.1 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* [Trautman and Krause, 2010] or result in displays of confusion or unpredictability. Humans may perceive these and other nonsocial behaviors as rude or dangerous [Ljungblad et al., 2012, Mutlu and Forlizzi, 2008].

In this project, we try to predict social group splits and merges. Our analysis of social groups is inspired by a human perceptual process known as Gestalt [Gadol, 1981], 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 [Chatterjee and Steinfeld, 2016]. Likewise, previous work suggests that individuals within such a group may have social ties among them [Moussaïd et al., 2010], making it inappropriate for a robot to plan a path that cuts through the group. Predicting splits and merges is also important because it may lead to more efficient robot navigation. Knowing when and where a split or merge will occur allows path planning toward a split point or preemptive avoidance of merging groups. This planning consideration can form more natural navigation paths, increasing trust and acceptance from humans.

We formulate our group dynamics analysis as a video event prediction and localization problem. As shown in Figure 3.1, we attempt to predict if a split or merge will occur 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 [Liang and Srikant, 2017], 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 3.1. The model needs to rely on subtle temporal

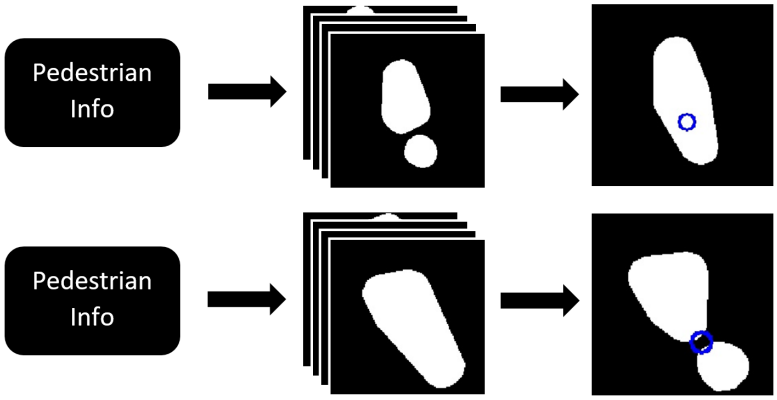

Figure 3.1: Predicting social group splits and merges offers navigation benefits for mobile robots. We first generate social group shape sequences from available pedestrian information. Then, we use our model to predict splits and merges. The blue circles represent the locations of splits and merges.

features in group shape transformations because it is impossible to predict splits and merges from a single image, unlike action recognition tasks [Ibrahim et al., 2016, Ji et al., 2013, Shu et al., 2017]. 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 [Chatterjee and Steinfeld, 2016]. This makes our task different from tasks that require precise tracking of pedestrians, such as trajectory prediction tasks [Alahi et al., 2016, Gupta et al., 2018, Robicquet et al., 2016, Vemula et al., 2018]. 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 the tracking tasks [Henriques et al., 2011, Perera et al., 2006, Zhu et al., 2014].

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 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 [Tran et al., 2015] with demonstrated effectiveness;

3. Discussions of human behavioral implications from our model's learned features.

3.2 Related Work

3.2.1 Social Interaction Between Pedestrians

Traditionally, researchers have tried to understand pedestrian social behavior from a classical model-based point of view, drawing inspiration from topics in physics, such as fluid dynamics [Helbing, 1992] and potential energy [Cui et al., 2011]. The Social Force model approach by Helbing and Molnár [1995] employs attractive and repulsive forces to model pedestrian interactions. 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. Alahi et al. [2016] uses the SocialLSTM model to account for the actions of neighboring pedestrians. Vemula et al. [2018] uses Social Attention to bypass the assumption of the local neighborhood. Gupta et al. [2018] uses generative adversarial models with a global social pooling layer. More sophisticated models developed by Sadeghian et al. [2019] employ a multimodal approach by taking 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, Solera et al. [2017] proposed using structural support vector machine-based learning to model social groups, and Mihaylova et al. [2014] modeled grouping as a sequential Monte Carlo process. Additionally, the dynamics of grouping have been actively incorporated into pedestrian tracking problems. Perera et al. [2006] and Zhu et al. [2014] believe that social groups can be used to enhance the tracking of pedestrians, including social group merges and splits. Henriques et al. [2011] incorporated group splits and merges in an extended maximum-a-posteriori problem. Makris and Prieur [2014] 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, 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. Ibrahim et al. [2016] uses hierarchical recurrent networks to identify joint actions performed by athletes during a sporting event. Shu et al. [2017] and Shu et al. [2015]

utilize a similar concept on pedestrians to predict group activities, such as queueing or crossing the street. Similarly to 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.

3.2.2 Deep Learning of Videos

The Long-Short Term Memory (LSTM) network, a kind of Recurrent Neural Network (RNN), has shown recent success in sequential data analysis tasks. Some models use LSTMs to process the features produced by CNNs [Donahue et al., 2015, Wu et al., 2015]. Shi et al. [2015], Patraucean et al. [2015] proposed Convolutional LSTM (ConvLSTM) for video prediction tasks. However, Varol et al. [2018] suggested that any form of RNN breaks the video patches into short clips, likely leading to suboptimal 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 large numbers of images [Zeiler and Fergus, 2014]. Ji et al. [2013] proposed a 3D CNN that was similar to a traditional CNN, but used 3D kernels to jointly encode spatial-temporal features. Shortly after, Tran et al. [2015] created a C3D network to classify videos successfully. Since then, many networks based on 3D convolutions [Liu et al., 2016, Sun et al., 2015, Varol et al., 2018] have shown great performance in tasks such as action recognition. We believe that C3D's success lies in its ability to combine video frames as video patches.

3.3 Approach

Group shapes can be generated using arbitrary algorithms, assuming that the resulting group shapes reflect temporal patterns as 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 our model.

3.3.1 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 to each other share largely similar motion characteristics. Therefore, when a

Figure 3.2: 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.

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 that 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 Chatterjee and Steinfeld [2016], which uses the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm [Ester et al., 1996]. We apply DBSCAN three times, each time within the clusters from the previous DBSCAN iteration. We perform DBSCAN in the order of Θ_i, S_i, P_i, and obtain 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 Kirby [2010], shown in Figure 3.2. 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:

$$\sigma_f = \max(2s, 0.5), \sigma_s = \frac{2}{3}\sigma_f, \sigma_r = \frac{1}{2}\sigma_f \tag{3.1}$$

Then, according to Kirby [2010], each individual's social space can be defined by the following

equations:

$$L_e(\varphi) = \sqrt{\frac{C}{\cos^2 \gamma/(2\sigma_1) + \sin^2 \gamma/(2\sigma_2)}} \tag{3.2}$$

$$f_s(x, y, \theta, s) = \begin{pmatrix} x + \cos(\varphi + \theta) L_e(\varphi) \\ y + \sin(\varphi + \theta) L_e(\varphi) \end{pmatrix}, \quad \text{for } 0 < \varphi \leq 2\pi \tag{3.3}$$

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

Group Social Space. For each group membership j in the image frame V_i, we first obtain all 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 social group space for the group label j in the image frame V_i. Some examples of social group shapes are shown in Figure 3.1.

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 3.1, 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 $j \neq j_1 \neq j_2$.

3.3.2 Group Shapes from Simulated Laser Scan Points

Most robots will not have data representing overhead views where the full perimeter of convex hulls is 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 that each pedestrian is 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 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 3.3.1, we now have laser scan point information that can also be

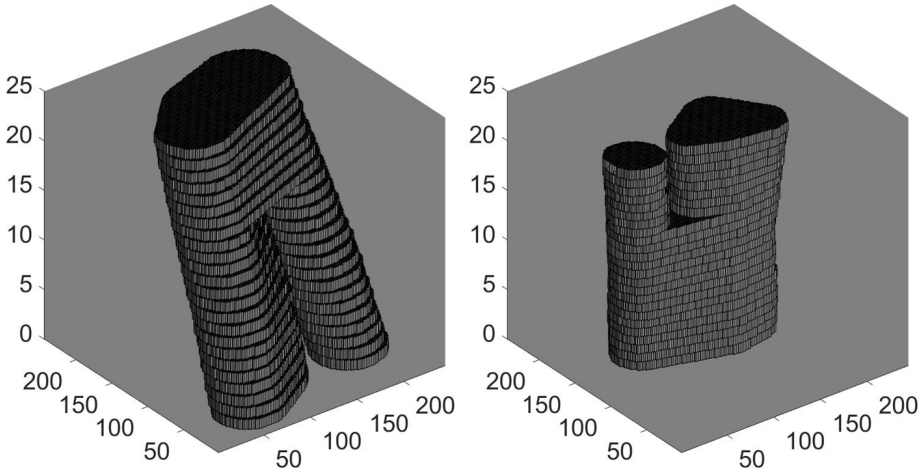

Figure 3.3: Volumetric features of a merge (left) and a split (right). Only features before the branch (the 17th layer) are visible to our model.

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 3.3.1, we apply DBSCAN on Θ_i, S_i, P_i to obtain 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 3.2, right), so we generate social spaces directly from laser scan points. As would occur with a real lidar, the "robot" sometimes fails to observe occluded pedestrians. We obtain the group space H_{ij} of the 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: $H_{ij} = \text{convexhull}\left(\{(x_{ik}, y_{ik}) | k \in G_{ij}\}\right)$. The splits and merges follow the same formulation as in Section 3.3.1.

3.3.3 3D Convolution Neural Network

The Third Spatial Dimension. Ji et al. [2013] and Tran et al. [2015] proposed 3D Convolutions based on the intuition that 3D kernels connect spatial features and temporal features together. Although Tran et al. [2015] provided examples of learned features, it is hard to distinguish whether these features belong to the spatial dimension or the temporal dimension, since even a single 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.

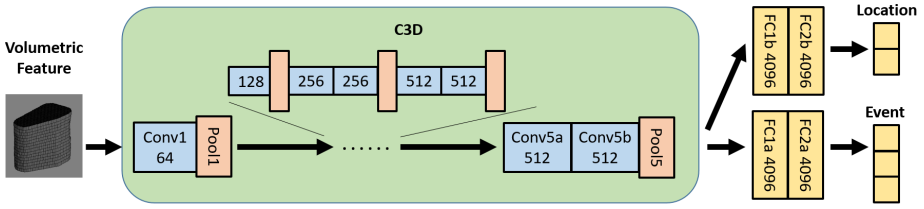

Figure 3.4: Our 3D convolutional neural network architecture. Each blue block represents a 3D convolution block with the number indicating the number of output channels. Each orange block represents a pooling block. Each yellow block represents a fully connected layer. The portion of our architecture within the green block is the same as C3D [Tran et al., 2015] before the fully connected layers. Sharing the `pool5` layer, two branches of fully connected layers predict split and merge occurrences and locations respectively.

As shown in Figure 3.3, splits and merges have unique volumetric features [Ke et al., 2005] in the temporal dimension. 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 predicting whether these 3D objects will evolve into tree branches.

The architecture. We use the C3D architecture shown in Figure 3.4 [Tran et al., 2015] as our architecture backbone, because it has demonstrated strong performance in action recognition tasks. A difference between 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, merge, and split. The second branch predicts 2D pixel coordinates $r = (r_x, r_y)$, 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 location of the ground truth as the midpoint of the shortest line that connects the two social group spaces involved as shown in Figure 3.1. In practice, the pursuit of the exact location accuracy of the split and merge 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 as follows:

$$\mathcal{L}(p, r, p_t, r_t) = \mathcal{L}_{cls}(p, p_t) + \lambda\delta(p_t)\mathcal{L}_{loc}(r, r_t) \tag{3.4}$$

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

$$\mathcal{L}_{cls}(p, p_t) = -\sum_{i=0}^{2} p_{ti} \log \left(\frac{e^{p_i}}{\sum_{j=0}^{2} e^{p_j}} \right) \tag{3.5}$$

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

$$\mathcal{L}_{loc}(r, r_t) = (r_{tx} - r_x)^2 + (r_{ty} - r_y)^2 \tag{3.6}$$

$\delta(p_t)$ 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:

$$\delta(p_t) = \begin{cases} 1, & \text{if } \arg\max_i(p_{ti}) \geq 1 \\ 0, & \text{otherwise} \end{cases} \tag{3.7}$$

The hyperparameter λ in Eq. (3.4) controls how much the location loss influences the overall loss.

3.4 Experiments

3.4.1 Setup

Our default experiment setting is very similar to Tran et al. [2015]. 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 in [Tran et al., 2015]. 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 our 16-frame inputs.

During training for all 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.

3.4.2 Datasets

Our raw dataset was a mixture of the publicly available ETH dataset [Pellegrini et al., 2009] and UCY dataset [Lerner et al., 2007]. These two datasets have been commonly used in pedestrian trajectory prediction problems [Alahi et al., 2016, Robicquet et al., 2016, Vemula et al., 2018, Yamaguchi et al., 2011]. Both datasets contain complex group interaction behaviors [Pellegrini

et al., 2009] and their videos are recorded at 25 FPS. The ETH dataset contains two sets of data from two scenes (ETH and HOTEL). The UCY dataset contains three sets of data from another two scenes (ZARA1, ZARA2, UCY1).

To generate training data for a merge instance, suppose that the merge takes place at time $i+1$ and that 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} following the procedures in Section 3.3. Next, we paste 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 are 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. For this, 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 centroids of the convex hulls 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 at an arbitrary angle. Also, due to the scarcity of the events, we did not perform evaluations on time horizons longer 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.

3.4.3 Evaluation

Due to the size of the dataset, we performed leave-one-out cross-validation. This evaluation approach has also been adopted by others [Alahi et al., 2016, Gupta et al., 2018, Zhang et al., 2019]. We trained our model on 4 sets of data and evaluated it on the remaining set.

Table 3.1: Comparison of our approach with Social-LSTM, Social-GAN, and SR-LSTM models (F1 score)

Method		S-LSTM		S-GAN		SR-LSTM		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	59.1

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 for a ground-based, real-world robot, we did not compare them with models that take local image patches as inputs (e.g., Sadeghian et al. [2019], Xue et al. [2018]). Therefore, we used Social-LSTM [Alahi et al., 2016] and Social-GAN [Gupta et al., 2018] as baselines, and SR-LSTM [Zhang et al., 2019] 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 original group memberships of the pedestrians. We then fed the pedestrian 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 allows 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 prior work [Alahi et al., 2016, Gupta et al., 2018, Zhang et al., 2019].

To allow location prediction accuracy comparisons, we also modified the trajectory-based models by applying our group shape generation pipeline to 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 3.1, but saw that our model regularly outperforms the others for all three metrics.

As shown in Table 3.1, our method was generally better than these techniques. Although designed for a different task, the Social-GAN and SR-LSTM models still outperformed the baseline Social-LSTM models. However, our approach outperformed 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 in 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 in equal numbers for 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 [Buda et al., 2018]. The issue of large data variety with limited amounts of data is an important area for future work.

Accuracy across parameters. Recall from Section 3.3 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 3.3.1. Varying these values also simulates how pedestrian behavior in social groups can vary greatly across cultures (e.g., Sorokowska et al. [2017]). 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 difficult scenario, because our model performed moderately on ETH and the worst on UCY1 as shown in Table 3.1. Then, we adjusted each grouping parameter to examine our model's adaptability.

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 3.4.2. This error can be significantly lower when the prediction is projected back to the raw image.

In Table 3.2, 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 in the corresponding parameter from the value used for the single frame prediction in Table 3.1. The values of average location errors are in parentheses. 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 3.3, we approximated a simulated laser scan for use in our model. This was done 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 compared to those generated from basic pedestrian informa-

Table 3.2: 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)			

tion. Training for this analysis was stopped once performance had reached levels comparable to our regular approach. As shown in Table 3.1, applying our model to simulated laser scans results in similar prediction accuracies, but location predictions are less accurate because the inputs are 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 in 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 [Zeiler and Fergus, 2014] 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 which parts of the input image contribute to these activation values (Figure 3.5). The results suggest that 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.

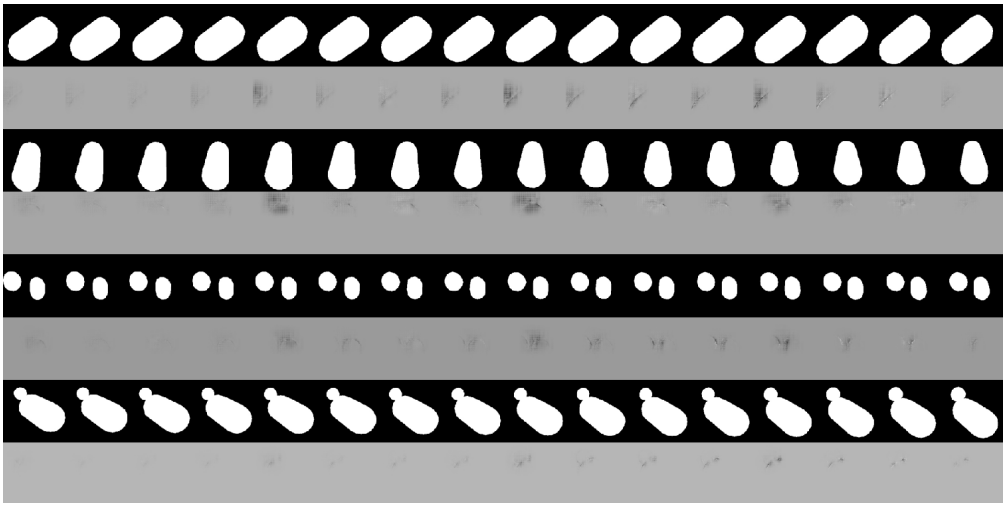

Figure 3.5: Visualization of our model using the method from Zeiler and Fergus [2014]. The first four rows are two inputs corresponding to two cases of splits and their learned contributions to the highest activation value in `conv5b`. The last four rows correspond to two cases of merges.

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 counterclockwise. 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 learned features shown in Figure 3.5, we can infer that our model captures the 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.

3.5 Conclusion

To improve robot navigation efficiency and social navigation near moving pedestrian groups, we present a 3D CNN model to predict 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 [Tran et al., 2015] since volumetric features [Ke et al., 2005] 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 project included secondary results that are valuable for 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 the 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.

Part III

Group Based Navigation

CHAPTER 4

GROUP-BASED REPRESENTATION WITH MODEL PREDICTIVE CONTROL

Work in this chapter is featured as a paper at CoRL2021 [Wang et al., 2022] (oral presentation).

4.1 Introduction

Over the past three decades, there has been vivid interest in the area of robot navigation in pedestrian environments [Thrun et al., 1999, Kruse et al., 2013, Kretzschmar et al., 2016, Trautman et al., 2015, Mavrogiannis et al., 2019]. Planning robot motion in such environments can be challenging due to the lack of rules regulating traffic, the close proximity of agents, and complex emerging multiagent interactions. In addition, accounting for human safety and comfort as well as robot efficiency add to the complexity of the problem.

To address such specifications, a common [Luber et al., 2012, Trautman et al., 2015, Kretzschmar et al., 2016, Kim and Pineau, 2016, Everett et al., 2018] paradigm involves the integration of a behavior prediction model into a planning mechanism. Recent models tend to predict individual interactions among agents to enable the robot to determine collision-free candidate paths [Kretzschmar et al., 2016, Trautman et al., 2015, Mavrogiannis et al., 2017]. While this paradigm is well-motivated, it tends to ignore the structure of interaction in such environments. Often, the motion of pedestrians is coupled as a result of social grouping. Furthermore, the motion of multiple agents can often be *effectively* grouped as a result of similarity in motion characteristics. Lacking a mechanism for understanding the emergence of this structure, the robot motion generation mechanism may yield unsafe or uncomfortable paths for human bystanders, often violating the space of social groups.

Motivated by such observations, we draw inspiration from human navigation to propose the use of group-based prediction for planning in crowd navigation domains. We argue that humans do not employ detailed individual trajectory prediction mechanisms. In fact, our motion prediction capabilities are short-term and do not scale with the number of agents. We do, however, employ effective grouping techniques that enable us to discover safe and efficient paths among motions of crowd networks. This anecdotal observation is aligned with Gestalt theory from psychology [Koffka, 1935] which suggests that organisms tend to perceive and process *formations of entities*, rather than individual components. Such techniques have recently led to advances in

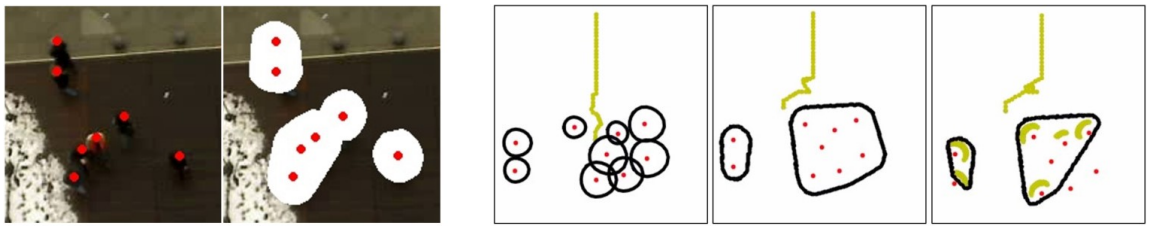

Figure 4.1: Based on a representation of social grouping [Wang and Steinfeld, 2020], we build a group behavior prediction model to empower a robot to perform safe and socially compliant navigation in crowded spaces. The images to the left demonstrate an example of our representation overlayed on top of a scene from a real-world dataset [Pellegrini et al., 2009]. The images to the right demonstrates that a model predictive controller equipped with our prediction model is able to navigate around the group socially (middle) as opposed to the baseline that cuts through the group (left). Our formulation is also able to handle imperfect state estimates (right) where the green arcs are scan points from a simulated 2D lidar laser scan.

computer vision [Desolneux et al., 2007] and computational photography [Vázquez and Steinfeld, 2011]. Similarly, we envision that a robot could reason about the formation of groups in a crowded environment and react to their motion as an effective way to navigate safely.

Here, we propose a group-based representation coupled with a encoder-decoder prediction model based on the group space approximation model of Wang and Steinfeld [2020]. This model groups a crowd into sets of agents with similar motion characteristics and draws geometric enclosures around them, given observation of their states. The prediction module then predicts the future states of these enclosures. We conduct an extensive empirical evaluation on 5 different pedestrian datasets [Pellegrini et al., 2009, Lerner et al., 2007], each with a flow following and a crossing scenario. We further conduct the same set of evaluations with agents powered by ORCA [van den Berg et al., 2011] that share the start and end locations in the datasets. Last but not least, we conduct evaluation given inputs in the form of simulated laser scans, from which pedestrians are only partially observable or even completely occluded. We compare the performance of our group-based formulation against three individual reasoning baselines: a) a reactive baseline without prediction; b) a constant velocity prediction baseline; and c) one based on individual S-GAN trajectory predictions [Gupta et al., 2018]. We present statistically significant evidence suggesting that agents powered by our formulation produce safer and more socially compliant behavior and are potentially capable of handling imperfect state estimates.

4.2 Related Work

In recent years, considerable research has been focused on the problem of robot navigation in crowded pedestrian environments [Trautman et al., 2015, Kretzschmar et al., 2016, Mavrogiannis et al., 2018, Everett et al., 2018, Chen et al., 2019]. Such environments often comprise groups of pedestrians, who navigate as coherent entities. This has motivated recent work on group detection and group motion modeling.

Groups are often perceived as sub-modular entities that collectively define the behavior of the crowd. Šochman and Hogg [2011] suggests that 50-70% of pedestrians walk in groups. Many works exist in group detection. One popular area in this domain is static group detection, often leveraging F-formation theories [Kendon, 1990]. However, dynamic groups often dominate pedestrian-rich environments and exhibit different spatial behavior [Yang and Peters, 2019]. Among dynamic group detection, the most common approach treats grouping as a probabilistic process where groups are a reflection of the close probabilistic association of pedestrian trajectories [Bazzani et al., 2012, Chang et al., 2011, Gennari and Hager, 2004, Pellegrini et al., 2010, Zanotto et al., 2012]. Others use graph models to build inter-pedestrian relationships with strong graphical connections that indicate groups [Chamveha et al., 2013, Khan et al., 2015]. The social force model [Helbing and Molnár, 1995] also inspires Mazzon et al. [2013], Šochman and Hogg [2011] to develop features that indicate groups. Clustering is another common technique for grouping pedestrians with similar features into groups [Solera et al., 2016, Ge et al., 2012, Taylor et al., 2020, Chatterjee and Steinfeld, 2016]. In this project, we do not intend to explore the state-of-the-art grouping practice. For our formulation, it is sufficient to employ a simple clustering-based grouping method proposed by Chatterjee and Steinfeld [2016].

Applications of groups often focus on a specific behavior aspect. In terms of interaction with pedestrians, a major focus in this area is how a robot should behave as part of group formation [Cuntoor et al., 2012]. In dyad groups involving a single human and a robot, some researchers examined socially appropriate following behavior [Gockley et al., 2007, Granata and Bidaud, 2012, Jung et al., 2012, Zender et al., 2007] and, conversely, guiding behavior [Nanavati et al., 2019, Feil-Seifer and Matarić, 2011, Pandey and Alami, 2009]. In works that do not include robots as part of pedestrian groups, some research teams studied how a robot should guide a group of pedestrians [Garrell and Sanfeliu, 2010, Shiomi et al., 2007, Martinez-Garcia et al., 2005]. From a navigation perspective, [Yang and Peters, 2019] leverage groups as obstacles, but their group space is a collection of individual personal spaces with occasional O-space modeling from F-formation theories. Without the engineered occasional occurrence of O-space, their representation reduces

to one of our baselines. [Katyal et al., 2020] introduce an additional cost term that leverages the robot's distance to the closest group in a reinforcement learning framework. They model groups using convex hulls directly generated from pedestrian coordinates instead of taking personal spaces into consideration. This less principled approach often leads to the robot approaching dangerously close to pedestrians. In this project, we additionally explore the capabilities of our grouping technique in handling imperfect sensor inputs. While our focus is on analyzing the benefits of using groups, our group-based formulation can be easily incorporated into Katyal et al. [Katyal et al., 2020]'s framework.

4.3 Group-based Prediction

We recap the framework for group-based representations introduced in Chapter 3 [Wang and Steinfeld, 2020], but with a different set of mathematical notations that better fit into our Model Predictive Control framework later introduced in Section 4.4. We then introduce a group-based prediction model that is suited for use in decentralized multi-agent navigation.

4.3.1 Group Representation

Define as $\theta^i \in [0, 2\pi)$ the orientation of agent $i \in \mathcal{N}$, which is assumed to be aligned with the direction of its velocity u^i extracted via finite differencing of its position over a timestep dt. Denote by $v^i = ||u^i|| \in \mathbb{R}^+$ its speed. We define an augmented state for agent i as $q^i = (s^i, \theta^i, v^i)$.

We treat a social group as a set of agents who are in close proximity and share similar motion characteristics. Assume that a set of J groups, $\mathcal{J} = \{1, \ldots, J\}$ navigate in a scene. Define by $g^i \in \mathcal{J}$ a variable indicating the group membership of agent i. We then define a group $j \in \mathcal{J}$ as a set $G^j = \{i \in \mathcal{N} \mid g^i = j\}$ and collect the set of all groups in a scene into a set $\boldsymbol{G} = \{G^j \mid j \in \mathcal{J}\}$.

Extracting Group Membership. We define the combined augmented state of all agents as $\boldsymbol{q} = \cup_{i=1:n} q^i$. To obtain group memberships for a set of agents $\mathcal{N}$, we apply the Density-Based Spatial Clustering of Applications with Noise algorithm (DBSCAN) [Ester et al., 1996] on agent states:

$$\boldsymbol{G} \longleftarrow \texttt{DBSCAN}(\boldsymbol{q} \mid \epsilon_s, \epsilon_\theta, \epsilon_v), \tag{4.1}$$

where $\epsilon_s, \epsilon_\theta, \epsilon_v$ are respectively threshold values on agent distances, orientation, and speeds for the clustering method.

Extracting the Social Group Space. For each group G^j, $j \in \mathcal{J}$, we define a *social group space* as a geometric enclosure $\mathcal{G}^j$ around the agents of the group. For each agent $i \in G^j$, we define

a personal space $\mathcal{P}^i$ as an asymmetric two-dimensional Gaussian based on the model introduced by Kirby [2010]. Refer to Section 3.3.1 for detailed descriptions.

Given the personal spaces $\mathcal{P}^i$, $i \in G^j$, of all agents in a group j, we extract the social group space of the whole group as a convex hull:

$$\mathcal{G}^j = \text{Convexhull}(\{\mathcal{P}^i \mid i \in G^j\}). \tag{4.2}$$

The shape described by $\mathcal{G}^j$ represents an obstacle space representation of a group containing agents in close proximity with similar motion characteristics. For convenience, we collect the spaces of all groups in a scene into a set $\boldsymbol{\mathcal{G}} = \{\mathcal{G}^j \mid j \in \mathcal{J}\}$.

4.3.2 Group Space Prediction Oracle

Based on the group-space representation of Sec. 4.3.1, we describe a prediction oracle that outputs an estimate of the future spaces occupied by a set of groups $\boldsymbol{\mathcal{G}}_{t:t_f}$ up to a time $t_f = t + f$, where f is a future horizon given a past sequence of group spaces $\boldsymbol{\mathcal{G}}_{t_h:t}$ from time $t_h = t - h$ where h is a window of past observations:

$$\boldsymbol{\mathcal{G}}_{t:t_f} \leftarrow \mathcal{O}(\boldsymbol{\mathcal{G}}_{t_h:t}) = \cup_{j=1:J} \mathcal{O}_j(\mathcal{G}^j_{t_h:t}), \tag{4.3}$$

where $\mathcal{O}_j$ is a model generating a group space prediction for group G^j.

We implement the oracle $\mathcal{O}_j$ of Eq. (4.3) using a simple encoder-decoder model. The encoder follows the 3D convolutional architecture in [Tran et al., 2015] and the decoder mirrors the model layout of the encoder. The encoder-decoder model takes as input a sequence[1] $\mathcal{G}_{t_h:t}$ and outputs a sequence $\mathcal{G}_{t+1:t_f}$ which we pass through a sigmoid layer. We supervise the encoder-decoder model's output using the binary cross-entropy (BCE) loss function:

$$L_{BCE} = \frac{1}{f} \sum_{k=t+1}^{t+f} \text{BCE}(\mathcal{G}_k, \mathcal{G}^*_k), \tag{4.4}$$

where $\mathcal{G}^{j*}_{t+1:t_f}$ denotes the ground truth group spaces in the image coordinates for a group.

We verify the effectiveness of our encoder-decoder model on the 5 scenes of our experiments by conducting a cross-validation comparison against a baseline. The baseline predicts the future shapes by linearly translating the last social group shape using its geometric center's velocity. We

[1]The oracle input sequence is first converted into image-space coordinates using the homography matrix of the scene. We also preprocess inputs to have normalized scale and group positions. The encoder-decoder model output is converted back into Cartesian coordinates using the inverse homography transform.

Table 4.1: Encoder-Decoder Model Performance

	Metric	ETH	HOTEL	ZARA1	ZARA2	UNIV
Baseline	mIoU (%)	83.52	90.37	88.04	89.30	85.32
	fIoU (%)	76.32	85.38	82.14	83.88	77.24
Ours	mIoU (%)	86.66	92.10	89.97	90.94	87.52
	fIoU (%)	78.64	86.83	83.77	85.09	78.55

use Intersection over Union (IoU) as our metric. Between the ground truths and the predictions, this metric divides the number of overlapped pixels by the number of pixels occupied by either of them. As shown in Table 4.1, our encoder-decoder model outperforms the baseline.

4.3.3 Partial Input Handling

Note that in a dynamic pedestrian scene, we will have frequent occurrences of partial inputs for individual agents or groups due to new agents entering the scene or new groups being formed, respectively. Therefore, our prediction model must be able to handle cases in which the input is complete up to a past window $t_{\hat{h}}$ with $t_{\hat{h}} = t - \hat{h}$, $\hat{h} < h$, i.e., $\boldsymbol{\mathcal{G}}_{t_{\hat{h}}:t}$. To handle these cases, for time $t_h < \tau < t_{\hat{h}}$, we compute $\mathcal{G}_\tau^j$ by making the following membership assumptions:

- For any agent $i \in G_t^j$ such that $i \notin G_\tau^j$ and for whom we have the complete state history $s_{t_h:t}^i$, we set $g_\tau^i = j$. In other words, the prior group membership of any recent members of group j is set to j (although agent i may be a member of another group j' in those instances).
- For any agent $i \in G_t^j$ such that $i \notin G_\tau^j$ and for whom we only have a *partial* state history $s_{t_{\hat{h}}:t}^i$, we take the agent's last known state s_t^i and velocity u_t^i and backpropagate it as $s_{\tau-1}^i = s_\tau^i - u_\tau^i dt$.

Given a small h, these assumptions should reflect a close approximation of the group's complete history state, because the pedestrian group switching process is gradual and pedestrian movements are smooth and predictive in short time windows.

4.4 Model Predictive Control with Group-based Prediction

We describe G-MPC, a group-prediction informed model predictive control (MPC) framework for navigation in multiagent environments that leverages the group-based prediction oracle of Sec. 4.3.

At planning time t, given a (possibly partial) augmented world state history $\boldsymbol{Q}_{t_{\hat{h}}:t}$, we first extract a sequence of group spaces $\boldsymbol{\mathcal{G}}_{t_h:t}$ based on the method of Sec. 4.3.1. Given these, the

robot computes an optimal control trajectory $\boldsymbol{u}^* = u^*_{1:K}$ of length K by solving the following optimization problem:

$$(\boldsymbol{s}^*, \boldsymbol{u}^*) = \arg\min_{u_{1:K}} \sum_{k=1:K}^{K} \gamma^k J(s_{k+1}, \boldsymbol{\mathcal{G}}_{k+1}, s_T) \tag{4.5}$$

$$s.t.\ \boldsymbol{\mathcal{G}}_{2-h:1} \leftarrow \boldsymbol{\mathcal{G}}_{t_h:t} \tag{4.6}$$

$$s_1 \leftarrow s_t \tag{4.7}$$

$$\boldsymbol{\mathcal{G}}_{k+1:k_f} = \mathcal{O}(\boldsymbol{\mathcal{G}}_{k_h:k}) \tag{4.8}$$

$$u_k \in \mathcal{U} \tag{4.9}$$

$$s_{k+1} = s_k + u_k \cdot dt, \tag{4.10}$$

where γ is the discount factor and J represents a cost function; eq. (4.6) initializes the group space history ($k = 2 - h$ is the timestep displaced by a horizon h in the past from the first MPC-internal timestep $k = 1$); eq. (4.7) initializes the robot state to the current robot state s_t; eq. (4.8) is an update rule that recursively generates a predicted future group sequence up to timestep $k_f = k+f$ given history from time $k_h = k-h$ up to time k; $\mathcal{O}$ represents a group-space prediction oracle based on Sec. 4.3; and eq. (4.10) is the robot state transition assuming a fixed time parameterization of step size dt.

We employ a weighted sum of costs J_g and J_d, penalizing respectively distance to the robot's goal and proximity to groups:

$$J(s_k, \boldsymbol{\mathcal{G}}_k, s_T) = \lambda J_g(s_k, s_T) + (1 - \lambda) J_d(s_k, \boldsymbol{\mathcal{G}}_k), \tag{4.11}$$

where λ is a weight representing the balance between the two costs and

$$J_g(s_k) = \begin{cases} 0, & \text{if } s_k \in \boldsymbol{\mathcal{G}}_k \\ ||s_{k-1} - s_T||, & \text{else,} \end{cases} \tag{4.12}$$

penalizes a rollout according to the distance of the last collision-free waypoint to the robot's goal. Further, we define J_d as:

$$J_d(s_k, \boldsymbol{\mathcal{G}}_k) = \exp(-\mathcal{D}\left(s_{k+1}, \boldsymbol{\mathcal{G}}_k\right)), \tag{4.13}$$

where

$$\mathcal{D}(s_k, \boldsymbol{\mathcal{G}}_k) = \begin{cases} \min_{j \in \mathcal{J}} D\left(s_k - \mathcal{G}_k^j\right), & \text{if } s_k \notin \mathcal{G}_k^j \\ -\min_{j \in \mathcal{J}} D\left(s_k - \mathcal{G}_k^j\right), & \text{else,} \end{cases} \tag{4.14}$$

where $D(s_k - \mathcal{G}_k^j)$ returns the minimum distance between the robot state and the space occupied by group j at time k. Using D, the function $\mathcal{D}$ computes the minimum distance to any group for a given time. In most cases, the robot lies outside of groups, that is, $s_k \notin \mathcal{G}_k^j$ —- therefore, the cost J_d tries to maximize the distance $\mathcal{D}$. Sometimes, the robot might end up entering the group space $\boldsymbol{\mathcal{G}}$ –in those cases, J_d tries to minimize $\mathcal{D}$, to steer the robot towards the direction of quickest escape from the group. If the robot is inside a group to begin with, we shrink the group sizes in Sec. 4.3.1 until the robot is outside the groups again.

To solve Eq. (4.5), we search over a finite set $\boldsymbol{\mathcal{U}}$ of control trajectories of horizon K. With the assumption that the robot is holonomic and is not under any kinematic constraints, we use a set of R control rollouts $\boldsymbol{\mathcal{U}} = \{\boldsymbol{u}^1, ..., \boldsymbol{u}^R\}$ with three levels of tangential speeds and a fixed number set of turning speeds, i.e.,

$$u_{1:K}^r = (v\cos\psi, v\sin\psi, \omega),\ \psi = \frac{2\pi r}{R}, v \in \left\{\frac{1}{3}v_{max}, \frac{2}{3}v_{max}, v_{max}\right\}, \omega \in \left\{0, \pm\frac{\pi}{2}\right\} \tag{4.15}$$

To ensure compatibility between our group-based prediction model and our MPC formulation, we set the control rollout time horizon to be the prediction model's prediction horizon, or $K = f$.

4.5 Evaluation

We evaluate our framework through a simulation study in which the robot performs a navigation task (a transition between two points) within a crowd of dynamic agents in a set of scenes.

4.5.1 Experimental Setup

We consider a set of realistic pedestrian scenes drawn from the ETH [Pellegrini et al., 2009] (ETH and HOTEL scenes) and UCY [Lerner et al., 2007] (ZARA1, ZARA2, and UNIVERSITY scenes) datasets, which often serve as benchmarking testbeds in the motion prediction and social navigation literature [Alahi et al., 2016, Gupta et al., 2018, Zhang et al., 2019, Cao et al., 2019]. In each scene, we define two navigation tasks (see Fig. 4.2): *Flow*, in which the robot navigates along the crowd flow, and *Cross*, in which the robot intersects vertically with the traffic flow. For each task, we generate a set of trials by segmenting the scene recording into blocks involving challenging interactions. We define a challenging interaction as a segment involving at least 5 pedestrians inside the test region drawn in black in Fig. 4.2. This process provides us with a distribution of trials as shown in table Table 4.2. Across all trials, we keep the robot's maximum at $1.75m/s$ and use a fixed timestep size $dt = 0.1$.

Table 4.2: Number of trials per task and scene.

Task	ETH	HOTEL	ZARA1	ZARA2	UNIV
Flow	58	43	25	127	106
Cross	58	44	28	129	114

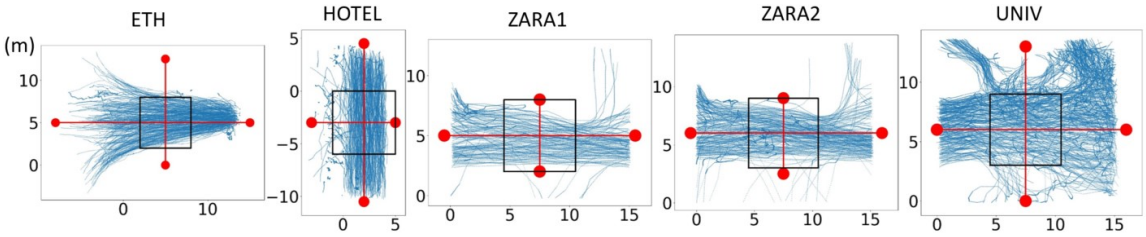

Figure 4.2: Trajectories of all pedestrians in the datasets. The red dots represent the task start and end locations. The red lines represent the task paths. The black box represents the test region to check for non-trivial tasks.

We consider two experimental conditions: *Offline* and *Online*. In the *Offline* one, the robot navigates among a crowd moving according to a recording of a human crowd. Under this condition, pedestrians act as dynamic obstacles that do not react to the robot, a situation which could arise in cases where robots are shorter and could thus be easily missed by navigating pedestrians. In the *Online* one, the robot navigates among a crowd[2] moving by running ORCA [van den Berg et al., 2011], a policy that is frequently used as a simulation engine for benchmarking in the social navigation literature [Cao et al., 2019, Everett et al., 2018, Mavrogiannis et al., 2021].

To investigate the value of G-MPC, we develop three variants of it. **group-auto** is a G-MPC in which the encoder-decoder model has a history $h = 8$ and a horizon $f = 8$. **group-nopred** is a variant that features no prediction at all–it just reacts to observed groups at every timesteps and is equivalent to the framework of Yang and Peters [2019]. Finally, **laser-group-auto** is identical to **group-auto** but instead of using ground truth pose information, it takes as input noisy lidar scan readings. We simulate this by modeling pedestrians as $1m$-diameter circles and lidar scans as rays projecting from the robot. We refer to the spec sheet of a SICK LMS511 2D lidar for simulation parameters. We further inject noise into the readings according to the spec sheet. In this simulation, pedestrians may only be partially observable or even completely occluded from the robot.

We compare the performance of these policies against a set of MPC variants using differing

[2]For consistency, the agents in the crowd start and end at the same spots as the agents in the recorded crowd from the Offline condition.

mechanisms for individual motion prediction. **ped-nopred** is a vanilla MPC that reacts to the current states of other agents without making predictions about their future states. **ped-linear** is a vanilla MPC that estimates future states of agents by propagating agents' current velocities forward. This baseline is motivated by recent work showing that constant-velocity models yield competitive performance in pedestrian motion prediction tasks [Schöller et al., 2020]. Finally, **ped-sgan** is an MPC that uses S-GAN [Gupta et al., 2018] to extract a sequence of future state predictions for agents based on inputs from their past states. We selected S-GAN because it is a recent high-performing model.

We measure the performance of the policies with respect to four different metrics: a) *Success rate*, defined as the ratio of successful trials over the total number of trials; b) *Comfort*, defined as the ratio of trials in which the robot does not enter any social group space over the total number of trials; c) *Minimum distance to pedestrians*, defined as the smallest distance between the robot and any agent observed over each trial; d) *Path length*, defined as the total distance traversed by the robot in a trial.

We define a trial as successful if the robot never collides with a pedestrian and reaches the goal within a time limit. Whenever the robot is within $0.5m$ of any pedestrian, we say that a collision occurs. Additionally, we define the time limit to be three times the time required for the robot to reach its goal by following a straight line without any surrounding pedestrians.

To track the performance of G-MPC, we design a set of hypotheses targeting aspects of safety and group space violation, which we investigate under both experimental conditions, i.e., offline and online:

H1: To explore the benefits of group-based representations alone, we hypothesize that **group-nopred** is safer than **ped-nopred** while achieving similar success rates but worse efficiency.

H2: To explore the full benefit of the group-based formulation, we hypothesize that **group-auto** is safer than **ped-linear** and **ped-sgan** while achieving similar success rates but worse efficiency.

H3: To explore how our formulation handles imperfect input, we hypothesize that **laser-group-auto** achieves similar safety to **group-auto** while achieving similar success rates and efficiency.

H4: To check that our formulation is socially compliant, we hypothesize that **group-nopred**, **group-auto** and **laser-group-auto** violate agents' group space less often than the baselines.

4.5.2 Encoder-Decoder Model Details

Our encoder-decoder model largely leverages [50]'s C3D network. As shown in Fig. 4.3, the encoder architecture contains the following layers (beginning from the input layer): one $3 \times 3 \times 3$

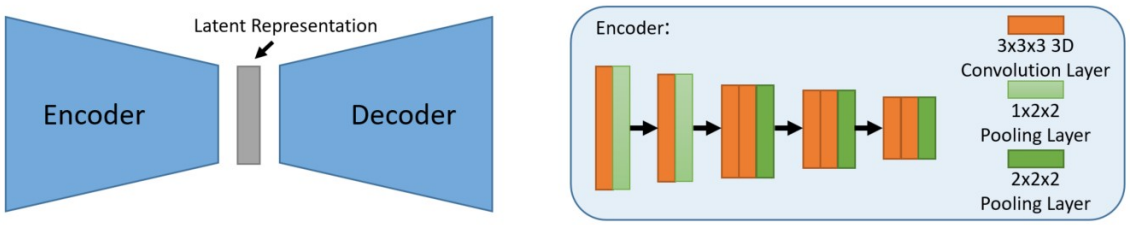

Figure 4.3: Our simple encoder-decoder model's architecture. The decoder's deconvolution layers mirror the layout of the encoder.

convolution layer with 64 channels, one $1 \times 2 \times 2$ maxpool layer, one $3 \times 3 \times 3$ convolution layer with 128 channels, another $1 \times 2 \times 2$ maxpool layer, another $3 \times 3 \times 3$ convolution layer with 128 channels, one $3 \times 3 \times 3$ convolution layer with 256 channels, one $2 \times 2 \times 2$ maxpool layer, another $3 \times 3 \times 3$ convolution layer with 256 channels, one $3 \times 3 \times 3$ convolution layer with 512 channels, another $2 \times 2 \times 2$ maxpool layer, two $3 \times 3 \times 3$ convolution layers with 512 channels, and another $2 \times 2 \times 2$ maxpool layer.

We used an initial learning rate of $1e-5$ and a batch size of 1 and trained for 200 epochs. We used the Adam optimizer with default PyTorch settings. The data samples are generated by sampling a random segment during the evolution of a group for all groups in all the datasets. The data samples are normalized in scale and position so that the entire group space sequence fits inside the 224×224 image sequences and the geometric center of the group in the last input sequence frame is at the center of the image. After obtaining the predictions from the model, we filter out pixel predictions with a confidence level lower than 0.5. An example comparison between the ground truth and the preidcted sequence is shown in Fig. 4.4. For evaluation on each dataset scene, including both evaluation of the encoder-decoder model's performance and the policies in the navigation setting, we use a model that was trained on the other four datasets.

In simulated laser scan settings, we do not retrain the group shape prediction models. Instead, we transfer the learned group shape prediction models on perfect perception settings directly into this new setting. We use a nearest neighbor approach based on geometric centers to identify the history sequence of a group in order to predict the group's future states. If the nearest neighbor of a group in the previous frame is more than $0.25m$ away, then we say that no prior history of this group is available and use the technique in Section 4.3.3 to linearly back-propagate the group's history.

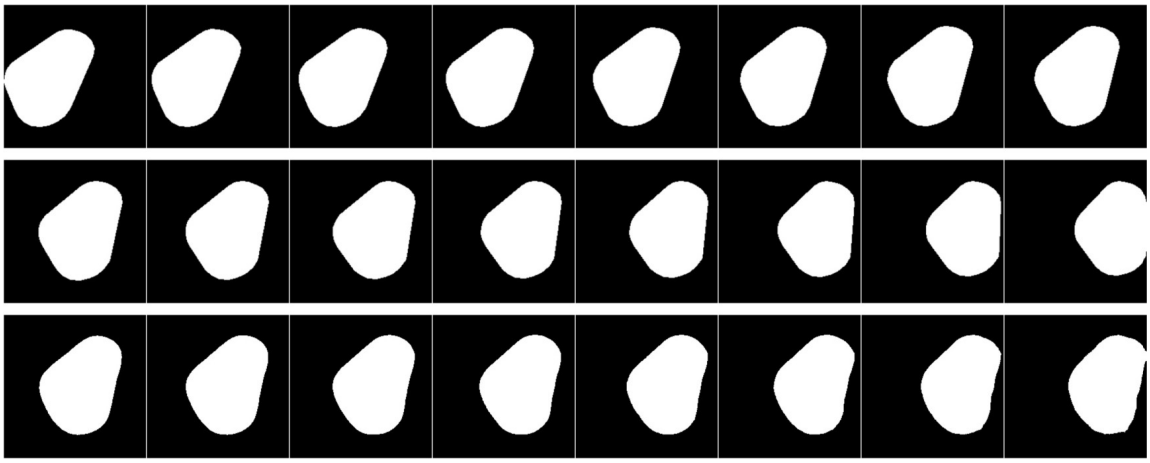

Figure 4.4: Top: An example group space input sequence for our encoder-decoder model. Mid: The ground truth future sequence of the group. Bottom: The predicted future sequence of the group as output by our encoder-decoder model.

4.5.3 Parameter Details

For the parameters of Eq. 4.1, we picked $\epsilon_s, \epsilon_\theta, and \epsilon_v$ so that the grouping results match our qualitative inspection of human grouping in the datasets, similarly to our prior project [Wang and Steinfeld, 2020]. For ETH, HOTEL, ZARA1 and ZARA2 we set $\epsilon_s = 2.0m, \epsilon_\theta = 30°, and \epsilon_v = 1.0m/s$. Because UNIV is more crowded than the other four datasets, group formations are tighter, and we set $\epsilon_s = 1.5m, \epsilon_\theta = 15°, and \epsilon_v = 0.5m/s$.

For the personal space constants of Eq. (3.2), we selected C under the assumption that closely interacting pedestrians walk around the boundaries of each other's personal space. For ETH, HOTEL, ZARA1 and ZARA2, we set $C = 0.35$. Again, because UNIV has denser crowds, we set $C = 0.25$. If at any given time the robot enters a social group space, we incrementally reduce C by 0.1 with a minimum value of 0.05 until the robot is outside the group space.

For the time horizon parameter f and the history window parameter h from Section 4.2, we set $f = 8$ and $h = 8$ to ensure our MPC formulation's compatibility with the SGAN models.

For the weight parameter λ in the cost function of Eq. 4.11, we perform a complete parameter sweep to tune λ. We tested λ with values from 0.1 to 0.9 with increments of 0.05 on 100 randomly sampled test cases. We then select a λ that results in high success rates (at least 90%) for both agent-based and group-based policies without predictions where the success rates of the two policies are the closest to each other. For trials with nonreactive agents, we set $\lambda = 0.65$. For trials with reactive agents, we set $\lambda = 0.3$. Note that we want the weight parameter to be the same for both pedestrian-

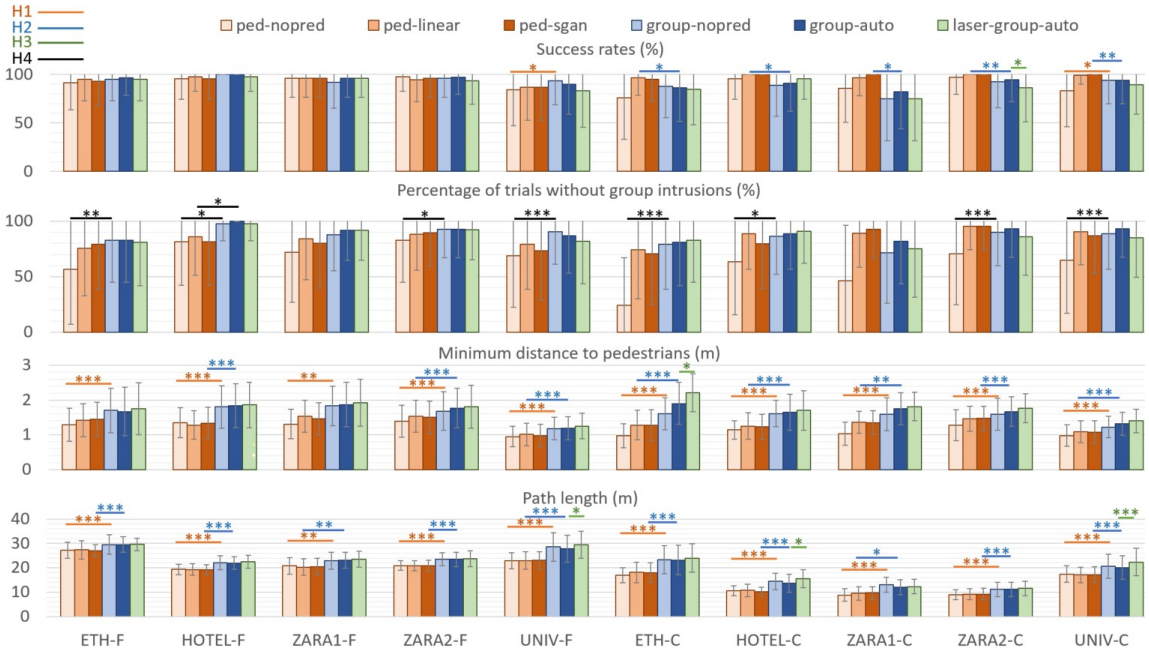

Figure 4.5: Performance per scene under the *Offline* condition. Horizontal lines indicate statistically significant results corresponding to different hypotheses.

based and group-based policies because the distance from the pedestrians to the boundaries of the social space are the same in both settings. Keeping the same weight allows fair evaluations of these two types of policies.

For the number of control rollouts R in Eq. 4.15, we set $R = 12$.

4.5.4 Results

4.5.4.1 Quantitative Analysis.

Fig. 4.5 and Fig. 4.6 contain bar charts representing the performance of G-MPC compared with its baselines under Offline and Online settings respectively. Bars indicate means, error bars indicate standard deviations, "F" and "C" are flow and cross scenarios respectively, and the number of asterisks indicates increasing significance levels: $\alpha = 0.05, 0.01, 0.001$ according to two-sided Mann-Whitney U-tests.

H1: We can see from both Fig. 4.5 and Fig. 4.6 that G-MPC achieves statistically significantly larger minimum distances to pedestrians across all scenarios, often with $p < 0.001$. This illustrates that the group representation is in itself capable of upgrading a simple MPC with no prediction. As

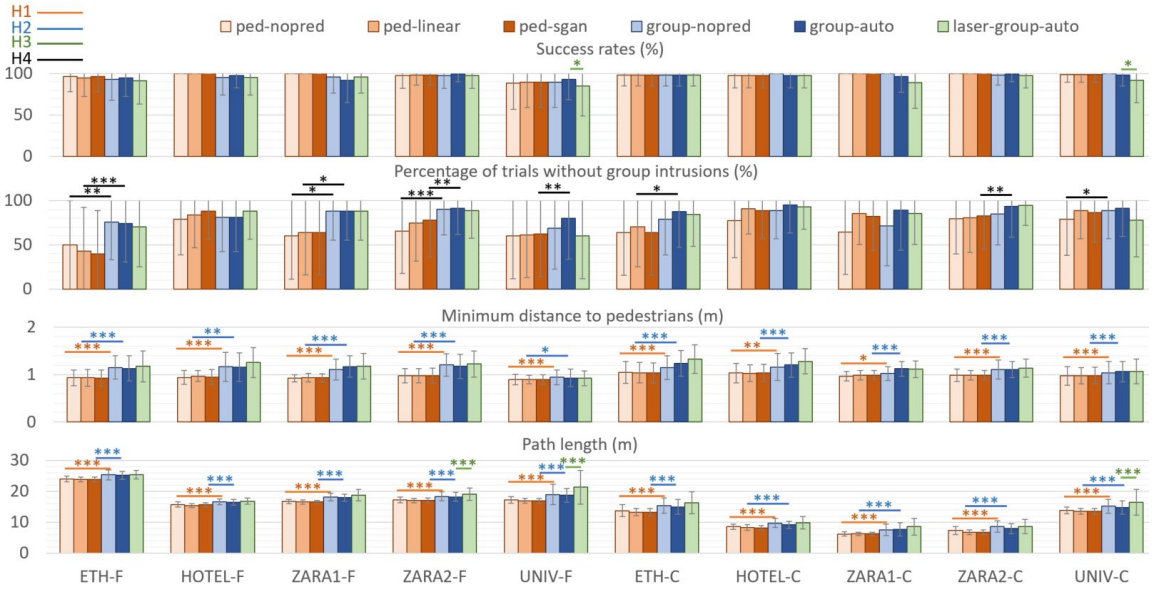

Figure 4.6: Performance per scene under the *Online* condition (simulated pedestrians powered by ORCA [van den Berg et al., 2011]). Horizontal lines indicate statistically significant results corresponding to different hypotheses.

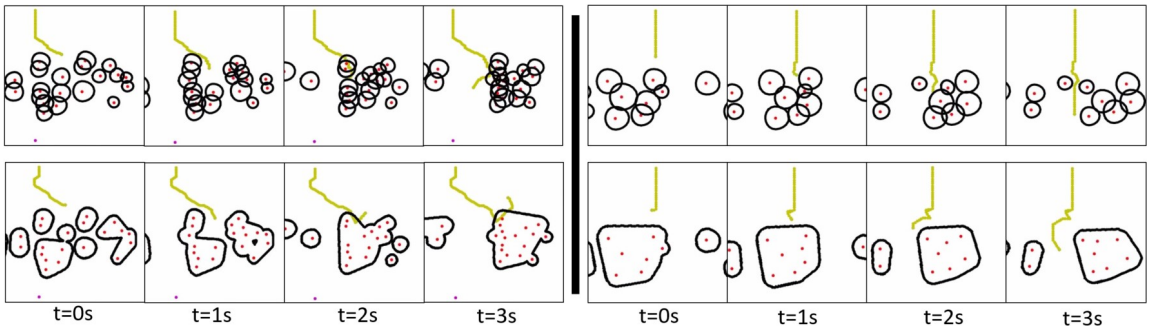

Figure 4.7: Qualitative performance difference between approaches leveraging pedestrian-based (top) and group-based (bottom) representations. Left: non-reactive agents. Right: reactive agents.

expected, we observe that the tradeoff G-MPC pays for increased safety is a larger average path length. We also see that success rates are comparable. Overall, we conclude that H1 holds.

H2: When future state predictions are considered, G-MPC obtains statistically significant results in most scenes that support its attributes of being safer at the cost of worse efficiency. Thus, H2 is partially confirmed. In offline scenarios, G-MPC has lower success rates in crossing scenarios. Upon closer inspection, most failure cases are due to timeouts from the conservative behavior of G-MPC. However, in online scenarios where pedestrians react to the robot, G-MPC achieves high success rates. In real-world situations, to cross dense traffic, the robot needs to plan its actions with the expectations of reactive pedestrians. Otherwise, the robot will probably face the *freezing robot problem* [Trautman et al., 2015].

H3: Overall, we observe that with simulated imperfect states, G-MPC does not perform statistically significantly worse in terms of safety, but in dense crowds of the UNIV scenes, it has worse efficiency and worse success rates in online cases. This shows that H3 holds in terms of safety and, in moderately dense human crowds, holds in terms of efficiency. Future work on better group representation is needed to achieve better efficiency in high-density human crowds given imperfect states.

H4: From Fig. 4.5 and Fig. 4.6, we can see that G-MPC often has fewer group space intrusions than its baselines. Although this relationship is not always statistically significant, we do see a general trend of group-based approaches to respect group spaces more often than individual ones. Therefore, we conclude that H4 is partially confirmed.

4.5.4.2 Numeric Results of Figures

Tab. 4.3 and Tab. 4.4 are the numerical results of Fig. 4 and Fig. 5. $\mathcal{S}$ is the success rate. $\mathcal{C}$ is the percentage of trials in which the robot does not enter any group space (collisions also count as group intrusions). $\mathcal{D}$ is the average minimum distance to pedestrians. $\mathcal{L}$ is the average path length.

4.5.4.3 Qualitative Analysis.

Qualitatively, it is more common for regular MPCs to perform aggressive and socially inappropriate maneuvers than G-MPC. As shown in the two examples in Fig. 4.7 executed by **ped-sgan** and **group-auto** agents, we can see that under offline conditions, the MPC agent aggressively cuts in front of the two pedestrians to the left before proceeding headlong into the cluster of pedestrians, only managing to avoid the deadlock by escaping through the narrow gap that opens up. In contrast, G-MPCtracks the movements of the two pedestrian groups that come from the left. When the two

Table 4.3: Performance per scene under the *Offline* condition.

Scene		ETH		HOTEL		ZARA1		ZARA2		UNIV	
Task	Metric	**Flow**	**Cross**	**Flow**	**Cross**	**Flow**	**Cross**	**Flow**	**Cross**	**Flow**	**Cross**
ped-nopred	$\mathcal{S}(\%)$	91.38	75.86	95.35	95.45	96.00	85.71	97.64	96.90	83.96	83.33
	$\mathcal{C}(\%)$	56.9	24.14	81.4	63.64	72.0	46.43	82.68	70.54	68.87	64.91
	$\mathcal{D}(m)$	1.29	0.97	1.35	1.14	1.31	1.03	1.39	1.27	0.95	0.98
	$\mathcal{L}(m)$	27.16	16.93	19.31	10.55	20.75	8.80	20.87	8.92	22.96	17.38
ped-linear	$\mathcal{S}(\%)$	94.83	96.55	97.67	100	96	96.43	94.49	100	86.79	99.12
	$\mathcal{C}(\%)$	75.86	74.14	86.05	88.64	84.0	89.29	88.19	95.35	79.25	90.35
	$\mathcal{D}(m)$	1.42	1.28	1.28	1.25	1.53	1.36	1.53	1.46	1.01	1.09
	$\mathcal{L}(m)$	27.31	18.08	19.28	10.72	20.29	9.48	20.84	9.16	22.93	17.06
ped-sgan	$\mathcal{S}(\%)$	93.1	94.83	95.35	100	96	100	96.06	100	86.79	100
	$\mathcal{C}(\%)$	79.31	70.69	81.4	79.55	80.0	92.86	89.76	95.35	73.58	86.84
	$\mathcal{D}(m)$	1.45	1.27	1.34	1.23	1.46	1.35	1.50	1.47	0.98	1.08
	$\mathcal{L}(m)$	27.05	17.99	19.20	10.10	20.51	9.66	20.83	9.21	23.05	17.22
group-nopred	$\mathcal{S}(\%)$	94.83	87.93	100	88.64	92	75	96.06	92.25	93.4	93.86
	$\mathcal{C}(\%)$	82.76	79.31	97.67	86.36	88.0	71.43	92.91	89.92	90.57	88.6
	$\mathcal{D}(m)$	1.70	1.61	1.8	1.61	1.83	1.59	1.68	1.59	1.18	1.22
	$\mathcal{L}(m)$	29.52	23.32	21.98	14.38	22.86	13.02	23.47	11.17	28.57	20.61
group-auto	$\mathcal{S}(\%)$	96.55	86.21	100	90.91	96	82.14	96.85	94.57	89.62	93.86
	$\mathcal{C}(\%)$	82.76	81.03	100.0	88.64	92.0	82.14	92.91	93.02	86.79	92.98
	$\mathcal{D}(m)$	1.67	1.90	1.83	1.65	1.87	1.75	1.77	1.67	1.19	1.32
	$\mathcal{L}(m)$	29.51	23.17	21.88	13.63	23.01	11.95	23.45	11.13	27.82	20.06
laser-group-auto	$\mathcal{S}(\%)$	94.83	84.48	97.67	95.54	96	75	93.7	86.05	83.02	89.47
	$\mathcal{C}(\%)$	81.03	82.76	97.67	90.91	92.0	75.0	92.13	86.05	82.08	85.09
	$\mathcal{D}(m)$	1.75	2.21	1.86	1.70	1.92	1.81	1.8	1.76	1.25	1.40
	$\mathcal{L}(m)$	29.57	23.99	22.42	15.45	23.50	12.26	23.63	11.58	29.49	22.36

pedestrian groups merge, the agent turns around and re-evaluates its approach to cross. In the online condition, we observe that the MPC agent cuts through a pedestrian group to reach the other side, forcing a member of the group to stop and yield as indicated by the pedestrian's shrinking personal space which is proportional to its speed. In the same situation, the G-MPC agent chooses to cross behind the social group.

4.6 Conclusion

We introduced a methodology for generating group-based representations and predicting their future states. Through an extensive evaluation over the flow and crossing scenarios drawn from 10 different real-world scenes from 2 different human datasets with both reactive and nonreactive agents, we demonstrate that our approach is safer and more socially compliant. Through exper-

Table 4.4: Performance per scene under the *Online* condition (simulated pedestrians powered by ORCA [van den Berg et al., 2011]).

Scene		ETH		HOTEL		ZARA1		ZARA2		UNIV	
Task	Metric	**Flow**	**Cross**	**Flow**	**Cross**	**Flow**	**Cross**	**Flow**	**Cross**	**Flow**	**Cross**
ped-nopred	$\mathcal{S}(\%)$	96.55	98.28	100	97.73	100	100	97.64	100	88.68	99.12
	$\mathcal{C}(\%)$	50.0	63.79	79.07	77.27	60.0	64.29	65.35	79.84	60.38	78.95
	$\mathcal{D}(m)$	0.93	1.05	0.94	1.04	0.92	0.97	0.98	0.99	0.89	0.98
	$\mathcal{L}(m)$	24.02	13.73	15.75	8.50	16.71	6.13	17.16	7.32	17.20	13.82
ped-linear	$\mathcal{S}(\%)$	94.83	98.28	100	97.73	100	100	98.43	100	89.62	99.12
	$\mathcal{C}(\%)$	43.1	70.69	83.72	90.91	64.0	85.71	74.8	80.62	61.32	88.6
	$\mathcal{D}(m)$	0.94	1.04	0.97	1.03	0.94	0.99	0.98	0.99	0.90	0.98
	$\mathcal{L}(m)$	23.83	13.25	15.43	8.32	16.54	6.22	17.03	6.74	16.87	13.53
ped-sgan	$\mathcal{S}(\%)$	96.55	98.28	100	97.73	100	100	98.43	100	89.62	99.12
	$\mathcal{C}(\%)$	39.66	63.79	88.37	88.64	64.0	82.14	77.95	82.95	62.26	86.84
	$\mathcal{D}(m)$	0.93	1.04	0.95	1.04	0.94	0.99	0.98	0.99	0.90	0.98
	$\mathcal{L}(m)$	23.85	13.20	15.63	8.14	16.54	6.18	17.06	6.72	16.90	13.53
group-nopred	$\mathcal{S}(\%)$	93.1	98.28	95.35	100	96	100	97.64	98.45	89.62	100
	$\mathcal{C}(\%)$	75.86	79.31	81.4	88.64	88.0	71.43	90.55	85.27	68.87	88.6
	$\mathcal{D}(m)$	1.15	1.15	1.17	1.16	1.11	1.03	1.21	1.11	0.94	1.04
	$\mathcal{L}(m)$	25.36	15.37	16.62	9.72	18.16	7.50	18.36	8.56	18.98	15.15
group-auto	$\mathcal{S}(\%)$	94.83	98.28	97.67	97.73	92	96.43	99.21	99.22	93.4	98.25
	$\mathcal{C}(\%)$	74.14	87.93	81.4	95.45	88.0	89.29	91.34	93.8	80.19	91.23
	$\mathcal{D}(m)$	1.13	1.24	1.16	1.21	1.17	1.13	1.18	1.11	0.93	1.07
	$\mathcal{L}(m)$	25.19	14.99	16.45	9.14	17.93	7.62	18.22	7.88	18.71	14.74
laser-group-auto	$\mathcal{S}(\%)$	91.38	98.28	95.35	97.73	96	89.29	97.64	97.67	84.91	92.11
	$\mathcal{C}(\%)$	70.69	84.48	88.37	93.18	88.0	85.71	88.98	94.57	60.38	78.07
	$\mathcal{D}(m)$	1.18	1.33	1.26	1.28	1.18	1.12	1.23	1.14	0.93	1.07
	$\mathcal{L}(m)$	25.40	16.27	16.81	9.82	18.72	8.54	19.07	8.57	21.39	16.46

imentation with simulated laser scans, our model displays promising potential to process noisy sensor input without much performance downgrade.

Various improvements to our control framework are possible. For example, we could incorporate state-of-the-art oracles in the form of advanced video prediction models [Guen and Thome, 2020]. Furthermore, additional considerations such as the set of rollouts or the cost functions used could possibly increase performance. Finally, alternative control frameworks such as reinforcement learning approaches could be applicable. However, our goal was to illustrate the value of group-based representations for navigation tasks.

CHAPTER 5

EDGE-BASED GROUP REPRESENTATION WITH MODEL PREDICTIVE CONTROL

5.1 Introduction

In Chapter 4, we discovered that the use of group-based representations allows a basic MPC controller-based robot to perform safer and more social navigation behavior. Newer work such as [Katyal et al., 2020] also supports this. However, there are still limitations in this project that hinder the ability to deploy group-based representation in real-world robots. To address these, we present improvements to prior group-based representations.

Our current definition of group space is in the shape of convex hulls of intricate personal spaces [Kirby, 2010]. These convex hulls contain too many vertices, often more than the number of pedestrians within the corresponding groups. Therefore, it is more time consuming to evaluate MPC rollout costs with respect to the group space vertices than with individuals. Additionally, it is difficult to run group prediction models directly with the vertices, which vary in number from group to group. Instead, we have to convert the vertices, which are in metric coordinates, into image space, as shown in Section 4.3.2. We draw groups on empty canvases and use a video

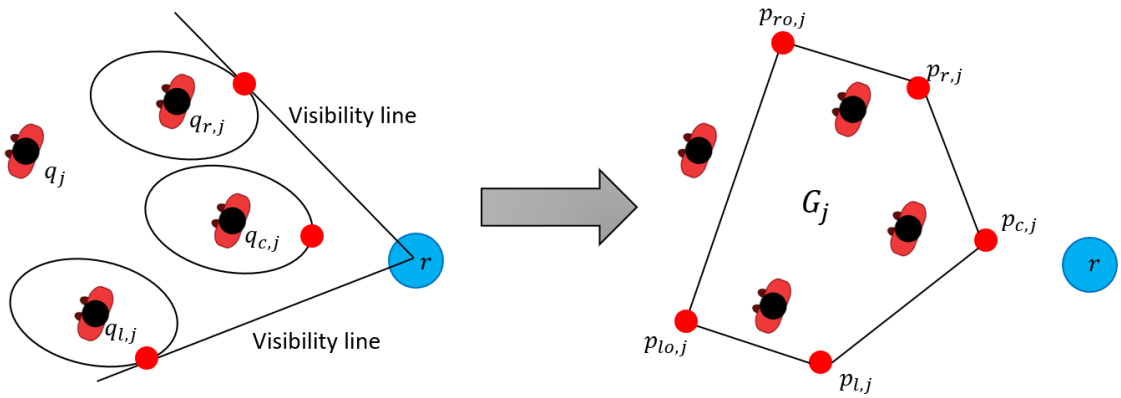

Figure 5.1: An illustration of our proposed new group space definition. The blue circles are the robot. We propose to leverage the visible edges of the groups to build a simplified group space representation. We also add offsets to the back of the visible edges to account for occlusions. The pentagon shape on the right is the resulting simplified social group space.

prediction-based encoder-decoder model to train and predict future group states. The predictions are then converted back into metric space.

With this technique, a great amount of computation is invested in the conversion processes between metric space and image space. The second problem associated with image space conversion is that occasionally the predicted group space can extend beyond the image boundary on the last predicted frame. Because there is no information about the group space outside the image boundary, a cut-off artifact can occur when the group space is converted back to the metric space. Last but most importantly, image sequence prediction models typically contain larger amounts of parameters than models that work with point trajectories (e.g. trajectory prediction models). As is the case with the encoder-decoder model in Chapter 4, the 3D convolution-based models require many 3D convolution model parameters to achieve sufficient prediction accuracy. This results in a long inference time.

To overcome these challenges, we present a new group representation based on simple point coordinates inspired by Chatterjee and Steinfeld [2016], as shown in Figure 5.1. However, the approach presented in this project is different from the leading edges in Chatterjee and Steinfeld [2016], where leading edges are defined using splines and are intractable to compute using simple mathematical equations. Our new representation is scalable so that conversion into image space will be unnecessary, and a trajectory-based encoder-decoder model instead of image sequence-based models can be used to make predictions. Our initial observation from Chapter 4 was that accurate group space size and location predictions matter more than predictions on their geometry details. Therefore, we hypothesize that a simplified group space representation will suffice when integrated into an MPC-based control framework, as long as the representation conveys the location and size of the group. In our extensive simulation evaluation, we show that with our new representation, we can achieve a much faster computation time in representation generation and prediction model inference, while maintaining similar levels of navigation performance.

5.2 Method

5.2.1 Visible-Edge-Based Group Representation

We follow the notation from Section 4.3.1. As a quick recap, we define an augmented state for agent i as $q^i = (s^i, \theta^i, v^i)$. We also define a group $j \in \mathcal{J}$ as a set $G^j = \{i \in \mathcal{N} \mid g^i = j\}$ and collect the set of all the groups in a scene into a set $\boldsymbol{G} = \{G^j \mid j \in \mathcal{J}\}$. We use the same grouping method, DBSCAN [Ester et al., 1996], with the same parameters to assign group memberships to

agents.

For each group G^j, $j \in \mathcal{J}$, we define a *simplified social group space* as a geometric enclosure $\mathcal{G}^j$ around the agents of the group based on the visible edge of the group. The visible edge of the group is the edge of the group that is closest to the robot in the scene.

As shown in Figure 5.1, we first define the visible edge of the group by identifying three key points: the point closest to the robot $p_{c,j}$; the point that is the leftmost visible point from the robot's perspective $p_{l,j}$; and the point that is the rightmost visible point from the robot's perspective $p_{r,j}$. To obtain $p_{c,j}$, $p_{l,j}$, and $p_{r,j}$, we first find the closest agent of the group $q_{c,j}$, the leftmost visible agent of the group $q_{l,j}$, and the rightmost visible agent of the group $q_{r,j}$. We then apply the egg-shaped personal space model $\mathcal{P}$ from Section 3.3.1 and establish a set of boundary points for each of the three agents $\mathcal{P}_{c,j}$, $\mathcal{P}_{l,j}$ and $\mathcal{P}_{r,j}$. Finally, $p_{c,j}$ is the closest point to the robot among $\mathcal{P}_c$. $p_{l,j}$ and $p_{r,j}$ are the leftmost and rightmost visible points from the robot's perspective among $\mathcal{P}_{l,j}$ and $\mathcal{P}_{r,j}$ respectively. After the three key points are identified, the visible edge consists of the two connecting line segments $(p_{c,j}, p_{l,j})$ and $(p_{c,j}, p_{r,j})$.

The method we used to identify $p_{c,j}$, $p_{l,j}$, and $p_{r,j}$ from $q_{c,j}$, $q_{l,j}$, and $q_{r,j}$ is partially supported by Theorem 5.2.1. Because $q_{c,j}$, $q_{l,j}$, and $q_{r,j}$ all belong to the same group, there are only small variations in their speed and orientation, and their personal spaces have a similar size and orientation. The egg-shaped personal spaces are reasonably similar to circles, so we can use this method to obtain the key points with negligible errors. This means that we can skip the need to generate personal spaces for all agents in the same group, and we do not need to obtain the convexhull group space first.

Theorem 5.2.1. *For all agents q_j who belong to group G^j, assume that they all have the same orientation θ^j and velocity v^j and their personal spaces $\mathcal{P}_j$ are all circles. If an agent q_x is the agent closest / leftmost / rightmost to the robot r, then the closest / leftmost / rightmost point p_x in its personal space $\mathcal{P}_x$ is the closest/leftmost/rightmost point to the robot among all the points from the convex hull social group space $\mathcal{G}^j$ defined in Section 7.3.1.*

Proof. We can prove this by contradiction. Simple mathematical deductions can show that if there exists another agent q_y who is not the closest / leftmost / rightmost agent among the group to the robot, its closest / leftmost / rightmost point in its personal space cannot be closer / more left / more right than that of q_x and this point will not be on the convex hull $\mathcal{G}^j$. □

Next, to account for occlusions, we define a fixed offset behind the visible edge. Given $p_{l,j}$ and $p_{r,j}$, we draw a rectangle of width d away from the robot and obtain the two offset points $p_{lo,j}$ and

$p_{ro,j}$. We set d to 1 meter to account for occlusions that are close to the visible edge. We rely on the regeneration of group space to find new visible edges and offsets of the group for future timesteps if more occluded parts of the group emerge.

With this, our updated group space is now a pentagon.

$$\mathcal{G}^j = \text{Pentagon}(p_{c,j}, p_{l,j}, p_{r,j}, p_{lo,j}, p_{ro,j}). \tag{5.1}$$

We collect the spaces of all groups in a scene into a set $\boldsymbol{\mathcal{G}} = \{\mathcal{G}^j \mid j \in \mathcal{J}\}$.

5.2.2 Group Space Prediction Oracle

Our social group space is now only defined by five points $p_{c,j}, p_{l,j}, p_{r,j}, p_{lo,j}, p_{ro,j}$. Only three of them $(p_{c,j}, p_{l,j}, p_{r,j})$ need to be tracked. Compared to our original convex hull representations that have to be converted into image space, our new representation is much more tractable.

Similar to Section 4.3.2, we collect information from time $t_h = t - h$ and make predictions up to time $t_f = t + f$. Both h and f are set to 8, just like in the previous chapter. We collect a history of the trajectories for $p_{c,j}, p_{l,j}, p_{r,j}$ and obtain $\tau_{c,j} = p^{c,j}_{t_h:t}, \tau_{l,j} = p^{l,j}_{t_h:t}, \tau_{r,j} = p^{r,j}_{t_h:t}$. After that, we do this for every group $\boldsymbol{\mathcal{T}}_{t_h:t} = \{\tau_{c,j}, \tau_{l,j}, \tau_{r,j} \mid j \in \mathcal{J}\}$. We use Social-GAN [Gupta et al., 2018], a popular trajectory prediction prediction model, to predict future trajectories.

$$\boldsymbol{\mathcal{T}}_{t:t_f} = \text{SGAN}(\boldsymbol{\mathcal{T}}_{t_h:t}) \tag{5.2}$$

We then obtain $p^{c,j}_{t:t_f}, p^{l,j}_{t:t_f}, p^{r,j}_{t:t_f}$ from $\boldsymbol{\mathcal{T}}_{t:t_f}$. Following the procedures in the previous section, from these predicted key points we then calculate the offset points $p^{lo,j}_{t:t_f}, p^{ro,j}_{t:t_f}$. Finally, we obtain the predicted future simplified social group space $\mathcal{G}^j_{t:t_f}$ using equation 5.1.

We chose Social-GAN because it is a well-established trajectory prediction model. Further, we need a trajectory prediction model that only requires positional trajectories as input. Models such as Sophie [Sadeghian et al., 2019] or Y-net [Mangalam et al., 2021] require image patches or semantic segmentation as additional input and are not suitable for us. There may be better-performing trajectory prediction models such as Trajectron++ [Salzmann et al., 2020] or AgentFormer [Yuan et al., 2021], but [Poddar et al., 2023] has shown that better trajectory prediction models only offer marginal benefits when integrated into a navigation system.

5.2.3 Integration into MPC

Integration of the group representation based on visible edges into MPC follows the same steps as in Section 4.4. The only differences are in how we define the distance function $D(s_k - \mathcal{G}^j_k)$ and how

we determine whether the robot is not inside a group $s_k \notin \mathcal{G}_k^j$ in the cost function (eq. 4.14). The visible edge-based group spaces are pentagons defined by five line segments, so instead of using sampled group boundaries, we can directly apply geometric formulas to calculate the elements in this cost function.

5.3 Evaluation

5.3.1 Evaluation Setup

We follow the exact same simulation evaluation setup as in Section 4.5.1 of the previous chapter. We name our new MPC framework with the simplified visual edge-based group representation **edge-sgan**. Our goal is to show that **edge-sgan** is computationally faster than **group-auto** with navigation performance similar to **group-auto**. To measure performance levels, we use *Success Rate*, *Minimum Distance to Pedestrians*, and *Path Lengths* as described in Section 4.5.1.

We also design two hypotheses to test our claims.

H1: To explore the computation benefits of simplified group space representations, we hypothesize that **edge-sgan** is much faster than **group-auto** in terms of computation time.

H2: To check that integrating simplified group space representations does not significantly affect navigation performance, we hypothesize that **edge-sgan** achieves similar *Success Rate*, *Minimum Distance to Pedestrians*, and *Path Lengths* as **group-auto**.

5.3.2 Results

Similarly to the analysis performed in Section 4.5.4, we compiled our results into bar charts (Fig. 5.2, Fig. 5.3, Fig. 5.4, and Fig. 5.5) that compare the results between **group-auto** and **edge-sgan** on four different metrics. Bars indicate means, error bars indicate standard deviations, "F" and "C" are flow and cross scenarios respectively, and the number of asterisks indicates increasing significance levels: $\alpha = 0.05, 0.01, 0.001$ according to two-sided Mann-Whitney U-tests.

H1: From Fig. 5.2, although the asterisks indicating statistical significance levels are not shown, we can clearly see that **edge-sgan** performs about ten times faster than **group-auto** in UNIV scenarios and about five times faster in other less crowded scenarios. As mentioned in Section 5.2, our visual edge-based representation does not require conversion into the image space, and the trajectory prediction models that work with these representations have many fewer parameters than image sequence prediction models. Both factors contribute to the significantly reduced computation time. Computation times are longest in UNIV scenarios due to the high crowd densities.

Figure 5.2: Performance in terms of computation time per step.

Figure 5.3: Performance in terms of success rates. Horizontal lines indicate statistically significant results corresponding to different hypotheses.

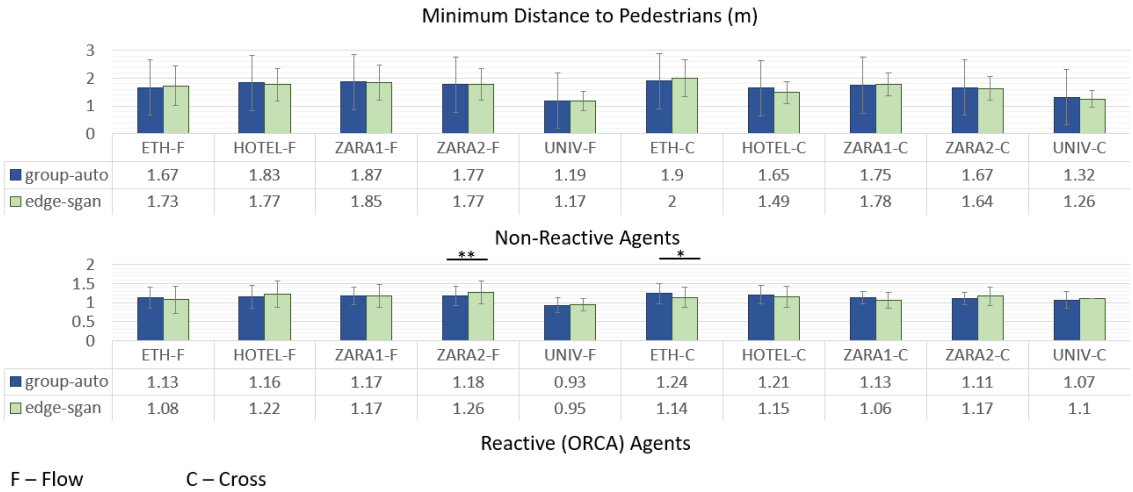

Figure 5.4: Performance in terms of minimum distance to pedestrians. Horizontal lines indicate statistically significant results corresponding to different hypotheses.

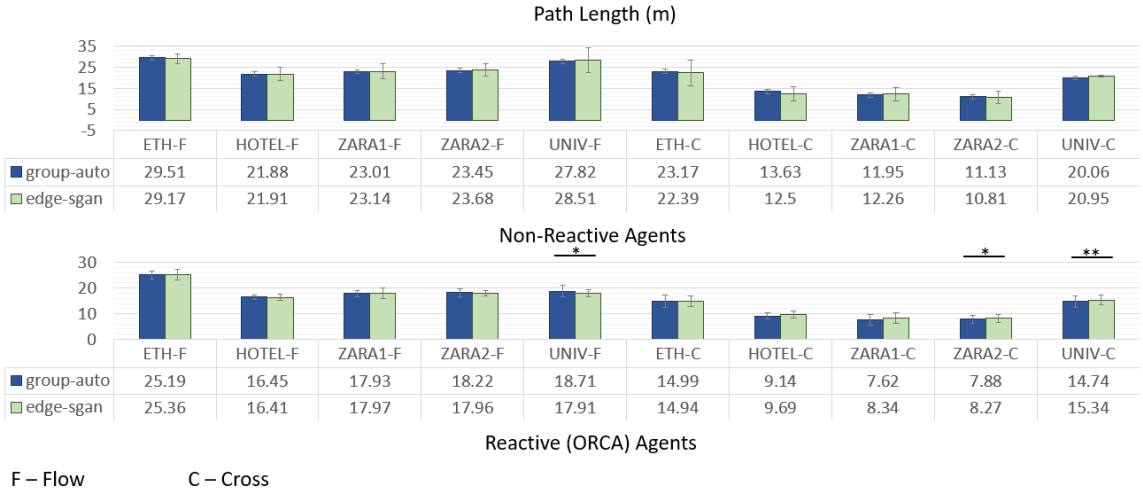

Figure 5.5: Performance in terms of path lengths. Horizontal lines indicate statistically significant results corresponding to different hypotheses.

H2: From Fig. 5.3, Fig. 5.4, and Fig. 5.5, we can see that **edge-sgan** and **group-auto** perform similarly in terms of success rates, minimum distance to pedestrians, and path lengths. Many of the comparisons do not yield statistically significant results, which indicate that both methods perform similarly. Interestingly, we do observe occasional statistically significant comparisons in the **Online** scenarios where simulation agents react to the robot, but no conclusions can be made from these observations, because in some cases **edge-sgan** performs better while in other cases **group-auto** performs better.

Overall, we conclude that both hypotheses hold.

5.4 Conclusion

We proposed a method of defining visual edges and generating simplified group space representations based on these visual edges. This formulation skips the need for image space conversions. We additionally adopted a trajectory prediction model that contains many fewer parameters than our image sequence predictor model as our oracle. In a setting similar to Section 4.5.1 from the previous chapter, we show that with these improvements, we are able to achieve much faster computation speeds while maintaining similar levels of navigation performance.

Many improvements can still be applied to our framework. As suggested in Section 4.6, the same possible MPC improvements are applicable and it is still possible to switch to a reinforcement learning-based framework. Although unlikely as suggested by [Poddar et al., 2023], better trajectory prediction models such as [Salzmann et al., 2020, Yuan et al., 2021] can be explored and tested to see if they offer significant improvements in navigation performance.

Part IV

Naturalistic Pedestrian Data Collection

CHAPTER 6

PROJECT INTRODUCTION

6.1 Introduction

Pedestrian datasets are essential tools for modeling socially appropriate robot behaviors, recognizing and predicting human actions, and studying pedestrian behavior. Researchers may use these data to predict future pedestrian motions, including forecasting their trajectories [Alahi et al., 2016, Gupta et al., 2018, Ivanovic and Pavone, 2019] and/or navigation goals [Kitani et al., 2012, Liang et al., 2020]. In social navigation, datasets can also be used to model [Okal and Arras, 2016, Kretzschmar et al., 2016] or evaluate robot navigation behavior [Biswas et al., 2021]. For this, an in-the-wild pedestrian dataset that is large-scale and supports ground-truth metric labels is desired.

However, existing public pedestrian datasets are either unlabeled [Karnan et al., 2022, Paez-Granados et al., 2022], only contain labels produced by an automated pipeline [Brščić et al., 2013, Majecka, 2009], only contain pixel level information [Robicquet et al., 2016, Oh et al., 2011], or are small in scale [Pellegrini et al., 2009, Lerner et al., 2007, Martin-Martin et al., 2021]. We propose a system that can efficiently collect large quantities of quality data. The data collected using our system feature a novel combination of three critical elements: a combination of top-down and egocentric views, natural human motion, and human-verified labels grounded in the metric space. This allows the data collected using our system to contain rich information.

Large datasets with high-quality labels and rich information can help address human behavioral research questions that require modeling interactions. For example, a key problem researchers have tried to address is the *freezing robot problem* [Trautman and Krause, 2010]. Researchers have attributed this problem to the robot's inability to model interactions [Sun et al., 2021]. Some works [Nishimura et al., 2020] have used datasets to show that modeling human reactions to the robot's actions enables the robot to deliver better performance. However, interactions are diverse and uncommon in human crowds. The set of possible interactions contains many types [Mavrogiannis et al., 2021] and can be further diversified by the environment (e.g. an open plaza or a narrow corridor), so pedestrian datasets need to be large-scale in order to capture enough interaction data.

Autonomous vehicle datasets [Caesar et al., 2020, Cordts et al., 2016] have inspired a plethora

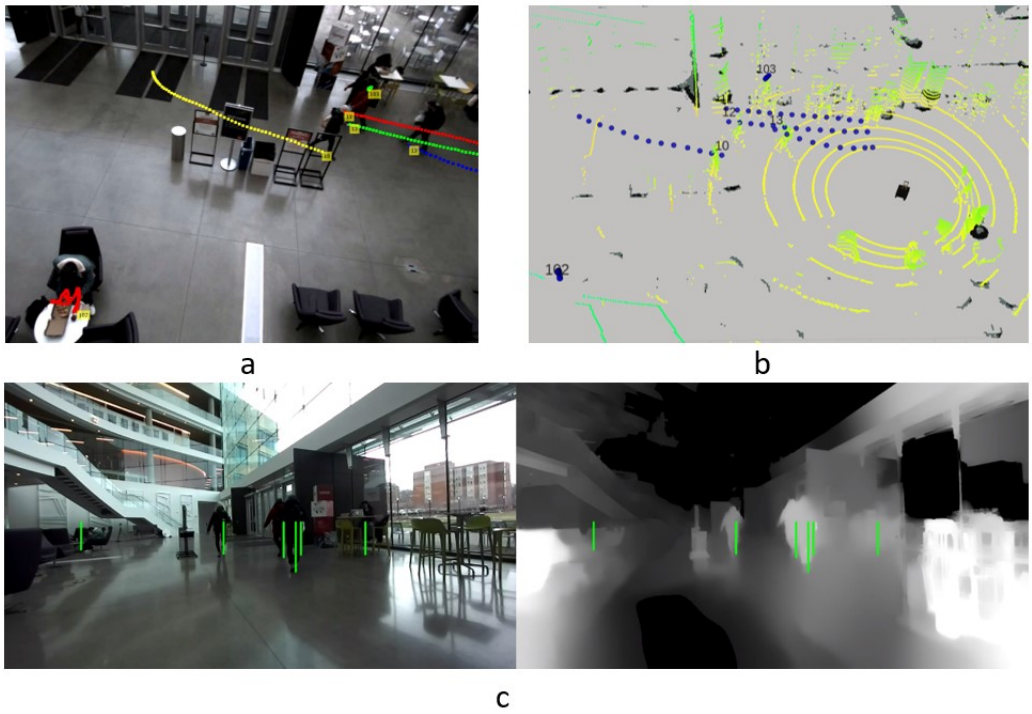

Figure 6.1: This set of images represents the same moment recorded from multiple sensors: a) Top-down view image taken by a static camera with grounded pedestrian trajectory labels shown. b) Ego-centric point cloud from a 3D lidar with the projected trajectories from (a). c) Ego-centric RBG and depth images from a mounted stereo camera. Green vertical bars represent the projected labels. Note that two pedestrians at the back are respectively partially and completely occluded from the stereo camera.

of research. However, a dataset of similar caliber and label quality in pedestrian-dominant environments has yet to arrive. As a step toward this goal, we have constructed a data collection system that can achieve these two requirements: large data quantity and diversity and human-verified positional labels. First, we ensure that our equipment is portable and easy to set up. This allows data to be collected at a variety of locations with limited lead time. Second, we address the challenge of labeling large quantities of data using a semi-autonomous labeling pipeline. We employ a state-of-the-art deep learning-based [Zhang et al., 2021] tracking module combined with a human inspection and tracking error-fixing web app to semi-automatically produce high-quality ground truth pedestrian trajectories in metric space. We make the web app open source[1] so that other researchers can use this tool or contribute to this effort.

[1] https://github.com/CMU-TBD/tbd_label_correction_UI

While we hope our contributions support robot system improvements in the community and we aim to accommodate a wide variety of pedestrian behavior research, our dataset primarily supports human environment navigation research that requires ground truth pedestrian positional information, such as social navigation, pedestrian trajectory prediction, and ego-centric perception. Specifically, we include three important characteristics.

(1) Top-down view and ego-centric views: This ensures that the ego-centric view data has access to ground-truth data even with occlusions.

(2) Natural human motion: Manually pushing the inconspicuous suitcase robot mitigates the curiosity effects of nearby pedestrians.

(3) Ground truth labeling in metric space: This allows our dataset to be useful for research where positional pedestrian data are needed.

To the best of our knowledge, other publicly available datasets have at most two of these characteristics.

We demonstrate our system through a dataset collected in a large indoor space: the TBD Pedestrian Dataset[2]. Our dataset contains scenes with a variety of crowd densities and pedestrian interactions. We show through our analysis that our dataset (Batch 1: 133 minutes - 1416 trajectories; Batch 2: 626 minutes - 10300 trajectories) is larger in scale and contains unique characteristics compared to prior similar datasets. This is an ongoing effort, and we plan to collect additional data in more diverse locations.

6.2 Related Work

With the explosion of data-hungry machine learning methods in robotics, demand for pedestrian datasets has been on the rise in recent years. One popular category of research in this domain is human trajectory prediction (e.g., [Alahi et al., 2016, Gupta et al., 2018, Sadeghian et al., 2019, Mohamed et al., 2020, Ivanovic and Pavone, 2019, Kitani et al., 2012, Liang et al., 2020, Wang and Steinfeld, 2020]). Much of this research utilizes selected mechanisms to model pedestrian interactions in hopes for better prediction performance (e.g., pooling layers in the deep learning frameworks [Alahi et al., 2016, Gupta et al., 2018] or graph-based representations [Mohamed et al., 2020]). [Rudenko et al., 2019] provides a good summary of this topic. While the state-of-the-art performance continues to improve with the constant appearance of newer models, it is often unclear

[2] https://tbd.ri.cmu.edu/tbd-social-navigation-datasets

how well these models can generalize in diverse environments. As shown in [Rudenko et al., 2019], many of these models only conduct their evaluation on the relatively small-scale ETH [Pellegrini et al., 2009] and UCY [Lerner et al., 2007] datasets.

Another popular demand for pedestrian datasets comes from social navigation research. Compared to human motion prediction research, social navigation research focuses more on planning. For example, much of social navigation research uses learning-based methods to identify socially appropriate motions for better robot behavior. These methods include deep reinforcement learning [Everett et al., 2018, Chen et al., 2019, 2020] and inverse reinforcement learning [Okal and Arras, 2016, Tai et al., 2018]. Due to the lack of sufficiently large datasets, these models often train in simulators that lack realistic pedestrian behavior. Apart from training, datasets are also increasing in popularity in social navigation evaluation due to their realistic pedestrian behavior [Gao and Huang, 2021]. Social navigation methods are often evaluated in environments using pedestrian data trajectory playback (e.g., [Trautman et al., 2015, Cao et al., 2019, Sun et al., 2021, Wang et al., 2022]). However, similar to human motion prediction research, these evaluations are typically only conducted on the ETH [Pellegrini et al., 2009] and UCY [Lerner et al., 2007] datasets, as shown by [Gao and Huang, 2021]. These two datasets only use overhead views and therefore lack the ego-centric view used by most robots.

Large-scale and high-quality datasets exist for other navigation-related applications and research. Autonomous vehicle datasets such as nuScenes [Caesar et al., 2020], Cityscapes [Cordts et al., 2016] and ArgoVerse [Wilson et al., 2021] also contain pedestrian-related data. However, pedestrians often have limited appearances on sidewalks or in crosswalks. There is also no data on how pedestrians navigate indoors in these autonomous vehicle datasets. Another group of similar datasets mainly supports computer vision-related research, such as MOT [Dendorfer et al., 2020] for pedestrian tracking and the Stanford Drone Dataset (SDD) [Robicquet et al., 2016] and VIRAT [Oh et al., 2011] for prediction of pedestrian motions/goals at the image level. Detailed comparisons of the characteristics between the TBD Pedestrian Dataset and similar existing datasets can be found in Section 7.2.1.

Simulators can fill the role of datasets for both training and evaluation. Simulators such as PedSIM [Gloor, 2016], CrowdNav [Chen et al., 2019], SocNavBench [Biswas et al., 2021] and SEAN [Tsoi et al., 2020] are in use by the research community. However, sim-to-real transfer is an unsolved problem in robotics. Apart from the lack of fidelity in visuals and physics, pedestrian simulators in particular entail the additional paradox of pedestrian behavior realism [Mavrogiannis et al., 2021]: If pedestrian models are realistic enough for use in simulators, why don't we apply the same model to social navigation?

CHAPTER 7

RICH, PORTABLE, AND LARGE-SCALE NATURAL PEDESTRIAN DATA: SET 1

Work in this chapter is featured in IROS2022 EMPP Workshop [Wang et al., 2022].

7.1 System Description - Set 1

In this project, we introduce a portable and easy-to-configure data collection system that will allow scalable collection of large quantities of data. The data collection setup also contains a cart that provides data on naturalistic pedestrian reactions to the robot from a typical perspective view.

7.1.1 Hardware Setup

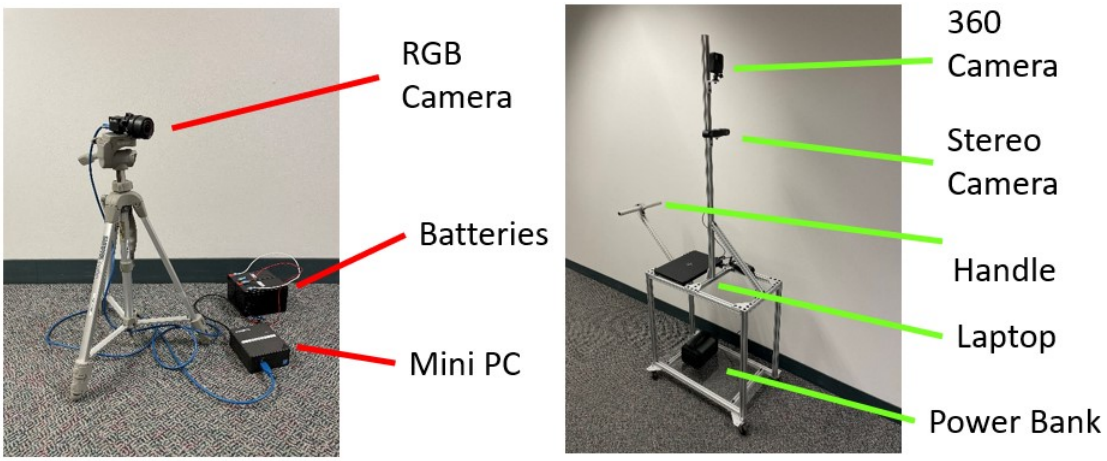

Figure 7.1: Sensor setup used to collect the TBD Pedestrian Dataset. (Left) One of three nodes used to capture top-down RGB views. Each node is self contained with an external battery and communicates wirelessly with other nodes. (Right) Cart used to capture sensor views from the mobile robot perspective during data collection. The cart is powered by an onboard power bank and laptop.

As shown in Figure 7.2, we positioned three FLIR Blackfly RGB cameras (Figure 7.1) surrounding the scene on the upper floors overlooking the ground level at angles of approximately 90 degrees apart from each other. The RGB cameras are connected to portable computers powered by

lead-acid batteries. We also positioned three more units on the ground floor, but did not use them for pedestrian labeling. Compared to a single overhead camera, multiple cameras ensure better pedestrian labeling accuracy. This is achieved by labeling pedestrians from cameras that have the highest image resolution for a given pedestrian (i.e., are closest).

In addition to the RGB cameras, we pushed a cart (Figure 7.1) equipped with a ZED stereo camera through the scene to collect both perspective RGB views and depth information. A GoPro Fusion 360 camera for capturing high-definition 360° videos of nearby pedestrians was mounted above the ZED. Data from on-board cameras are useful for capturing pedestrian pose data and facial expressions. The ZED camera was powered by a laptop with a power bank. Our entire data collection hardware system is portable and does not require power outlets, allowing data collection outdoors or in areas where wall power is inaccessible.

During each data collection session, we pushed the cart from one end of the scene to another end, avoiding pedestrians and obstacles along the way in a natural motion similar to a human pushing a delivery cart. The purpose of this cart was to represent a mobile robot that traverses the human environment. However, unlike other datasets such as [Yan et al., 2017], [Martin-Martin et al., 2021], [Karnan et al., 2022] and [Paez-Granados et al., 2022] that use a teleoperated robot or [Rudenko et al., 2020] that uses a scripted policy to act autonomously, we chose to have all motion performed by the human walking with the system. This provides better trajectory control, increased safety, and further reduces the novelty effect on surrounding pedestrians.

The first batch of our data collection occurred at ground level in a large indoor atrium area (Figure 7.2). Half of the atrium area had fixed entry/exit points that led to corridors, elevators, stairs, and doors to the outside. The other half of the atrium was adjacent to another large open area and was unstructured with no fixed entry/exit points. We collected data around lunch and dinner times to ensure higher crowd densities.

7.1.2 Post-processing and Labeling

A summary of our post-processing pipeline is summarized in Figure 7.3. We expand on select nodes to explain the post-processing procedures in greater detail.

7.1.2.1 Time synchronization

To ensure time synchronization across the captured videos, we employed Precision Time Protocol over a wireless network to synchronize each of the computers powering the cameras, which allows for sub-microsecond synchronization. For redundancy, we held an LED light at a location inside

Figure 7.2: Hardware setup for the TBD Pedestrian Dataset. Red circles indicate positions of RGB cameras. The green box shows our mobile cart with a 360° camera and stereo camera which imitate a mobile robot sensor suite. The cart is manually pushed by a researcher during recording. The white area is where trajectory labels are collected.

the field of view of all cameras and switched it on and off at the beginning of each recording session. We then checked for the LED light signal during the post-processing stage to synchronize the starting frame of all the captured videos for each recording session. We observed very little time drift in the individual recording computer clocks throughout the duration of each recording session, meaning that one synchronization point at the beginning of the recording sufficed.

Due to the portable nature of our system and the long distances between the cameras and the scene, we used scene reconstruction techniques to retrieve the intrinsics and poses of the cameras. We used COLMAP [Schönberger, 2018] to perform a 3D reconstruction of the scene and estimated static camera poses and intrinsics by additionally supplying it with dozens of static pictures of the atrium taken from a smartphone. The effectiveness of obtaining the camera parameters this way may also be applied to future work. For example, it may be possible to use crowdsourced approaches to collect such data when trying to repeat our effort with other camera deployments (e.g., a building atrium with multiple security cameras), since hundreds of images and videos may be available in populous areas.

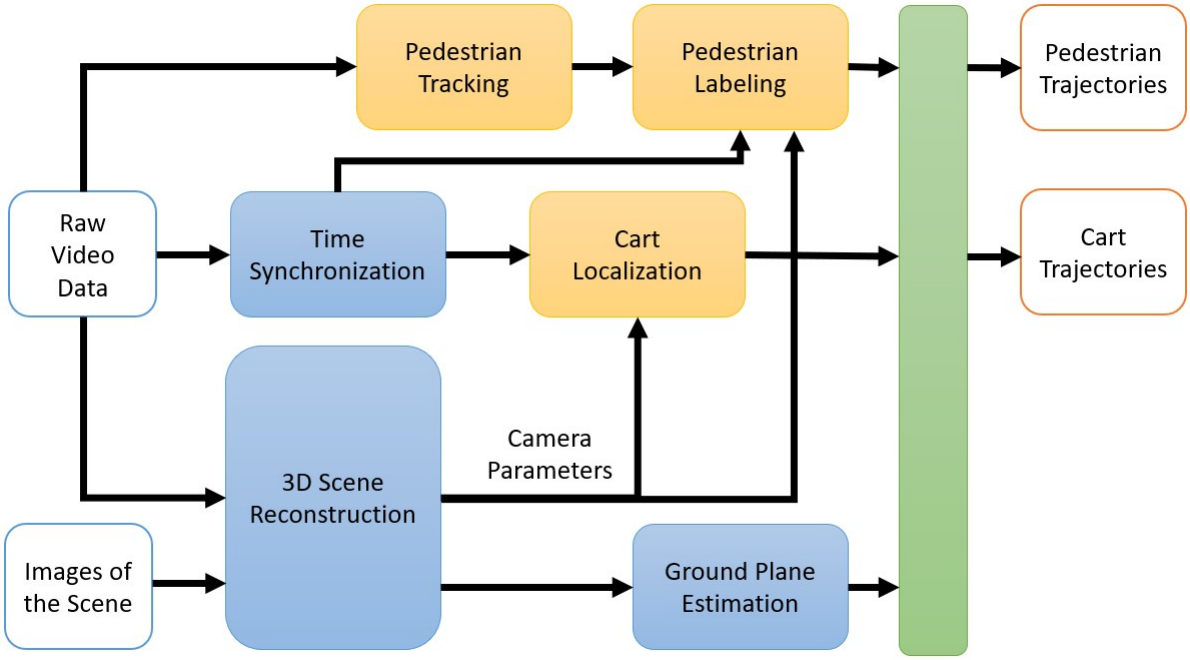

Figure 7.3: Flowchart for our post-processing pipeline. Blue blocks are preparation procedures and orange blocks are labeling procedures. The green block transforms all the trajectory labels onto the ground plane $z = 0$.

7.1.2.2 Ground plane identification

After the 3D reconstruction, the ground plane was not always $z = 0$, but $z = 0$ is usually the assumption for pedestrian datasets. To normalize our data, we first defined an area on the ground plane and selected all the points within the area $\mathcal{P}$. We then used RANSAC [Fischler and Bolles, 1981] for maximum accuracy to identify a 2D surface G within $\mathcal{P}$.

$$G = \text{RANSAC}(\mathcal{P}), \tag{7.1}$$

where G is expressed as $g_a x + g_b y + g_c z + g_d = 0$. Once the ground plane was identified, it was trivial to apply simple geometry to identify the homography matrix that transforms the coordinates on G to $G' : z = 0$.

7.1.2.3 Cart localization

After the cameras were synchronized and calibrated, the next step was to localize the cart in the scene. This was achieved by first identifying the cart on the static camera videos and then applying

the camera matrices to obtain the metric coordinates. We are exploring other localization methods (e.g., visual odometry and ultra wide band positioning) and will continue to track progress on large-space localization. For the first batch of data included in our dataset, we manually labeled the locations of the cart.

7.1.2.4 Pedestrian tracking and labeling

Similar to cart localization, we first tracked the pedestrians in static camera videos and then identified their coordinates on the ground plane G. We found ByteTrack [Zhang et al., 2021] to be very successful in tracking pedestrians in the image space. Upon human verification over our entire first batch of data, ByteTrack successfully aided the trajectory labeling of 91.8% of the pedestrians automatically.

Figure 7.4: Smoothing of noise in auto-generated pedestrian trajectories by applying 3D correction. (Left) Raw tracking results from ByteTrack [Zhang et al., 2021] (pixel space). Some noise is present due to human body motion. (Right) Accounting for noise in 3D results in more accurate labeling.

However, the process to do so was different from cart localization in section 7.1.2.3, where the cart is tracked either manually or automatically (attached AprilTag). For automatic tracking of pedestrians, the body movements of the pedestrian while walking created significant noise, as shown in Figure 7.4. Therefore, the tracking noise was in 3D and assumptions that the noise solely exists on G may result in large labeling inaccuracies.

We addressed this issue by estimating the 3D metric coordinates from two cameras, instead of assuming that the metric coordinates are on the 2D plane G, and obtaining these coordinates from a single camera. For each camera, we had a 3×4 camera matrix P.

$$P = \begin{bmatrix} —\boldsymbol{p}_1— \\ —\boldsymbol{p}_2— \\ —\boldsymbol{p}_3— \end{bmatrix}, \tag{7.2}$$

where we had P_1, P_2, P_3 for the three cameras, respectively. For a given 2D point coordinate $\boldsymbol{x}$ we wanted to estimate its corresponding 3D coordinate $\boldsymbol{X}$, so we had $\boldsymbol{x} = \alpha P\boldsymbol{X}$. We then applied the cross-product technique to eliminate the scalar α. This gave us $\boldsymbol{x} \times P\boldsymbol{X} = \mathbf{0}$, or more precisely

$$\begin{bmatrix} y\boldsymbol{p}_3^\top - \boldsymbol{p}_2^\top \\ \boldsymbol{p}_1^\top - x\boldsymbol{p}_3^\top \end{bmatrix} \boldsymbol{X} = \mathbf{0} \tag{7.3}$$

With two cameras $P_i, P_j | i \neq j,\ (i,j) \in \{1,2,3\}$, their corresponding 2D image points $(x_i, y_i), (x_j, y_j)$, and the constraint that the 3D coordinates should be on the ground plane G, we constructed the following system of equations to estimate the 3D coordinates.

$$A\boldsymbol{X} = \begin{bmatrix} y_i\boldsymbol{p}_{i,3}^\top - \boldsymbol{p}_{i,2}^\top \\ \boldsymbol{p}_{i,1}^\top - x_i\boldsymbol{p}_{i,3}^\top \\ y_j\boldsymbol{p}_{j,3}^\top - \boldsymbol{p}_{j,2}^\top \\ \boldsymbol{p}_{j,1}^\top - x_j\boldsymbol{p}_{j,3}^\top \\ g_a, g_b, g_c, g_d \end{bmatrix} \boldsymbol{X} = \mathbf{0} \tag{7.4}$$

We then performed singular value decomposition (SVD) on A to obtain the solution.

Once we obtained the automatically tracked labels in pixel space, we needed to convert them into metric space. With ByteTrack, each camera video contained a set of tracked trajectories in the image space $T_i = \{t_1, ..., t_n\}, i \in \{1,2,3\}$ where i is the camera index. We estimated the 3D trajectory coordinates for each pair of 2D trajectories $(t_i, t_j) | t_i \in T_i, t_j \in T_j, i \neq j$ and the set of estimated coordinates that resulted in the lowest reprojection error were selected to be the final trajectory coordinates in the metric space. We then projected these 3D coordinates onto the ground plane G to obtain the final metric coordinates.

Finally, we performed human verification over the entire tracking output, fixing any errors observed during the process. We also manually identified pedestrians who were outside our target tracking zone but had interactions with pedestrians inside the tracking zone and included them as part of our dataset.

7.2 Dataset Characteristics

7.2.1 Comparison with Existing Datasets

Compared to existing datasets collected in natural pedestrian-dominant environments, our TBD pedestrian dataset contains three components that greatly enhance the dataset's utility. These components are:

Table 7.1: A survey of existing pedestrian datasets and how they incorporate the three components in section 7.2.1. For component 1, a "No" means either not human verified or not grounded in metric space. For component 2, TD stands for "top-down view" and "E" stands for "ego-centric view".

Datasets	Comp. 1 (metric labels)	Comp. 2 (views)	Comp. 3 ("robot")
TBD (Ours)	Yes	TD + E	Human + Robot
ETH [Pellegrini et al., 2009]	Yes	TD	N/A
UCY [Lerner et al., 2007]	Yes	TD	N/A
Edinburgh Forum [Majecka, 2009]	No	TD	N/A
VIRAT [Oh et al., 2011]	No	TD	N/A
Town Centre [Benfold and Reid, 2011]	No	TD	N/A
Grand Central [Zhou et al., 2012]	No	TD	N/A
CFF [Alahi et al., 2014]	No	TD	N/A
Stanford Drone [Robicquet et al., 2016]	No	TD	N/A
L-CAS [Yan et al., 2017]	No*	E	Robot
WildTrack [Chavdarova et al., 2018]	Yes	TD	N/A
JackRabbot [Martin-Martin et al., 2021]	Yes	E	Robot
ATC [Brščić et al., 2013]	No	TD	N/A
THÖR [Rudenko et al., 2020]	Yes	TD + E	Robot
SCAND [Karnan et al., 2022]	No	E	Robot
Crowd-Bot [Paez-Granados et al., 2022]	No	E	Human + Robot

Human verified labels grounded in metric space. ETH [Pellegrini et al., 2009] and UCY [Lerner et al., 2007] datasets are the most popular datasets among human behavior analysis papers [Rudenko et al., 2019]. We believe this is partly because the trajectory labels in these datasets are human verified and are grounded in metric space rather than pixel space (e.g. [Robicquet et al., 2016] and [Benfold and Reid, 2011] only contain labels in bounding boxes). Having labels grounded in metric space eliminates the possibility that camera poses might have an effect on the scale of the labels. It also makes the dataset useful for robot navigation related research because robots plan in the metric space rather than in pixel space.

Combination of top-down views and ego-centric views. Similar to datasets with top-down views, we used top-down views to obtain ground truth trajectory labels for every pedestrian present in the scene. Similar to datasets with ego-centric views, we gathered ego-centric views from a "robot" to imitate robot perception of human crowds. A dataset that contains both top-down views and ego-centric views will be useful for research projects that rely on ego-centric views. This

allows ego-centric inputs to their models while still having access to ground truth knowledge of the entire scene.

Naturalistic human behavior with the presence of a "robot". Unlike datasets such as [Yan et al., 2017], [Martin-Martin et al., 2021], [Karnan et al., 2022], and [Paez-Granados et al., 2022], the "robot" that provides ego-centric view data collection is a cart or a suitcase robot being pushed by a human. As mentioned in Section 7.1.1, doing so reduces the novelty effects on the surrounding pedestrians. Having the "robot" pushed by humans also ensures safety for the pedestrians, and its own motion has less jerk and more human-like behavior.

As shown in Table 7.1, current datasets contain only at most two of the three components[1]. A close comparison is the THÖR dataset [Rudenko et al., 2020], but its ego-centric view data are collected by a robot running on predefined trajectories. Additionally, unlike all other datasets in Table 7.1, the THÖR dataset is collected in a controlled lab setting rather than in the wild. This injects artificial factors into human behavior.

7.2.2 Dataset Statistics

Table 7.2: Comparison of statistics between our dataset and other datasets that provide human verified labels grounded in the metric space. For total time length, 51 minutes of our dataset includes the perspective view data.

Datasets	Time length	# of pedestrians	Label freq (Hz)
TBD Set 1 (Ours)	133 mins (51 mins)	1416	60
ETH [Pellegrini et al., 2009]	25 mins	650	15
UCY [Lerner et al., 2007]	16.5 mins	786	2.5
WildTrack [Chavdarova et al., 2018]	200 sec	313	2
JackRabbot [Martin-Martin et al., 2021]	62 mins	260	7.5
THÖR [Rudenko et al., 2020]	60+ mins	600+	100

Table 7.2 demonstrates the benefit of a semi-automatic labeling pipeline. With the aid of an autonomous tracker, the dataset we have collected so far has already surpassed all other datasets that provide human-verified labels in the metric space in terms of total time and number of pedestrians.

[1]*L-CAS dataset does provide human verified labels grounded in the metric space. However, its pedestrian labels do not contain trajectory data.

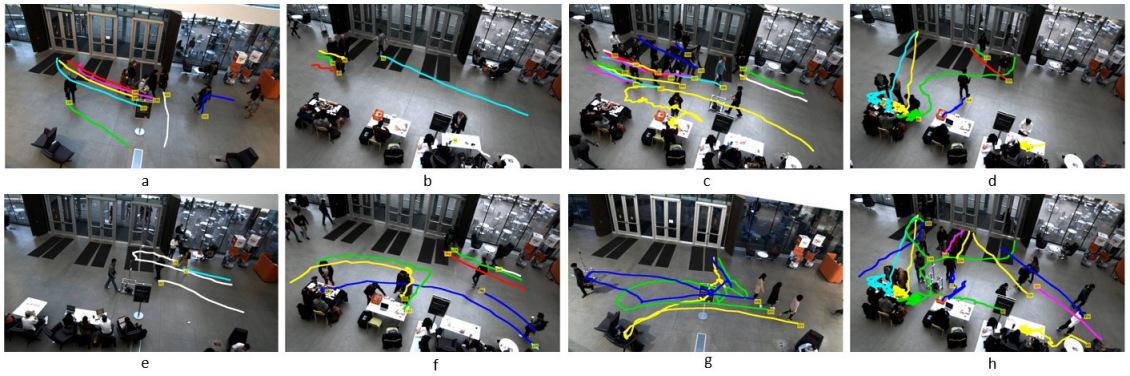

Figure 7.5: Example scenes from the TBD Pedestrian Dataset. a) A dynamic group. b) A static conversational group. c) A large tour group with 14 pedestrians. d) A pedestrian affecting other pedestrians' navigation plans by asking them to come to the table. e) Pedestrians stop and look at their phones. f) Two pedestrians change their navigation goals and turn towards the table. g) A group of pedestrians change their navigation goals multiple times. h) A crowded scene where pedestrians are heading towards different directions.

7.2.3 Qualitative Pedestrian Behavior

Due to the nature of the environment in which we collected the data, we observed a mixture of corridor and open-space pedestrian behavior, many of which are rarely seen in other datasets. As shown in Figure 7.5, we observed both static conversation groups and dynamic walking groups. We also observe that some pedestrians naturally change goals mid-navigation.

7.3 Conclusion

This chapter presents a data collection system that is portable and enables large-scale data collection. Our system offers better utility for pedestrian behavior research because our system contains human verified labels grounded in the metric space, a combination of both top-down views and perspective views, and a human-pushed cart that approximates naturalistic human motion with a socially-aware "robot". We further couple the system setup with a semi-autonomous labeling process that easily produces human-verified labels in order to meet the demands of the large-scale data collected by our hardware. Lastly, we present the TBD Pedestrian Dataset we have collected using our system, which not only exceeds the quantity of similar datasets, but also offers unique pedestrian interaction behavior that adds to the qualitative diversity of pedestrian interaction data.

CHAPTER 8

RICH, PORTABLE, AND LARGE-SCALE NATURAL PEDESTRIAN DATA: SET 2

Work in this chapter is under review for ICRA2024 [Wang et al., 2023]

8.1 System Description - Set 2

We have made many improvements on both the hardware setup and the post-processing pipeline for Set 2.

8.1.1 Hardware Setup

Similar to the system setup in Section 7.1.1, our system supports multiple static FLIR Blackfly RGB cameras for labeling and calculation of the metric space (Figure 8.1). The three cameras surround the scene on the upper floors overlooking the ground level scene spaced at angles approximately 90 degrees apart (Figure 8.2). The RGB cameras are connected to portable computers powered by lead-acid batteries.

Instead of a cart, we pushed a robotic suitcase [Kuribayashi et al., 2023] through the scene. The suitcase robot (Figure 8.1) is a converted carry-on suitcase. It is equipped with an IMU and a 3D lidar sensor. In addition, the same ZED camera and GoPro Fusion 360 camera are mounted on the

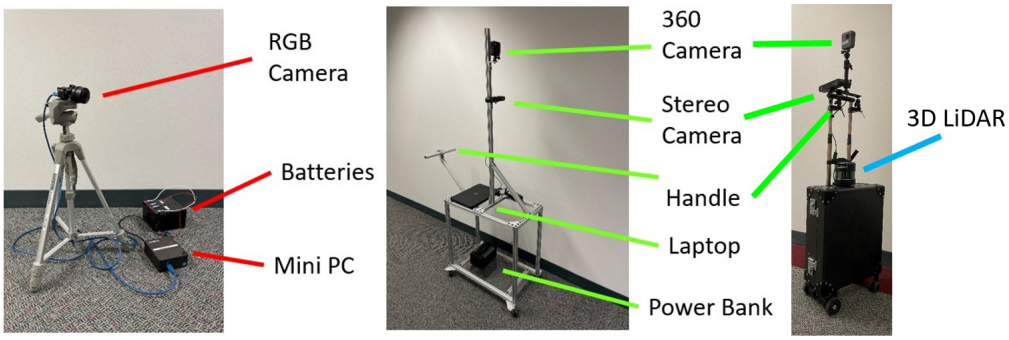

Figure 8.1: Updated sensor setup used to collect the TBD Pedestrian Dataset. (left) One of the nodes used to capture top-down RGB views. (middle) The cart used to capture ego-centric sensor views during data collection for Set 1. (right) The suitcase robot used to capture ego-centric sensor views during data collection for Set 2.

Figure 8.2: Hardware setup for the TBD Pedestrian Dataset. Blue circles indicate positions of RGB cameras. The green box shows our suitcase robot pushed through the scene. The white area is where trajectory labels are collected. The data collection area is much larger for Set 2.

suitcase handle. The robot's computer, batteries, and all its internal components are hidden inside the suitcase, so pushing the robot resembles pushing a suitcase. We selected this robot because of its inconspicuous design to reduce curious, unnatural reactions from nearby pedestrians, as curious pedestrians may intentionally block robots or display other unnatural movements [Brščić et al., 2015]. While it is true that real-world pedestrians will react to mobile robots curiously in the short term and some may argue in favor of a more robotic appearance, we envision that such curiosity effects will die down in the long term.

Similar to Section 7.1.1, during certain data collection sessions we pushed the suitcase robot from one end of the scene to another in a natural motion similar to a human walking with a suitcase. This also collects ego-centric views from a mobile robot traversing the human environment.

This set of data was collected in the same area as set 1 (Figure 8.2), but we have expanded the area where we label pedestrians. This allows us to collect pedestrian trajectory data with longer durations. Additional data collection efforts are planned at more diverse locations.

8.1.2 Post-processing Pipelines

The initial post-processing steps are the same as in Section 7.1.2. However, we have made many optimizations to these post-processing steps, and we can obtain the ground plane and the camera parameters directly from the calibrated cameras. The extrinsics of the cameras are calibrated using

measured keypoints from the environment. 3D scene reconstruction and ground plane estimation are no longer needed.

We used the suitcase robot's onboard localization stack to localize the suitcase in the scene. For Set 2, we first made a map inside the building and then computed the robot's location in the post-processing phase using the robotic suitcase software [1] powered by Cartographer [2].

For pedestrian tracking, we again tracked the pedestrians on the overhead camera videos. We found ByteTrack [Zhang et al., 2021] to be very successful in tracking pedestrians in the image space. We have additionally prepared the inputs to be of higher quality than those of Set 1. As a result, after human verification of all our data, this time ByteTrack successfully tracked 95.1% of the pedestrians automatically.

For the automatically tracked labels in the pixel space, we needed to convert them into metric space. Each camera video contained a set of tracked trajectories in the image space. We estimated the 3D trajectory coordinates for each pair of 2D trajectories from different cameras, and the set of estimated coordinates that resulted in the lowest reprojection error were selected to be the trajectory coordinates in the metric space. This process is also much simpler than the process described in Section 7.1.2.4.

8.1.3 Human Label Verification

To ensure the quality of the data labels, human verification of the tracked trajectories from ByteTrack is desired. Semi-autonomous labeling procedures are common in autonomous driving datasets and pedestrian datasets. However, in a survey of existing pedestrian dataset literature, we noticed that datasets that contain human-verified metric space labels are often relatively small [Pellegrini et al., 2009, Lerner et al., 2007, Chavdarova et al., 2018, Martin-Martin et al., 2021], and large-scale datasets often only use automated tracking pipelines [Majecka, 2009, Oh et al., 2011, Brščić et al., 2013] or do not label surrounding pedestrians [Karnan et al., 2022, Paez-Granados et al., 2022]. We attribute this to a lack of tools to streamline the human verification process.

To this end, we designed an open-source web app (Figure 8.3) using Matlab App Designer. The tool was designed to minimize the complete human relabeling of erroneously tracked trajectories. The app contains a media player. When using the app, human labelers watch videos with the automatically tracked trajectories. When an error is noticed, the labeler only needs to indicate to the system the type and location of the error. The system then fixes the errors and updates the

[1] https://github.com/cmu-cabot
[2] https://github.com/cartographer-project/cartographer

Figure 8.3: App interface for the human verification process. It contains a media player and various options to fix tracking errors automatically and manually.

trajectory visualization accordingly. Currently, the app contains the following set of error-fixing options:

- **Break**: Used when ByteTrack incorrectly assigns the same trajectory to two different pedestrians.
- **Join**: Used when two different trajectories actually belong to the same pedestrian.
- **Delete**: Used when a ghost trajectory appears, such as incorrectly tracking an unworn jacket as a pedestrian.
- **Disentangle**: Used when the two trajectories of two pedestrians are swapped in the middle, which can happen when one partially occludes the other.

Table 8.1: Comparison statistics for datasets with human verified labels grounded in metric space. Numbers in parenthesis are for data that includes the ego-centric view.

Datasets	Time length	# Trajectories	Label Freq (Hz)
TBD Set 1	133 (51) mins	1416	60
TBD Set 2	626 (213) mins	10300	10
ETH [Pellegrini et al., 2009]	25 mins	650	15
UCY [Lerner et al., 2007]	16.5 mins	786	2.5
WildTrack [Chavdarova et al., 2018]	200 sec	313	2
JackRabbot [Martin-Martin et al., 2021]	62 mins	260	7.5
THÖR [Rudenko et al., 2020]	60+ mins	600+	100

The web app also supports undoing previous actions, partial or complete relabeling of trajectories, and labeling missing trajectories. For future work, we are looking at possible platforms to launch the app so that the human verification process can be a crowd-sourced effort.

Combined with ByteTrack, it took an expert labeler about 30 hours to produce human-verified labels for 375K frames of data, or 10300 trajectories. ByteTrack successfully tracks 95.1% of trajectories. For trajectories that contain errors requiring human rectification, 0.35% are fixed by "Break", 1.25% are fixed by "Join", 0.29% are fixed by "Delete", 0.89% are fixed by "Disentangle", 1.21% are fixed by "Relabel", and 0.48% are fixed by "Missing".

8.2 Dataset Characteristics and Analysis

8.2.1 Dataset Size

Table 8.1 again demonstrates the ability of a semi-automatic labeling pipeline to produce large amounts of data. With the aid of our error-fixing tool, humans only need to verify and make occasional corrections on the tracking results rather than locating pedestrians on every single frame. Set 2 of the data we have collected so far surpasses all other datasets that provide human-verified labels in the metric space as well as Set 1 by a great margin in terms of total time and number of pedestrians. It is more than 10 times bigger in total time and number of trajectories. We will continue this effort and collect more data for future work.

Table 8.2: Comparison of statistics between our dataset and other datasets according to the methods in [Rudenko et al., 2020].

Datasets	Tracking Duration $[s]$	Percep. Noise $[ms^{-2}]$	Motion Speed $[ms^{-1}]$	Min Dist. To Ped. $[m]$
TBD Set 2	25.6 ± 57.1	0.55	0.88 ± 0.52	1.25 ± 1.44
THÖR [Rudenko et al., 2020]	16.7 ± 14.9	0.12	0.81 ± 0.49	1.54 ± 1.60
ETH [Pellegrini et al., 2009]	9.4 ± 5.4	0.19	1.38 ± 0.46	1.33 ± 1.39
ATC [Brščić et al., 2013]	39.7 ± 64.7	0.48	1.04 ± 0.46	0.61 ± 0.16
Edinburgh [Majecka, 2009]	10.1 ± 12.7	0.81	1.0 ± 0.64	3.97 ± 3.5

8.2.2 Dataset Statistics

Extending the evaluations performed in THÖR [Rudenko et al., 2020], we added the same suite of analyses to Set 2 of our TBD dataset. The evaluation metrics were the following. (1) *Tracking Duration* (s): Average time duration of the tracked trajectories. (2) *Perception Noise* (ms^{-2}): The average absolute acceleration of the trajectories. (3) *Motion Speed* (ms^{-1}): Velocities of the trajectories measured in 1 second intervals. (4) *Minimum Distance Between People* (m): Minimum Euclidean distance between two closest observed people.

As shown in Table 8.2, our dataset has a considerable average trajectory duration (± 25.6) and a large variation (± 57.1), second only to ATC, which has a coverage area of $900m^2$. Although our dataset has a much smaller coverage, we attribute this to the presence of pedestrians changing navigation goals and static pedestrians in our dataset. Static pedestrians include standing pedestrians having conversations or pedestrians sitting on chairs. Their presence in our dataset often has a long duration, which also causes a big variation in this metric. The tracking noise of our system was sub-optimal when compared to other datasets, which is likely due to noisy tracking of the sitting pedestrians. We observed that sitting pedestrians change their body poses frequently, which causes the tracked bounding boxes to change size frequently. We will investigate how to improve this for future work. The motion speeds of our dataset trajectories are lower ($0.88ms^{-1}$), suggesting the presence of more static pedestrians. We also have the second-highest variation in motion speed ($\pm 0.52ms^{-1}$), suggesting that our dataset captures a wide range of pedestrian behavior. From the minimum distance between people, it can be inferred that our dataset captures both dense and sparse population scenarios as indicated by the middle mean value ($1.25m$) among the

Table 8.3: Trajectory prediction displacement error on ETH/UCY datasets and TBD dataset Set 2.

Models	ETH/UCY Dataset			
	Static + Dynamic		Dynamic	
	ADE(m)	FDE(m)	ADE(m)	FDE(m)
Social-GAN [Gupta et al., 2018]	0.48	0.96	0.59	1.13
Trajectron++ [Salzmann et al., 2020]	0.27	0.49	0.35	0.65
AgentFormer [Yuan et al., 2021]	0.23	0.39	0.25	0.44
	TBD Set 2			
Social-GAN	0.36	0.72	0.64	1.30
Trajectron++	0.16	0.28	0.43	0.83
AgentFormer	0.15	0.23	0.30	0.52

others and the high variance ($\pm 1.44m$). Note that [Rudenko et al., 2020] also measures trajectory curvatures, but we noticed that this measurement is heavily affected by how static pedestrians are processed. [Rudenko et al., 2020] does not provide details on this, so we decided not to evaluate this metric.

8.2.3 Behavior Distribution Analysis

Additionally, we leveraged trajectory prediction models to evaluate our dataset. We believe that these well-trained models can be utilized in other ways, such as characterizing the variety of pedestrian behavior in datasets. Almost all trajectory prediction models have been tuned and trained on ETH/UCY datasets. Some have additionally made predictions on SDD [Robicquet et al., 2016] or autonomous driving datasets. Because we were primarily concerned with metric labels and pedestrian environments, we did not evaluate models trained in SDD or autonomous vehicle datasets. We also chose models that largely leverage pedestrian positional data and can work independently without image patch inputs [Sadeghian et al., 2019] or semantic segmentation [Mangalam et al., 2021].

Our dataset contains simple sessions and challenging sessions. To test whether our dataset contains pedestrian behavior outside the ETH/UCY dataset domains, we analyzed those sessions of our dataset that contained great variety in pedestrian behavior for this evaluation. We selected Social-GAN [Gupta et al., 2018] as the baseline model and Trajectron++ [Salzmann et al., 2020] and AgentFormer [Yuan et al., 2021] as relatively state-of-the-art models. Because the models trained on each of the other four subdatasets did not perform significantly differently in our dataset,

we only report the average *Average Displacement Error* (ADE) and the average *Final Displacement Error* (FDE) across the five models.

We observed that when all pedestrians are included, the prediction models all perform better on our dataset compared to other datasets (Table 8.3). We believe this can be attributed to greater numbers of static pedestrians in our datasets compared to ETH/UCY because the models are unlikely to yield large errors when predicting future trajectories of static pedestrians. We define dynamic pedestrians as pedestrians who move at least $1m$ during the prediction window. We included static pedestrians during model inference, but evaluating only on dynamic pedestrians we discovered that all the prediction models' performance degrades. This indicates that the models have encountered more unseen scenarios in our dataset and that the moving pedestrians in our dataset exhibit more diverse navigation behavior and wider behavior distribution compared to the ones in ETH/UCY.

8.3 Conclusion

This project presents an upgraded version of our data collection system that enables large-scale data collection. This project also presents a label verification tool that streamlines the labeling process. Our semi-autonomous pipeline easily produces human-verified labels in order to meet the demands of the large-scale data collected by our hardware. The second set of the TBD Pedestrian Dataset we have collected using our system exceeds the quantity of Set 1 and similar datasets. In addition, it offers more unique pedestrian interaction behavior that expands the qualitative diversity of pedestrian interaction data.

As mentioned above, our approach enables additional data collection in a wide range of locations and constraints. Additional data collection and public updates to this initial dataset are planned. We have also discovered additional challenges with our labeling pipeline on static pedestrians. Because static pedestrians have long trajectory duration and constantly adjust their body poses, the resulting trajectories can be noisy and escape the labeler's attention when using our tool. For future work, we would also explore expanding the diversity of labels. Some examples include: adding activity labels indicating whether the pedestrian is walking, talking or sitting; adding static obstacle labels for human-object interaction studies; adding group labels for pedestrian groups; and adding gaze direction and head orientation labels for the onboard high-definition 360 camera.

Part V

Conclusion

CHAPTER 9

FUTURE WORK AND LIMITATIONS

9.1 Future Work on Group-Based Representations

In Chapters 4 and 5 we explored the benefits of group-based representations in social navigation. In these projects, the robot acts as an outsider agent and navigates around the pedestrian groups. We argue that the potential of using groups in social navigation cannot be fully achieved unless the robot also plays an active role in the grouping process. In other words, when a nearby group reflects the robot's planning strategy, the robot simply needs to follow the group instead of planning its own course of actions. We call this navigation strategy *group surfing* [Du et al., 2019].

As mentioned in the Introduction (Section 1), a common problem that a mobile robot encounters when navigating in dense human crowds is the *freezing robot problem* [Trautman and Krause, 2010]. This problem occurs when all possible navigation paths are blocked during planning, possibly due to conservative estimates of pedestrian dynamics. We encountered similar problems in Chapter 4 where group-based representations occupy larger obstacle space than individual-based

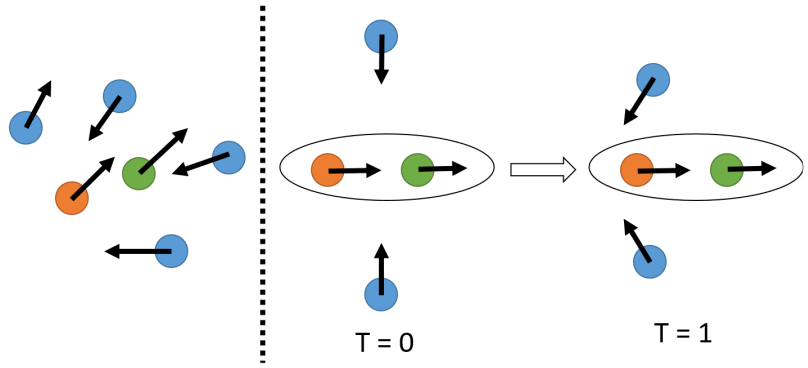

Figure 9.1: The blue circles are pedestrians, the red circle is the robot and the green circle is the pedestrian that the robot has formed a group with. Left: In a crowded situation, we hypothesize that by following the group, the robot can navigate out of the area without the need to predict the surrounding pedestrians' future states and model its prediction uncertainties. Right: We expect pedestrians to respect the group the robot has formed with other pedestrians. In this case, we expect the robot to be able to influence the crossing pedestrians to not cut in front of the robot.

representations. This results in lower efficiency for the robot. If the robot runs into the freezing robot problem but is able to detect a nearby group with similar motions to itself, the robot can latch onto the group and let the humans in the group lead the robot out of the deadlocked situation. An illustrative example is shown in Figure 9.1. In addition to resolving the planning issue, we hypothesize that by forming a group with nearby pedestrians, the robot joins a social space within the group. This may send a signal to all pedestrians that this space is not to be intruded on. As a result, the robot discourages them from cutting into the group that it has joined. This in turn may further help the robot navigate out of the freezing robot problem, as shown in Figure 9.1.

Group surfing has behavioral implications for pedestrians. As mentioned in Chapters 3 and 4, humans employ a psychological process known as Gestalt to group entities with similar motions together. Similar processes also apply when humans navigate in crowds. Not only do humans perceive surrounding pedestrians as groups, they also act in conformity with the actions of surrounding pedestrians, sometimes only using peripheral vision to detect them. This is the reason why humans can navigate while checking their smartphones and why humans form lanes in highly dense crowds.

9.2 Future Work on Datasets

Future work on datasets can be broken down into a) improvements in data collection and b) projects that become viable with the availability of large-scale datasets.

Future work in data collection will include data collection in different environments. The environment plays a key role in shaping the context of pedestrian behavior. Collecting data in more diverse locations will allow our dataset to cover a larger distribution of pedestrian behaviors. A key type of environment in which we need to collect data is a narrow corridor. So far, our data have been collected in a semi-open space environment. In narrow-corridor environments, because the pedestrian's navigation space becomes much more constrained, different navigation behaviors might be observed. Additionally, we can replace the static overhead cameras with a hovering drone that follows the mobile robot. This also allows the overhead camera to be dynamic. A limitation of our current data collection setup is that as the robot approaches the edge of the data labeling area, it will no longer have access to top-down-view based labels of pedestrians who are outside of the data labeling area. A dynamic overhead camera can solve this problem. Lastly, to make our dataset more useful for other potential research areas, we will explore additional labeling pipelines to expand our label diversity. As mentioned in Section 8.3, examples include: adding activity labels indicating whether the pedestrian is walking, talking, or sitting; adding static obstacle labels

for human-object interaction studies; adding group labels for pedestrian groups; and adding gaze direction and head orientation labels for the onboard high-definition 360 camera.

A large-scale pedestrian dataset such as the TBD Pedestrian Dataset enables research in areas that were previously difficult. For example, the group split and merge prediction can be revisited. Having the model trained on a larger dataset with wider distribution coverage of group behavior can allow the model to generalize better to unseen scenarios. During the course of my PhD, we also paused two projects because we determined that a large dataset was needed to train an effective model.

One of the projects is the personal space modeling project. Throughout this thesis, we have been using the egg-shaped personal space project defined by [Kirby, 2010]. This personal space formulation, although better than circles, still does not capture the complexity of proxemics that pedestrians maintain among themselves. We frequently observed pedestrians intruding into other pedestrians' defined personal spaces in the datasets. Personal space should be the minimal space that a pedestrian feels comfortable around other pedestrians. Based on this idea, we proposed that when two pedestrians are closest to each other, they should be on the boundaries of each other's personal space. We developed a learning-based model that learns the parameters of the egg shape [Kirby, 2010], but the end result was not satisfactory. We believe that the lack of data on pairwise pedestrian interaction was the culprit behind this.

The other paused effort is the pedestrian interaction detection project. To better model human-human interaction or human-robot interaction, it would be beneficial to learn whether an interaction takes place in the first place. For example, a pedestrian on their phone will likely walk in a straight line and ignore all surrounding agents. We propose to leverage *counterfactuals* to build an interaction detection model. In other words, we run a trajectory prediction model with and without a select surrounding pedestrian and observe whether there is a difference in the predicted trajectories. If there is, an interaction is likely to take place. Similarly, the lack of a large-scale dataset to train a generalizable trajectory prediction model hindered our progress. We discovered that pedestrians walk in straight paths by default and that interactions resulting in path perturbations are relatively rare occurrences. This observation is supported by [Schöller et al., 2020]. Therefore, a large-scale dataset that contains more instances of interactions that result in path perturbations may help us train a better interaction detection model.

CHAPTER 10

CONCLUSION

10.1 Contributions

In this thesis, we begin by introducing a pipeline to generate group-based representations and explore building prediction models that take these group-based representations as inputs. We then build a group split and merge prediction model and show that this model performs better than converted trajectory prediction models. We additionally show that transferring the group split and merge prediction model directly into simulated laser scan settings results in similar levels of performance.

In the second part of the thesis, we integrate the group-based representations into an MPC-based framework and build a group state prediction model. We show that by leveraging group-based representations and future state predictions, the mobile robot produces safer and more social behavior in simulation. We similarly show that our G-MPC model does not suffer a significant performance downgrade when directly transferred to a simulated laser scan setting. However, a limitation to the framework in this form is that large amounts of computation are required. To address this, we design a visible edge-based simplified group space representation and show that it offers computational benefits while maintaining similar levels of performance when integrated into an MPC framework.

In the third part of the thesis, we describe a portable data collection system coupled with a semi-autonomous labeling pipeline. As part of the pipeline, we designed a label correction web app that facilitates human verification of automated pedestrian tracking results. Our system enables large-scale data collection in diverse environments and fast trajectory label production. Compared to existing pedestrian data collection methods, our system contains three components: a combination of top-down and ego-centric views, natural human behavior in the presence of a socially appropriate "robot", and human-verified labels grounded in the metric space. We further introduce our ever-expanding dataset from the ongoing data collection effort, the *TBD Pedestrian Dataset*, and show that our collected data is larger in scale and contains richer information when compared to prior datasets with human-verified labels.

10.2 Final Words

As described in the Introduction (Chapter 1), leveraging pedestrian groups can help address many issues in social navigation. However, to the best of my knowledge, work in this area has been scarce. There is still a long way to go in exploring groups in social navigation. In addition, the concept of pedestrian groups is a form of abstraction inspired by how humans process surrounding pedestrians. Many other forms of abstraction also possibly exist in pedestrian-rich environments. With enough data, maybe we can also use a large, learning-based model to automatically discover these implicit forms of abstractions. The ideas presented in this thesis are only the first steps towards this direction.

BIBLIOGRAPHY

[1] Alexandre Alahi, Vignesh Ramanathan, and Li Fei-Fei. Socially-aware large-scale crowd forecasting. In *Proc. IEEE Conf. on Comput. Vis. and Pattern Recognit.*, pages 2203–2210, 2014.

[2] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pages 961–971, June 2016.

[3] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social LSTM: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 961–971, 2016.

[4] L. Bazzani, M. Cristani, and V. Murino. Decentralized particle filter for joint individual-group tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1886–1893, 2012.

[5] Younes Belkada, Lorenzo Bertoni, Romain Caristan, Taylor Mordan, and Alexandre Alahi. Do pedestrians pay attention? eye contact detection in the wild. *arXiv preprint arXiv:2112.04212*, 2021.

[6] Ben Benfold and Ian Reid. Stable multi-target tracking in real-time surveillance video. In *CVPR 2011*, pages 3457–3464. IEEE, 2011.

[7] Abhijat Biswas, Allan Wang, Gustavo Silvera, Aaron Steinfeld, and Henny Admoni. Socnavbench: A grounded simulation testing framework for evaluating social navigation. *arXiv preprint arXiv:2103.00047*, 2021.

[8] Dražen Brščić, Takayuki Kanda, Tetsushi Ikeda, and Takahiro Miyashita. Person tracking in large public spaces using 3-d range sensors. *IEEE Trans. on Human-Machine Syst.*, 43(6): 522–534, 2013.

[9] Drazen Brščić, Hiroyuki Kidokoro, Yoshitaka Suehiro, and Takayuki Kanda. Escaping from children's abuse of social robots. In *Proc. of the tenth annual acm/ieee international Conf. on human-robot interaction*, pages 59–66, 2015.

[10] Mateusz Buda, Atsuto Maki, and Maciej A. Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249 – 259, 2018. ISSN 0893-6080.

[11] Wolfram Burgard, Armin B. Cremers, Dieter Fox, Dirk Hähnel, Gerhard Lakemeyer, Dirk Schulz, Walter Steiner, and Sebastian Thrun. The interactive museum tour-guide robot. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 11–18, 1998.

[12] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020.

[13] C. Cao, P. Trautman, and S. Iba. Dynamic channel: A planning framework for crowd navigation. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 5551–5557, 2019.

[14] Chao Cao, Peter Trautman, and Soshi Iba. Dynamic channel: A planning framework for crowd navigation. In *2019 International Conf. on Robotics and Automation (ICRA)*, pages 5551–5557. IEEE, 2019.

[15] Isarun Chamveha, Yusuke Sugano, Yoichi Sato, and Akihiro Sugimoto. Social group discovery from surveillance videos: A data-driven approach with attention-based cues. In *Proceedings of the The British Machine Vision Association (BMVC)*, 2013.

[16] M. Chang, N. Krahnstoever, and W. Ge. Probabilistic group-level motion analysis and scenario recognition. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 747–754, 2011.

[17] I. Chatterjee and A. Steinfeld. Performance of a low-cost, human-inspired perception approach for dense moving crowd navigation. In *Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 578–585, Aug 2016.

[18] Tatjana Chavdarova, Pierre Baqué, Stéphane Bouquet, Andrii Maksai, Cijo Jose, Timur Bagautdinov, Louis Lettry, Pascal Fua, Luc Van Gool, and François Fleuret. Wildtrack: A multi-camera hd dataset for dense unscripted pedestrian detection. In *Proc. IEEE Conf. on Comput. Vis. and Pattern Recognit.*, pages 5030–5039, 2018.

[19] Changan Chen, Yuejiang Liu, Sven Kreiss, and Alexandre Alahi. Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 6015–6022, 2019.

[20] Y. F. Chen, M. Liu, M. Everett, and J. P. How. Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 285–292, 2017.

[21] Yuying Chen, Congcong Liu, Bertram E. Shi, and Ming Liu. Robot navigation in crowds by graph convolutional networks with attention learned from human gaze. *IEEE Trans. Robot. Autom.*, 5(2):2754–2761, 2020.

[22] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[23] 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.*, pages 3161–3167, June 2011. doi: 10.1109/CVPR.2011.5995558.

[24] N. P. Cuntoor, R. Collins, and A. J. Hoogs. Human-robot teamwork using activity recognition and human instruction. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 459–465, 2012.

[25] P. Dendorfer, H. Rezatofighi, A. Milan, J. Shi, D. Cremers, I. Reid, S. Roth, K. Schindler, and L. Leal-Taixé. Mot20: A benchmark for multi object tracking in crowded scenes. *arXiv:2003.09003[cs]*, March 2020. arXiv: 2003.09003.

[26] Agns Desolneux, Lionel Moisan, and Jean-Michel Morel. *From Gestalt Theory to Image Analysis: A Probabilistic Approach*. Springer Publishing Company, Incorporated, 1st edition, 2007. ISBN 0387726357.

[27] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, and et al. Long-term recurrent convolutional networks for visual recognition and description. In *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pages 2625–2634, June 2015.

[28] Yuqing Du, Nicholas J. Hetherington, Chu Lip Oon, Wesley P. Chan, Camilo Perez Quintero, Elizabeth Croft, and H.F. Machiel Van der Loos. Group surfing: A pedestrian-based approach to sidewalk robot navigation. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 6518–6524, 2019. doi: 10.1109/ICRA.2019.8793608.

[29] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei 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*, pages 226–231, 1996.

[30] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei 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*, pages 226–231, 1996.

[31] Michael Everett, Yu Fan Chen, and Jonathan P. How. Motion planning among dynamic, decision-making agents with deep reinforcement learning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Madrid, Spain, September 2018.

[32] D. Feil-Seifer and M. Matarić. People-aware navigation for goal-oriented behavior involving a human partner. In *Proceedings of the IEEE International Conference on Development and Learning (ICDL)*, volume 2, pages 1–6, 2011.

[33] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24 (6):381–395, jun 1981. ISSN 0001-0782.

[34] Irv Gadol. Beyond the hot seat: Gestalt approaches to group. *Int. J. of Group Psychotherapy*, 31(2):262–264, 1981. doi: 10.1080/00207284.1981.11492325.

[35] Yuxiang Gao and Chien-Ming Huang. Evaluation of socially-aware robot navigation. *Frontiers in Robotics and AI*, page 420, 2021.

[36] A. Garrell and A. Sanfeliu. Local optimization of cooperative robot movements for guiding and regrouping people in a guiding mission. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3294–3299, 2010.

[37] W. Ge, R. T. Collins, and R. B. Ruback. Vision-based analysis of small groups in pedestrian crowds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(5):1003–1016, 2012.

[38] G. Gennari and G. D. Hager. Probabilistic data association methods in visual tracking of groups. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages II–II, 2004.

[39] Christian Gloor. Pedsim: Pedestrian crowd simulation. *URL http://pedsim. silmaril. org*, 5 (1), 2016.

[40] Rachel Gockley, Jodi Forlizzi, and Reid Simmons. Natural person-following behavior for social robots. In *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 17–24, 2007.

[41] C. Granata and P. Bidaud. A framework for the design of person following behaviors for social mobile robots. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4652–4659, 2012.

[42] Vincent Le Guen and Nicolas Thome. Disentangling physical dynamics from unknown factors for unsupervised video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[43] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pages 2255–2264, June 2018.

[44] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social GAN: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2255–2264, 2018.

[45] D. Helbing. A Fluid Dynamic Model for the Movement of Pedestrians. *Complex Syst.*, 6(5): 391–415, 1992.

[46] Dirk Helbing and Péter Molnár. Social force model for pedestrian dynamics. *Phys. Rev. E*, 51:4282–4286, May 1995. doi: 10.1103/PhysRevE.51.4282.

[47] Dirk Helbing and Péter Molnár. Social force model for pedestrian dynamics. *Physical Review E*, 51(5):4282–4286, 1995.

[48] 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.*, pages 2470–2477, Nov 2011. doi: 10.1109/ICCV.2011.6126532.

[49] Moustafa Ibrahim, Srikanth Muralidharan, Zhiwei Deng, Arash Vahdat, and Greg Mori. A Hierarchical Deep Temporal Model for Group Activity Recognition. In *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognit.*, pages 1971–1980, Jun 2016.

[50] Boris Ivanovic and Marco Pavone. The trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs. In *Proc. IEEE/CVF International Conf. on Comput. Vis.*, pages 2375–2384, 2019.

[51] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(1):221–231, Jan 2013. ISSN 0162-8828. doi: 10.1109/TPAMI.2012.59.

[52] E. Jung, B. Yi, and S. Yuta. Control algorithms for a mobile robot tracking a human in front. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2411–2416, 2012.

[53] Haresh Karnan, Anirudh Nair, Xuesu Xiao, Garrett Warnell, Sören Pirk, Alexander Toshev, Justin Hart, Joydeep Biswas, and Peter Stone. Socially compliant navigation dataset (scand): A large-scale dataset of demonstrations for social navigation. *IEEE Robotics and Automation Letters*, 7(4):11807–11814, 2022.

[54] Linh Kästner, Cornelius Marx, and Jens Lambrecht. Deep-reinforcement-learning-based semantic navigation of mobile robots in dynamic environments. In *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, pages 1110–1115. IEEE, 2020.

[55] Linh Kästner, Teham Bhuiyan, Tuan Anh Le, Elias Treis, Johannes Cox, Boris Meinardus, Jacek Kmiecik, Reyk Carstens, Duc Pichel, Bassel Fatloun, et al. Arena-bench: A bench-

marking suite for obstacle avoidance approaches in highly dynamic environments. *IEEE Robotics and Automation Letters*, 7(4):9477–9484, 2022.

[56] Kapil Katyal, Yuxiang Gao, Jared Markowitz, I-Jeng Wang, and Chien-Ming Huang. Group-Aware Robot Navigation in Crowded Environments. *arXiv e-prints*, art. arXiv:2012.12291, December 2020.

[57] Yan Ke, R. Sukthankar, and M. Hebert. Efficient visual event detection using volumetric features. In *Proc. IEEE Int. Conf. Comput. Vis.*, volume 1, pages 166–173, Oct 2005. doi: 10.1109/ICCV.2005.85.

[58] A. Kendon. Conducting interaction : Patterns of behavior in focused encounters. *Studies in International Sociolinguistics*, 7, 1990.

[59] Sultan Daud Khan, Giuseppe Vizzari, Stefania Bandini, and Saleh Basalamah. Detection of social groups in pedestrian crowds using computer vision. In Sebastiano Battiato, Jacques Blanc-Talon, Giovanni Gallo, Wilfried Philips, Dan Popescu, and Paul Scheunders, editors, *Advanced Concepts for Intelligent Vision Systems*, pages 249–260. Springer International Publishing, Cham, 2015.

[60] Beomjoon Kim and Joelle Pineau. Socially adaptive path planning in human environments using inverse reinforcement learning. *International Journal of Social Robotics*, 8(1):51–66, 2016.

[61] Rachel Kirby. *Social Robot Navigation*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, May 2010.

[62] Rachel Kirby. *Social Robot Navigation*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, May 2010.

[63] Kris M. Kitani, Brian D. Ziebart, James Andrew Bagnell, and Martial Hebert. Activity forecasting. In Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *Computer Vision – ECCV 2012*, pages 201–214, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. ISBN 978-3-642-33765-9.

[64] K. Koffka. *Principles of Gestalt psychology*. Harcourt, Brace, 1935.

[65] Henrik Kretzschmar, Markus Spies, Christoph Sprunk, and Wolfram Burgard. Socially compliant mobile robot navigation via inverse reinforcement learning. *The International Journal of Robotics Research*, 35(11):1289–1307, 2016.

[66] Thibault Kruse, Amit Kumar Pandey, Rachid Alami, and Alexandra Kirsch. Human-aware robot navigation: A survey. *Robotics and Autonomous Systems*, 61(12):1726–1743, 2013.

[67] Masaki Kuribayashi, Tatsuya Ishihara, Daisuke Sato, Jayakorn Vongkulbhisal, Karnik Ram, Seita Kayukawa, Hironobu Takagi, Shigeo Morishima, and Chieko Asakawa. Pathfinder: Designing a map-less navigation system for blind people in unfamiliar buildings. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9781450394215. doi: 10.1145/3544548.3580687.

[68] Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. Crowds by example. *Comput. Graph. Forum*, 26(3):655–664, 2007. doi: 10.1111/j.1467-8659.2007.01089.x.

[69] Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. Crowds by example. *Comput. Graph. Forum*, 26(3):655–664, 2007.

[70] Junwei Liang, Lu Jiang, Kevin Murphy, Ting Yu, and Alexander Hauptmann. The garden of forking paths: Towards multi-future trajectory prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[71] S. Liang and R. Srikant. Why Deep Neural Networks for Function Approximation? In *Proc. Int. Conf. Learn. Represent.*, 2017.

[72] Shuijing Liu, Peixin Chang, Weihang Liang, Neeloy Chakraborty, and Katherine Driggs-Campbell. Decentralized structural-rnn for robot crowd navigation with deep reinforcement learning. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3517–3524. IEEE, 2021.

[73] Zhi Liu, Chenyang Zhang, and Yingli Tian. 3d-based deep convolutional neural network for action recognition with depth sequences. *Image and Vis. Comput.*, 55:93 – 100, 2016. ISSN 0262-8856. doi: https://doi.org/10.1016/j.imavis.2016.04.004.

[74] Sara Ljungblad, Jirina Kotrbova, Mattias Jacobsson, Henriette Cramer, and Karol Niechwiadowicz. Hospital robot at work: Something alien or an intelligent colleague? In *Proc.*

ACM Conf. on Comput. Supported Cooperative Work, pages 177–186, 2012. ISBN 978-1-4503-1086-4. doi: 10.1145/2145204.2145233.

[75] M. Luber, L. Spinello, J. Silva, and K.O. Arras. Socially-aware robot navigation: A learning approach. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 902–907, 2012.

[76] Barbara Majecka. Statistical models of pedestrian behaviour in the forum. *Master's thesis, School of Informatics, University of Edinburgh*, 2009.

[77] A. Makris and C. Prieur. Bayesian multiple-hypothesis tracking of merging and splitting targets. *IEEE Trans. Geosci. and Remote Sens.*, 52(12):7684–7694, Dec 2014. ISSN 0196-2892. doi: 10.1109/TGRS.2014.2316600.

[78] Karttikeya Mangalam, Yang An, Harshayu Girase, and Jitendra Malik. From goals, waypoints & paths to long term human trajectory forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15233–15242, 2021.

[79] Roberto Martin-Martin, Mihir Patel, Hamid Rezatofighi, Abhijeet Shenoi, JunYoung Gwak, Eric Frankel, Amir Sadeghian, and Silvio Savarese. Jrdb: A dataset and benchmark of egocentric robot visual perception of humans in built environments. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2021.

[80] E. A. Martinez-Garcia, Ohya Akihisa, and Shin'ichi Yuta. Crowding and guiding groups of humans by teams of mobile robots. In *Proceedings of the IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO)*, pages 91–96, 2005.

[81] Christoforos Mavrogiannis, Valts Blukis, and Ross A. Knepper. Socially competent navigation planning by deep learning of multi-agent path topologies. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6817–6824, 2017.

[82] Christoforos Mavrogiannis, Francesca Baldini, Allan Wang, Dapeng Zhao, Pete Trautman, Aaron Steinfeld, and Jean Oh. Core challenges of social robot navigation: A survey. *arXiv preprint arXiv:2103.05668*, 2021.

[83] Christoforos Mavrogiannis, Francesca Baldini, Allan Wang, Dapeng Zhao, Pete Trautman, Aaron Steinfeld, and Jean Oh. Core Challenges of Social Robot Navigation: A Survey. *arXiv e-prints*, art. arXiv:2103.05668, March 2021.

[84] Christoforos I. Mavrogiannis, Wil B. Thomason, and Ross A. Knepper. Social momentum: A framework for legible navigation in dynamic multi-agent environments. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction (HRI '18)*, pages 361–369. ACM, 2018.

[85] Christoforos I. Mavrogiannis, Alena M. Hutchinson, John Macdonald, Patrícia Alves-Oliveira, and Ross A. Knepper. Effects of distinct robotic navigation strategies on human behavior in a crowded environment. In *Proceedings of the 2019 ACM/IEEE International Conference on Human-Robot Interaction (HRI '19)*. ACM, 2019.

[86] R. Mazzon, F. Poiesi, and A. Cavallaro. Detection and tracking of groups in crowd. In *Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 202–207, 2013.

[87] Lyudmila Mihaylova, Avishy Y. Carmi, François Septier, Amadou Gning, Sze Kim Pang, and Simon Godsill. Overview of bayesian sequential monte carlo methods for group and extended object tracking. *Digit. Signal Process.*, 25:1 – 16, 2014. ISSN 1051-2004. doi: https://doi.org/10.1016/j.dsp.2013.11.006.

[88] Abduallah Mohamed, Kun Qian, Mohamed Elhoseiny, and Christian Claudel. Social-stgcnn: A social spatio-temporal graph convolutional neural network for human trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[89] Mehdi Moussaïd, Niriaska Perozo, Simon Garnier, Dirk Helbing, and Guy Theraulaz. The walking behaviour of pedestrian social groups and its impact on crowd dynamics. *PLOS ONE*, 5:1–7, 04 2010. doi: 10.1371/journal.pone.0010047.

[90] Bilge Mutlu and Jodi 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.*, pages 287–294, 2008. ISBN 978-1-60558-017-3. doi: 10.1145/1349822.1349860.

[91] A. Nanavati, X. Z. Tan, J. Connolly, and A. Steinfeld. Follow the robot: Modeling coupled human-robot dyads during navigation. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3836–3843, 2019.

[92] Haruki Nishimura, Boris Ivanovic, Adrien Gaidon, Marco Pavone, and Mac Schwager. Risk-sensitive sequential action control with multi-modal human trajectory forecasting for safe crowd-robot interaction. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11205–11212. IEEE, 2020.

[93] Sangmin Oh, Anthony Hoogs, Amitha Perera, Naresh Cuntoor, Chia-Chih Chen, Jong Taek Lee, Saurajit Mukherjee, JK Aggarwal, Hyungtae Lee, Larry Davis, et al. A large-scale benchmark dataset for event recognit. in surveillance video. In *CVPR 2011*, pages 3153–3160. IEEE, 2011.

[94] Billy Okal and Kai O Arras. Learning socially normative robot navigation behaviors with bayesian inverse reinforcement learning. In *2016 IEEE International Conf. on Robotics and Automation (ICRA)*, pages 2889–2895. IEEE, 2016.

[95] Diego Paez-Granados, Yujie He, David Gonon, Dan Jia, Bastian Leibe, Kenji Suzuki, and Aude Billard. Pedestrian-robot interactions on autonomous crowd navigation: Reactive control methods and evaluation metrics. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 149–156, 2022. doi: 10.1109/IROS47612.2022.9981705.

[96] A. K. Pandey and R. Alami. A step towards a sociable robot guide which monitors and adapts to the person's activities. In *2009 International Conference on Advanced Robotics*, pages 1–8, 2009.

[97] V. Patraucean, A. Handa, and R. Cipolla. Spatio-temporal video autoencoder with differentiable memory. November 2015.

[98] 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.*, pages 261–268, Sept 2009.

[99] 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.*, pages 261–268, Sept 2009. doi: 10.1109/ICCV.2009.5459260.

[100] Stefano Pellegrini, Andreas Ess, and Luc Van Gool. Improving data association by joint modeling of pedestrian trajectories and groupings. In Kostas Daniilidis, Petros Maragos, and

Nikos Paragios, editors, *Computer Vision – ECCV 2010*, pages 452–465, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. ISBN 978-3-642-15549-9.

[101] A. G. A. Perera, C. Srinivas, A. Hoogs, and G. Brooksby and. Multi-object tracking through simultaneous long occlusions and split-merge conditions. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, volume 1, pages 666–673, June 2006. doi: 10.1109/CVPR.2006.195.

[102] Sriyash Poddar, Christoforos Mavrogiannis, and Siddhartha S Srinivasa. From crowd motion prediction to robot navigation in crowds. *arXiv preprint arXiv:2303.01424*, 2023.

[103] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[104] Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Learning social etiquette: Human trajectory understanding in crowded scenes. In *Proc. Eur. Conf. Comput. Vis.*, pages 549–565, 2016. ISBN 978-3-319-46484-8.

[105] Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Learning social etiquette: Human trajectory understanding in crowded scenes. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 549–565, Cham, 2016. Springer International Publishing. ISBN 978-3-319-46484-8.

[106] Andrey Rudenko, Luigi Palmieri, Michael Herman, Kris M Kitani, Dariu M Gavrila, and Kai O Arras. Human motion trajectory prediction: A survey. *arXiv preprint arXiv:1905.06113*, 2019.

[107] Andrey Rudenko, Tomasz P Kucner, Chittaranjan S Swaminathan, Ravi T Chadalavada, Kai O Arras, and Achim J Lilienthal. Thör: Human-robot navigation data collection and accurate motion trajectories dataset. *IEEE Trans. Robot. Autom.*, 5(2):676–682, 2020.

[108] Amir Sadeghian, Vineet Kosaraju, Ali Sadeghian, Noriaki Hirose, Hamid Rezatofighi, and Silvio Savarese. Sophie: An attentive gan for predicting paths compliant to social and physical constraints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[109] Amir Sadeghian, Vineet Kosaraju, Ali Sadeghian, Noriaki Hirose, S. Hamid Rezatofighi, and et al. SoPhie: An Attentive GAN for Predicting Paths Compliant to Social and Physical Constraints. In *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognit.*, pages 1349–1358, Jun 2019.

[110] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*, pages 683–700. Springer, 2020.

[111] Christoph Schöller, Vincent Aravantinos, Florian Lay, and Alois Knoll. What the constant velocity model can teach us about pedestrian motion prediction. *IEEE Robotics and Automation Letters*, 5(2):1696–1703, 2020.

[112] Johannes L Schönberger. *Robust methods for accurate and efficient 3D modeling from unstructured imagery*. PhD thesis, ETH Zurich, 2018.

[113] Christoph Schöller, Vincent Aravantinos, Florian Lay, and Alois Knoll. What the constant velocity model can teach us about pedestrian motion prediction. *IEEE Robotics and Automation Letters*, 5(2):1696–1703, 2020.

[114] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Adv. Neural Inf. Process. Syst.*, pages 802–810. 2015.

[115] M. Shiomi, T. Kanda, S. Koizumi, H. Ishiguro, and N. Hagita. Group attention control for communication robots with wizard of oz approach. In *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 121–128, 2007.

[116] Tianmin Shu, Dan Xie, Brandon Rothrock, Sinisa Todorovic, and Song-Chun Zhu. Joint Inference of Groups, Events and Human Roles in Aerial Videos. In *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognit.*, pages 4576–4584, Jun 2015.

[117] Tianmin Shu, Sinisa Todorovic, and Song-Chun Zhu. CERN: Confidence-Energy Recurrent Network for Group Activity Recognition. In *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognit.*, pages 5523–5531, Jun 2017.

[118] Phani Teja Singamaneni, Anthony Favier, and Rachid Alami. Human-aware navigation planner for diverse human-robot ineraction contexts. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021.

[119] F. Solera, S. Calderara, and R. Cucchiara. Socially constrained structural learning for groups detection in crowd. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38 (5):995–1008, 2016.

[120] F. Solera, S. Calderara, E. Ristani, C. Tomasi, and R. Cucchiara. Tracking social groups within and across cameras. *IEEE Trans. Circuits Syst. Video Technol.*, 27(3):441–453, March 2017. ISSN 1051-8215. doi: 10.1109/TCSVT.2016.2607378.

[121] Agnieszka Sorokowska, Piotr Sorokowski, Peter Hilpert, Katarzyna Cantarero, Tomasz Frackowiak, and et al. Preferred interpersonal distances: A global comparison. *J. Cross-Cultural Psychol.*, 48(4):577–592, 2017.

[122] Lin Sun, Kui Jia, Dit-Yan Yeung, and Bertram E. Shi. Human action recognition using factorized spatio-temporal convolutional networks. In *Proc. IEEE Int. Conf. Comput. Vis.*, December 2015.

[123] Muchen Sun, Francesca Baldini, Peter Trautman, and Todd Murphey. Move beyond trajectories: Distribution space coupling for crowd navigation. *arXiv preprint arXiv:2106.13667*, 2021.

[124] Lei Tai, Jingwei Zhang, Ming Liu, and Wolfram Burgard. Socially compliant navigation through raw depth inputs with generative adversarial imitation learning. In *2018 IEEE International Conf. on Robotics and Automation (ICRA)*, pages 1111–1117, 2018.

[125] Angelique Taylor, Darren M Chan, and Laurel D Riek. Robot-centric perception of human groups. *ACM Transactions on Human-Robot Interaction*, 9(3):1–21, 2020.

[126] S. Thrun, M. Bennewitz, W. Burgard, A.B. Cremers, F. Dellaert, D. Fox, D. Hahnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz. MINERVA: a second-generation museum tour-guide robot. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 3, pages 1999–2005, 1999.

[127] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proc. IEEE Int. Conf. Comput. Vis.*, December 2015.

[128] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proc. IEEE Int. Conf. Comput. Vis.*, December 2015.

[129] P. Trautman and A. Krause. Unfreezing the robot: Navigation in dense, interacting crowds. In *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, pages 797–803, Oct 2010. doi: 10.1109/IROS.2010.5654369.

[130] Peter Trautman, Jeremy Ma, Richard M. Murray, and Andreas Krause. Robot navigation in dense human crowds: Statistical models and experimental studies of human-robot cooperation. *International Journal of Robotics Research*, 34(3):335–356, 2015.

[131] Nathan Tsoi, Mohamed Hussein, Jeacy Espinoza, Xavier Ruiz, and Marynel Vázquez. Sean: Social environment for autonomous navigation. In *Proc. 8th International Conf. on Human-Agent Interaction*, pages 281–283, 2020.

[132] J. Šochman and D. C. Hogg. Who knows who - inverting the social force model for finding groups. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*, pages 830–837, 2011.

[133] Jur van den Berg, Stephen J. Guy, Ming Lin, and Dinesh Manocha. Reciprocal n-body collision avoidance. In *Robotics Research*, pages 3–19. Springer Berlin Heidelberg, 2011.

[134] G. Varol, I. Laptev, and C. Schmid. Long-term temporal convolutions for action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(6):1510–1517, June 2018. ISSN 0162-8828. doi: 10.1109/TPAMI.2017.2712608.

[135] A. Vemula, K. Muelling, and J. Oh. Social attention: Modeling attention in human crowds. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 1–7, May 2018. doi: 10.1109/ICRA.2018.8460504.

[136] M. Vázquez and A. Steinfeld. An assisted photography method for street scenes. In *2011 IEEE Workshop on Applications of Computer Vision (WACV)*, pages 89–94, 2011.

[137] A. Wang and A. Steinfeld. Group split and merge prediction with 3D convolutional networks. *IEEE Robotics and Automation Letters*, 5(2):1923–1930, 2020.

[138] Allan Wang, Abhijat Biswas, Henny Admoni, and Aaron Steinfeld. Towards rich, portable, and large-scale pedestrian data collection. *arXiv preprint arXiv:2203.01974*, 2022.

[139] Allan Wang, Christoforos Mavrogiannis, and Aaron Steinfeld. Group-based motion prediction for navigation in crowded environments. In *Conf. on Robot Learning*, pages 871–882. PMLR, 2022.

[140] Allan Wang, Daisuke Sato, Yasser Corzo, Sonya Simkin, and Aaron Steinfeld. Tbd pedestrian data collection: Towards rich, portable, and large-scale natural pedestrian data, 2023.

[141] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, Deva Ramanan, Peter Carr, and James Hays. Argoverse 2: Next generation datasets for self-driving perception and forecasting. In J. Vanschoren and S. Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1. Curran, 2021.

[142] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019.

[143] Zuxuan Wu, Xi Wang, Yu-Gang Jiang, Hao Ye, and Xiangyang Xue. Modeling spatial-temporal clues in a hybrid deep learning framework for video classification. In *Proc. ACM Int. Conf. Multimedia*, pages 461–470, 2015. ISBN 978-1-4503-3459-4. doi: 10.1145/2733373.2806222.

[144] 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.*, pages 1186–1194, Mar 2018. doi: 10.1109/WACV.2018.00135.

[145] 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.*, pages 1345–1352, June 2011. doi: 10.1109/CVPR.2011.5995468.

[146] Zhi Yan, Tom Duckett, and Nicola Bellotto. Online learning for human classification in 3d lidar-based tracking. In *2017 IEEE/RSJ International Conf. on Intell. Robots and Syst. (IROS)*, pages 864–871. IEEE, 2017.

[147] Fangkai Yang and Christopher Peters. Social-aware navigation in crowds with static and dynamic groups. In *2019 11th International Conference on Virtual Worlds and Games for Serious Applications (VS-Games)*, pages 1–4, 2019.

[148] Ye Yuan, Xinshuo Weng, Yanglan Ou, and Kris M Kitani. Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9813–9823, 2021.

[149] Matteo Zanotto, Loris Bazzani, Marco Cristani, and Vittorio Murino. Online bayesian non-parametrics for group detection. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 111.1–111.12, 2012.

[150] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Proc. Eur. Conf. Comput. Vis.*, pages 818–833, 2014. ISBN 978-3-319-10590-1.

[151] H. Zender, P. Jensfelt, and G. M. Kruijff. Human- and situation-aware people following. In *Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 1131–1136, 2007.

[152] Pu Zhang, Wanli Ouyang, Pengfei Zhang, Jianru Xue, and Nanning Zheng. Sr-lstm: State refinement for lstm towards pedestrian trajectory prediction. In *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pages 12085–12094, June 2019.

[153] Pu Zhang, Wanli Ouyang, Pengfei Zhang, Jianru Xue, and Nanning Zheng. Sr-lstm: State refinement for lstm towards pedestrian trajectory prediction. In *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pages 12085–12094, June 2019.

[154] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. Bytetrack: Multi-object tracking by associating every detection box. *arXiv preprint arXiv:2110.06864*, 2021.

[155] Bolei Zhou, Xiaogang Wang, and Xiaoou Tang. Understanding collective crowd behaviors: Learning a mixture model of dynamic pedestrian-agents. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2871–2878, 2012. doi: 10.1109/CVPR.2012.6248013.

[156] Feng Zhu, Xiaogang Wang, and Nenghai Yu. Crowd tracking with dynamic evolution of group structures. In *Proc. Eur. Conf. Comput. Vis.*, pages 139–154, 2014. ISBN 978-3-319-10599-4.