CoMet: Modeling Group Cohesion for Socially Compliant Robot Navigation in Crowded Scenes

Adarsh Jagan Sathyamoorthy, Utsav Patel, Moumita Paul, Nithish K Sanjeev Kumar, Yash Savle, and Dinesh Manocha

Abstract—We present CoMet, a novel approach for computing a group’s cohesion and using that to improve a robot’s navigation in crowded scenes. Our approach uses a novel cohesion-metric that builds on prior work in social psychology. We compute this metric by utilizing various visual features of pedestrians from an RGB-D camera on-board a robot. Specifically, we detect characteristics corresponding to proximity between people, their relative walking speeds, the group size, and interactions between group members. We use our cohesion-metric to design and improve a navigation scheme that accounts for different levels of group cohesion while a robot moves through a crowd. We evaluate the precision and recall of our cohesion-metric based on perceptual evaluations. We highlight the performance of our social navigation algorithm on a Turtlebot robot and demonstrate its benefits in terms of multiple metrics: freezing rate (57% decrease), deviation (35.7% decrease), and path length of the trajectory (23.2% decrease).

I. INTRODUCTION

Mobile robots are increasingly being used in crowded scenarios in indoor and outdoor environments. Applications for these robots include surveillance, delivery, logistics, etc. In such scenarios, the robots need to navigate in an unobtrusive manner and also avoid issues related to sudden turns or freezing [1]. Moreover, the robots need to integrate well with the physical and social environments.

Extensive research in social and behavioral psychology suggests that crowds in real-world scenarios are composed of (social) groups. A group is generally regarded as a mesolevel concept and corresponds to two or more pedestrians with similar goals over a short or long period of time. As a result, the pedestrians or agents in a group exhibit similar movements or behaviors. It is estimated that up to 70% of observed pedestrians in real-world crowds are part of a group [2], [3]. Therefore, it is important to understand group characteristics and dynamics to perform socially-compliant robot navigation [4], [5], [6].

The problem of efficient robot navigation among pedestrians has been an active area of research. Most existing robot navigation algorithms consider walking humans or pedestrians as separate obstacles [7], [4], [6], [8]. Some techniques tend to predict trajectories of each pedestrian using learning-based methods [8] but do not account for the influence of group characteristics on individuals. This could lead to obtrusive trajectories that may cut through groups of friends or families. Other methods use simple and conservative methods to detect locally sensed clusters of pedestrians and compute paths around them [6]. However, they do not work well as the crowd density increases.

One characteristic of groups that could be utilized to address these problems is the social-cohesion or the collective behavior of the group’s members. This is directly linked to the inter-personal relationships between group members. For example, a group of friends or family has higher cohesion than a group of strangers [9], [10]. Cohesion is inversely related to the permeability of the group in social settings, i.e. whether another individual can cut through the group while walking [11]. Many theories have been proposed in psychology and sociology to identify the human behaviors or features that are good indicators of group cohesion. Such features include proximity between group members [12], walking speed [13], group size [11], context or environment [14], etc. Estimating cohesion could help a robot plan a better or more socially compliant trajectory based on the context. For example, in dense crowds (i.e. pedestrian density is more than 1 person/m²), the robot could navigate around a group that has high cohesion or the robot could move between members of a group that has low cohesion, similar to how humans navigate in crowded scenarios.

Main Contributions: We present a novel algorithm to...
perform socially compliant navigation in crowded scenes. Our approach uses perception algorithms to identify groups in a crowd using visual features. We also present a novel group cohesion metric and efficient algorithms to compute this metric in arbitrary crowds using deep learning. We combine our cohesion metric with learning-based techniques to generate trajectories that tend to follow the social norms. Some of the novel components of our approach include:

- We present CoMet, a novel metric for estimating group cohesion. Our approach is based on social psychology studies and exploits features such as proximity between people, walking speeds, group sizes, and interactions. Our method uses an RGB-D sensor to detect groups and these visual features. CoMet has a near 100% precision and recall when identifying low-cohesion groups, when evaluated in real-world pedestrian or crowd datasets.

- We present a novel CoMet-based navigation method that accounts for group cohesion, while ensuring social-compliance in terms of naturalness, large deviations and freezing behaviors. Our formulation assumes human cooperation in dense crowds and plans less conservative trajectories than prior methods. We prove that the deviation angles computed by our method are less than or equal to the deviation angles computed using a prior social navigation algorithm [6].

- We implement CoMet on a real Turtlebot robot equipped with a commodity RGB-D sensor and demonstrate improvements in terms of social navigation. Our qualitative evaluations in dense scenes indicate that CoMet accurately identifies the cohesion in different groups (see Fig.1). This enables the Turtlebot robot to navigate through a group based on our cohesion metric. Compared to prior social navigation algorithms, we demonstrate improved performance in terms of following metrics: freezing rate (up to 57% decrease), path deviation or turns (up to 35.7% decrease), and path length (up to 23.2% decrease).

II. RELATED WORKS

In this section, we briefly review prior work in robot navigation among crowds, pedestrian and group detection, and group interactions.

A. Clustering-based Group Detection

Many techniques have been proposed in computer vision for pedestrian and group detection in a crowd. The first step in group detection is to detect individuals in the images or videos. Methods for this step include many deep learning-based approaches for pedestrian detection and tracking [15] and improved methods for high density crowds [16]. These methods have been extended from individual pedestrian detection to group detection [17], [18]. These group-based methods typically use different kinds of clustering based on the proximities between people, their trajectories and their velocities to segregate them into groups [18], [19].

B. Group Behavior and Interaction Detection

Different techniques have been proposed for detecting group behaviors and interactions in computer vision and social psychology [14], [20] as well as event identification [21]. Behavior detection and event identification involve the analysis of different features (e.g., collectiveness, stability, uniformity) that represent how people move and interact in a crowd. These include individuals, groups, leaders, followers, etc. They also involve detecting scenarios where groups either merge together or split while walking or running.

Other relevant techniques detect interactions among people in a group based on F-formations [17], [22]. These algorithms estimate features such as people’s body and head poses and identify the individuals who are facing each other. Our approach is complimentary to these methods and extends them by using many other features, including proximity, walking, and interaction, to gauge group cohesion.

C. Robot Navigation and Social Compliance

Many recent works have focused on socially-compliant navigation [23], [4], [5], [6], [24], [25], [26]. The underlying goal is to design methods that not only compute collision-free trajectories but also comply with social norms that increase the comfort level of pedestrians in a crowd. At a broad level, the three major objectives of social navigation are comfort, naturalness, and high-level societal rules [23]. For example, a robot needs to avoid movements that are regarded as obtrusive to pedestrians by following rules related to how to approach and pass a pedestrian [4], [6]. Other techniques are based on modeling intra-group interactions [24] or by learning from real-world static and dynamic obstacle behaviors [27].

Many techniques for social navigation have been proposed based on reinforcement learning (RL) or inverse reinforcement learning (IRL). The RL-based methods [7], [28], [29], [6] mostly focus on treating each pedestrian as a separate obstacle to avoid collisions, sudden turns, or large deviations. IRL methods are driven by real-world natural crowd navigation behaviors [30], [31] and are used to generate trajectories with high levels of naturalness. However, they can result in unsafe trajectories and may not work well as the crowd density increases. Some methods model pedestrian behaviors by learning about their discrete decisions and the variances in their trajectories [32]. Other works have modeled human personality traits [25] or pedestrian dominance [33] based on psychological characteristics for trajectory prediction and improved navigation. Bera et al. [34] present an algorithm to avoid negative human reactions to robots by reducing the entitativity of robots. Our work on modeling group cohesion is complimentary to these methods.

III. BACKGROUND

In this section, we give an overview of prior work in social psychology, pedestrian tracking and robot navigation that is used in our approach. We also introduce the symbols and notation used in the paper.
A. Social Psychology

We use four features based on prior work in social psychology to estimate the cohesion of a group. We give a brief overview of each of these features.

Proximity: Proximity is chosen based on the proxemics principles established by Hall [12]. The underlying theory states that humans have an intimate space, a social and consultative space and a public space when interacting with others. We extend this idea to unstructured social scenarios where people or pedestrians walk, stand, or sit together. In general, humans maintain a closer proximity to other people with whom they closely interact (high cohesion). Many techniques have been proposed for simulating pedestrian movement [35] and collision queries [36].

Walking Speed: [13] studied individual and mixed gender groups’ walking speeds in a controlled environment and observed significantly slower speeds when people walk with their romantic partners. Assertion 1: A slower-than-average walking speed in a group indicates a close relationship between group members (high cohesion).

Group Size: [11] analyzed the perceptions of people when passing through a group of two and four people in a university hallway. It was observed that people tend to penetrate through the 4-person group less than the 2-person group. Assertion 2: Humans perceive the cohesion of a bigger group to be higher (implying lower permeability) than that of a smaller group. Permeability of a group is a measure of the resistance that a moving non-group entity faces while passing between the members of a group.

Interactions: Jointly Focused Interactions (JFI) [37] entail a sense of mutual activity and engagement between people and imply their willingness to focus their attention on others. Therefore, it is chosen as an indicator for cohesion. We extrapolate JFI to signify a higher level of cohesion between group members. Visually, interactions can be detected by estimating peoples’ 3-D face vectors [22] and detecting when the vectors point towards each other.

B. Notations and Definitions

We highlight the symbols and notation used, in Table I. We use $i, j,$ and $k$ to represent indices. All distances, angles and velocities are measured relative to a rigid coordinate frame attached to the camera (on the robot) used to capture the scene. The X-axis of this frame points outward from the camera and the Y-axis points to the left, with its origin at the center of the image. We use such a representation since our overall approach is local, and has no global knowledge of the environment.

The time interval between two consecutive RGB images in the stream is $\Delta t$.

C. Frozone

Frozone [6] is a navigation method that tackles the Freezing Robot Problem (FRP) [1] arising in crowds. At the same time, it can generate trajectories that are less obtrusive to pedestrians. The underlying algorithm computes a Potential Freezing Zone (PFZ), which corresponds to a configuration of obstacles where the robot’s planner halts the robot and starts oscillating for a period of time when it deems that all velocities could lead to a collision. Frozone’s formulation is conservative and not suitable for dense crowd navigation (> 1 person/m$^2$). In addition, it treats pedestrians as individual obstacles. The main steps in Frozone’s formulation are:

1. Identifying potentially freezing pedestrians who could cause an FRP based on their walking vectors and predicting their positions after a time horizon $t_h$;
2. Constructing a Potential Freezing Zone ($PFZ_{froz}$) as the convex hull of the predicted positions $\hat{p}_{froz}^{i,t+h}$ (see $S$-sided convex polygon in Fig. 3) for all the potentially freezing pedestrians. $PFZ_{froz}$ is formulated as,

$$PFZ = ConvexHull(p_{froz}^{i,t+h}), \quad i \in 1, 2, ..., P_f.$$ (1)

and (3) Computing a deviation angle for the robot to avoid $PFZ_{froz}$ if its current trajectory intersects with it. $PFZ_{froz}$ corresponds to the set of locations where the robot has the maximum probability of freezing and being obstructive to the pedestrians around it. The deviation angle to avoid it is computed as

$$\phi_{froz} = \min(\phi_1, \phi_2),$$ (2)

where $\phi_1$ and $\phi_2$ are given by,

$$\phi_1 = \arg\min_{R_z, \phi_1, v_{rob}, t_h \in PFZ_{froz}} (dist(R_z, \phi_1, v_{rob}, t_h, g_{rob})), \quad (3)$$

$$\phi_2 = \tan^{-1}(y_{near,t}/x_{near,t}), \quad \phi_2 \neq 0.$$ (4)

Here, $R_z, \phi_1$ is the 3-D rotation matrix about the Z-axis (perpendicular to the plane of the robot), $v_{rob}, g_{rob}$ represent the current velocity and the goal of the robot, and $[y_{near,t}, x_{near,t}]$ denotes the current location of the nearest freezing pedestrian relative to the robot. This point is in the PFZ’s exterior. For navigating the robot towards its goal, and handling static obstacles and dense crowds, a Deep
algorithms to detect and track people, respectively. YOLOv5
position relative to the camera frame in a continuous stream
III-A is to first detect, track, and localize each pedestrian
B. Pedestrian Tracking and Localization

In this section, we first describe how we track and localize
people, detail conditions for a set of people to be classified
A. Detecting Group Features

In this section, we present our group cohesion metric,
which first classifies pedestrians into groups and then mea-
Computation of Group Cohesion Metric: We use four
features to compute the metric. We highlight how our approach is
used on two different groups of pedestrians at a given time instant.
These groups are shown in different rows. Group 1 consists of two
pedestrians in close proximity, walking away from each other. Their
distance is increasing in subsequent frames and they are looking
in different directions. This implies little interaction and low group
cohesion. Group 2 has two people walking together with their faces
oriented towards each other, which indicates high interaction and a
high group cohesion.

Reinforcement Learning (DRL)-based method [29] is used.
However, the resulting navigation may cut through groups
oriented towards each other, which indicates high interaction and a
cohesion. Group 2 has two people walking together with their faces
in different directions. This implies little interaction and low group

IV. CoMET: Modeling Group Cohesion

In this section, we present our group cohesion metric,
which first classifies pedestrians into groups and then
measures their closeness or cohesion. Our method runs in real-
time, taking a continuous stream of RGB and depth images as
input and detecting the group features highlighted in Section
III-A. Our overall approach based on these features is shown in
Fig. 2

A. Detecting Group Features

In this section, we first describe how we track and localize
people, detail conditions for a set of people to be classified
as a group, and then explain efficient techniques to detect
group features from RGB and depth images.
B. Pedestrian Tracking and Localization

A key issue in detecting the features mentioned in Section
III-A is to first detect, track, and localize each pedestrian
position relative to the camera frame in a continuous stream
of RGB images. We use YOLOv5 [38] and Deep Sort [39]
algorithms to detect and track people, respectively. YOLOv5
outputs a set of bounding boxes \( \mathcal{B} = \{ \mathcal{B}^i \} \) for each detected
pedestrian \( i \) in an RGB image at time instant \( t \) (denoted as \( I_{rgb}^t \)). \( \mathcal{B}^i \) is denoted using its top-left and bottom-right
corners in the image-space or pixel coordinates. In addition, we also assign a unique integer number as an ID for each detected pedestrian.

Next, to accurately localize people, the distance of each
detected person relative to the camera coordinate frame must
be estimated. To this end, we use a depth image \( I_{depth}^t \), every
pixel of which contains the proximity (in meters) of an object
at that location of the image. The pixels in \( I_{depth}^t \) contain
values between a minimum and maximum distance range,
which depends on the camera used to capture the image.

1) Group Classification: Let us consider any set of people’s IDs \( G^{k,t} \subseteq I^t \). At any time \( t \), if the following conditions hold,

\[
\| p_i^{t,t} - p_j^{t,t} \|_2 \leq \Gamma \quad \forall i,j \in G^{k,t} \\
\| p_i^{t,t} - p_j^{t,t} \|_2 \geq \| p_i^{t,t} + v_i^{t,t} - (p_j^{t,t} + v_j^{t,t}) \|_2.
\]

then the set \( G^{k,t} \) is classified as a group in the image
\( I_{rgb}^t \). Here, \( \Gamma \) is a distance threshold set manually. The first
condition ensures that people are close to each other and the second condition ensures that the group members walk in the same direction. When \( |v_i^t|, |v_j^t| = 0 \) (static groups),
only the first condition is used for grouping.

2) Estimating Proximity: To estimate the proximity be-
tween people at time instant \( t \), first the bounding boxes
detected in the RGB image by YOLOv5 are superimposed
over the depth image. To estimate the distance of a person \( i \) from the camera ((\( d_i^t \)), the mean of all the pixel values within
a small square centered around \([x_{cen}^t, y_{cen}^t] \) is computed. The angular displacement \( \psi_i^t \) of person \( i \) relative to the
camera can be computed as, \( \psi_i^t = \left( \frac{x_{cen}^t - x_i^t}{w} \right) \ast \text{FOV}_{RGB} \).

Here \( \text{FOV}_{RGB} \) is the field of view of the RGB-D camera. Person \( i \)’s location relative to the camera can be estimated as:
\( [x_i^t, y_i^t] = d_i^t \ast \left[ \cos \psi_i^t \sin \psi_i^t \right] \).
The distance between a pair of people \( i \) and \( j \) can then be computed as,
\( \text{dist}(i, j) = \sqrt{(x_i^t - x_j^t)^2 + (y_i^t - y_j^t)^2} \).

3) Estimating Walking Speed and Direction: To estimate
the \( i \)th person’s walking vector \( v_i^t \) in \( I_{rgb}^t \), we use a Kalman
filter with a constant velocity motion model. All the detected
people in \( I_{rgb}^t \) (with their IDs stored in the set \( I^t \)) are modeled using the state vector \( x_i^t \) defined in Table I.
If \( I^t \) contains IDs that were not present in \( I^{t-\Delta t} \), we initialize
their corresponding state vectors \( x_i^t \) with constant values.
For all the pedestrians who were detected in previous RGB
images, i.e., with previously initialized states, we update their
states using the standard Kalman prediction and update the
steps [40]. We use a zero mean Gaussian noise with a pre-set
variance to model the process and sensing noise.

4) Estimating Group Size: The size of the group can be
trivially computed as the number of IDs in the set \( G^{k,t} \).

5) Detecting Interactions: We use two 3-D vectors to
represent the position and orientation of a person’s face
in \( I_{rgb}^t \). We use OpenFace [41] to localize person i’s face
position relative to the camera coordinate frame (\( f_i^t \)) on an
RGB image and to obtain a unit vector (\( f_i^t \)) for the face
orientation. Two individuals are considered to be interacting
if their face positions and orientations satisfy the following condition:

\[
\| f_p^t - f_p^t \|_2 > \| f_p^t + f_o^t - (f_p^t + f_o^t) \|_2.
\]
This condition checks if the distance between two people’s face positions is greater than the distance between the points computed by extrapolating the face orientations (see dashed lines in Fig. [1]), if they are facing each other. We reasonably assume that non-interacting people do not face each other.

C. Cohesion Components

We now discuss how the detected group features can be used for cohesion estimation. Using multiple features to compute cohesion is advantageous in scenarios where not all features are properly able to be sensed.

1) Proximity Cohesion Score: We use the average distance between group members to model Hall’s proxemics theory as previously extrapolated. As observed in Section III-A, cohesion between people is inversely proportional to the distance between them. Therefore, the cohesion score due to proximity is formulated as the reciprocal of the mean distance between group members as

\[ C_p(G^{k,t}) = K_p \cdot \frac{\sum_{i,j \in G^{k,t}; i \neq j} dist(i, j)}{n^k}. \]  

(8)

2) Walking Speed Cohesion Score: Based on the discussion in Section III-A, we next compare the average walking speeds of each group with the average walking speed of all the detected people in \( I_{tr}^{gb} \). Therefore, the cohesion score for a walking group (\( \|v^i\| \neq 0 \forall j \in G^{k,t} \)) due to its walking speed is formulated as

\[ C_w(G^{k,t}) = K_w \cdot \frac{\sum_{v \in I_{tr}^{gb}} \|v^i\|}{N} / \left( \frac{\sum_{j \in G^{k,t}} \|v^j\|}{n^k} \right). \]  

(9)

This reflects assertion 1 made in Section III-A since cohesion is inversely proportional to walking speed. The average walking speed of the entire scene is included in this formulation to normalize out the effects of crowding in the scene. If \( \sum_{j \in G^{k,t}} \|v^j\| = 0 \), i.e., the group is static, then \( C_w(G^{k,t}) = K_w \cdot \eta \), where \( \eta \) is a user-set large constant value.

3) Group Size Cohesion Score: Based on assertion 2 in Section III-A, the cohesion of a group \( k \) is directly proportional to the group size \( (n^k) \). Therefore, the group size cohesion score is computed as

\[ C_s(G^{k,t}) = K_s \cdot n^k. \]  

(10)

4) Interaction Cohesion Score: The interaction condition between any two people in a group (Equation [7]) can be applied to all pairs in a group, and its contribution to the cohesion score of a group can be re-written as

\[ C_i(G^{k,t}) = K_i \cdot \frac{1}{n^k} \cdot \sum_{i \neq j \in G^{k,t}} \frac{\text{sign}(\theta_{ij})}{\cos \theta_{ij}} \theta_{i,j} \in [-\frac{\pi}{4}, \frac{\pi}{4}]. \]  

(11)

Here \( \theta_{ij} \) is the angle between face orientation vectors \( \mathbf{f}_o^i \) and \( \mathbf{f}_o^j \) in the X-Y plane of the camera coordinate system. \( \text{sign}() \) is the signum function. \( \theta_{ij} \) is limited to \([-\frac{\pi}{4}, \frac{\pi}{4}] \) since face orientations are accurate in this range. Intuitively, we want the cohesion score to be positive and greater than 1, when people are facing towards each other and negative otherwise.

Therefore, we choose the ratio \( \frac{\text{sign}(\theta_{ij})}{\cos \theta_{ij}} \), as it belongs to the range \([-\sqrt{2}, -1] \cup \{0\} \cup [1, \sqrt{2}] \) when \( \theta_{ij} \in [-\frac{\pi}{4}, \frac{\pi}{4}] \). Since \( \cos() \) is an even function, the \( \text{sign}() \) function ensures that the formulation is sensitive to the sign of the angle.

D. CoMet: Overall Group Cohesion Metric

Using the individual cohesion scores in Equations 8, 9, 10, 11, the total cohesion score for a group at time \( t \) \( G^{k,t} \) is given as

\[ C_{tot}(G^{k,t}) = (C_p + C_w + C_s + C_i). \]  

(12)

Here, \( G^{k,t} \) is omitted in the RHS for readability. Note that \( K_p, K_w, K_s, K_i \) weigh the different features before adding them to the total cohesion score. If any of the features are not detectable, their contribution to \( C_{tot} \) will be zero. This acts as a measure of confidence, as our approach is able to better compute a group’s cohesion when more features are detected. Our formulation is not learning-based due to the lack of extensive datasets with annotations of cohesion or related metrics for groups.

Proposition IV.1. The value of the overall cohesion metric \( C_{tot}(G^{k,t}) \) for a group is bounded.

Proof. The proof to the proposition follows from the fact that \( C_p, C_w, C_s, C_i \) are bounded. The value of \( C_p \in (0, K_p, \Gamma] \), since \( \Gamma \) is used as a threshold to group people. The maximum value of \( C_w \) is \( K_w \cdot \eta \), a large finite constant that is used when a group is static. \( C_s \) is bounded above by \( K_g \cdot \eta \), which is finite. \( C_i \in [-\sqrt{2}K_i, -K_i) \cup [K_i, \sqrt{2}K_i] \).

We use these bounds on the cohesion metric to compute appropriate thresholds that are used to categorize groups as low-, medium- and high-cohesion groups.

V. COHESION-BASED NAVIGATION

In this section, we present our socially-compliant navigation algorithm, which uses the group cohesion metric.

A. Socially-Compliant Navigation

Our objective is to improve the naturalness of a robot’s trajectory. We attribute three qualities to natural trajectories: (1) Not suddenly halting or freezing (avoiding FRP), (2) Low deviation angles, (3) Not cutting between high cohesion groups while walking suddenly halt or significantly deviate from their goals. This is in accordance with humans’ walking behaviors where people do not dance with humans’ walking behaviors where people do not.

We extend Frozone [6] (Section III-C) by considering groups and their cohesions, and prove that our proposed method leads to smaller deviations from the robot’s goal, and shorter trajectory lengths. It also does not navigate the robot through high cohesion groups. We assume a higher density in the environment (in terms of crowds and static obstacles) than Frozone’s formulation and human cooperation for the
Here, \( \mathbf{v}_{\text{avg}}^{G_{k,t}} \) is the average group walking vector, computed as the mean of the walking vectors of the group members and, \( M \) is the total number of potentially freezing groups.

Definition [2] (Group PFZ) The region in the vicinity of a group where the robot has a high probability of freezing. Instead of constructing the single PFZ as the convex hull of all potentially freezing pedestrians (like in [6]), we construct a PFZ for each potentially freezing group (see Fig. 3) as

\[
PFZ^k = CH(\mathbf{p}_{pf}^{i,t+t_h}), \quad i \in G^{k,t}, \quad k \in \{1, 2, ..., M\}. \tag{14}
\]

Every potentially freezing group’s PFZ is computed for a future time instant \( t + t_h \).

1) Computing Deviation Angle: If the robot’s current trajectory navigates it into any of the group PFZs (implying an occurrence of FRP after time \( t_h \)), a deviation angle \( \phi_{\text{com}} \) to avoid it is computed. The robot’s current velocity \( \mathbf{v}_{\text{rob}} \) is deviated by \( \phi_{\text{com}} \), using a rotation matrix about the Z-axis as,

\[
\mathbf{v}'_{\text{rob}} = \bar{R}_z \cdot \mathbf{v}_{\text{rob}}, \quad \phi_{\text{com}} = \arg\min_{\mathbf{v}'_{\text{rob}} \in PFZ} \text{dist}(\mathbf{v}'_{\text{rob}} \cdot t_h, \mathbf{g}_{\text{rob}}). \tag{15}
\]

This equation implies that our navigation method deviates the robot by the least amount from its goal such that it does not enter any group’s PFZ. However, in dense scenarios, when the robot encounters many potentially freezing groups and their corresponding PFZs, Equation 16 may not be able to compute a feasible solution for \( \phi_{\text{com}} \). In such cases, a potential solution is to let the robot pass through the PFZ of a low cohesion group (see Fig. 6c). In such cases, we formulate the deviation angle as,

\[
\phi_{\text{com}} = \arg\min_{\mathbf{v}'_{\text{rob}} \in \mathcal{P}} \text{dist}(\mathbf{v}'_{\text{rob}} \cdot t_h, \mathbf{g}_{\text{rob}}), \quad \mathcal{P} = PFZ_{\text{min}} \setminus (PFZ_{\text{min}} \cap PFZ^k) \quad \forall k \in \{1, 2, ..., M\}, \tag{17}
\]

where \( PFZ_{\text{min}} \) is the PFZ of the group with the minimum cohesion in the scene. Since the permeability of low cohesion groups is high, the above formulation also lowers the probability of freezing.

Proposition V.1. The deviation angles computed by CoMet-based navigation (Equations 16 or 18, and Frozone 2) satisfy the relationship \( \phi_{\text{froz}} \leq \phi_{\text{com}} \).

Proof. Consider the scenario shown in Fig. 3 with \( PFZ_{\text{froz}} \) depicted as a 5-sided convex polygon. A vertical line segment connects the robot (\( O_{\text{rob}} \)) to its goal and the robot’s deviation is measured relative to it. Frozone deviates \( \mathbf{v}_{\text{rob}} \) such that this point lies on the boundary of \( PFZ_{\text{froz}} \) (Equation 5) or in the exterior of \( PFZ_{\text{froz}} \) (Equation 4), depending on whichever leads to a lower deviation. Let \( O_{\text{froz}} = [x_{\text{froz}}, y_{\text{froz}}] \) be the point on the boundary of \( PFZ_{\text{froz}} \) to which Frozone deviates (by angle \( \phi_{\text{froz}} \)) the robot. Based on Equations 16 and 18 our CoMet-based solution deviates the robot to a point in \( PFZ_{\text{froz}} \) as

\[
\mathbf{p}_{\text{froz}} = \mathbf{p}_{\text{com}} + \mathbf{v}_{\text{com}}^{G_{k,t}} \quad \forall k \in \{1, 2, ..., M\}, \tag{19}
\]
In dense scenarios, when there is no feasible solution for the deviation angle, our method navigates the robot through the group with the lowest cohesion. All these behaviors improve the naturalness of the robot trajectory’s.

Since, \[ \bigcup_{k=1}^{M} PFZ^k \subseteq PFZ_{froz}, \] there exists a set \( F \subseteq PFZ_{froz} \) such that all positions in \( F \) lead to a deviation angle \( \phi \leq \phi_{froz}. \) For instance, in Fig. 5, \( F \) can be a set just within the boundary of \( PFZ_{froz} \) with Y-coordinates greater than \( y_{froz}. \) Since Equations (16) and (18) optimize for minimum deviation from the goal, \( \phi_{com} \in F. \) This implies that \( \phi_{com} \leq \phi_{froz}. \) The equality holds when \( F = \emptyset \) or when the closest edge of \( PFZ_{froz} \) to the robot corresponds to the PFZ of a group. Based on the triangle inequality, shorter deviations lead to shorter trajectory lengths.

This bound also guarantees that our new navigation algorithm generates trajectories that are more natural than Frozone [6]. We integrate our CoMet-based navigation method with a DRL-based navigation scheme to evaluate it in real-world scenarios[29] and DWA [43] to evaluate it in simulated environments. Figure 4 shows the components of our navigation algorithm used to compute trajectories that are more natural in real-world scenes.

VI. IMPLEMENTATION AND RESULTS

In this section, we describe the implementation for computing group cohesion and socially-compliant navigation. We then evaluate CoMet in different standard pedestrian datasets that are annotated with perceived group cohesion levels. We highlight the performance of our navigation algorithm and show the benefits over prior methods in terms of the following quantitative metrics: freezing rate, deviation angle, and normalized path-length. We also qualitatively compare trajectories with an increased number of obstacles in the environment.

A. Implementation

In order to evaluate CoMet, we annotate groups in pedestrian datasets such as MOT, KITTI, ETH, etc. as low-, medium- and high-cohesion groups with the help of multiple human annotators. These annotated datasets are used as the ground truth since they reflect how humans perceive the groups in the videos. We choose these datasets because they depict groups in real-world scenarios with various lighting conditions, crowd densities and occlusions. We use a depth estimation method [44] with RGB images in the datasets to localize different pedestrians in the scene. We manually tune the weighing constants in the CoMet formulation \( (K_p, K_w, K_s, K_i) \) and set thresholds on the cohesion score to classify groups into the aforementioned categories based on the annotations in one of the datasets. We evaluate CoMet’s precision and recall in the groups in all other datasets. Since there are no prior methods that compute cohesion, our evaluation is only against the human annotations.

We have evaluated our navigation algorithm using simulations created in MATLAB: (i) with tens to hundreds of groups, each with two to five members (Fig 5) and a pre-assigned cohesion metric for each group and (ii) in corridor-like constrained scenarios with tens of pedestrians. The groups move linearly to a goal, and the simulated robot must take full responsibility to avoid collisions with them. We also evaluate our method on a real Turtlebot 2 robot mounted with an Intel RealSense RGB-D camera (for pedestrian tracking and localization) and a 2-D Hokuyo lidar (used by the DRL [29] method).

B. Analysis

**CoMet Classification:** Table II highlights CoMet’s classification precision and recall in multiple datasets. CoMet’s parameters were tuned to improve its accuracy corresponding to detecting low-cohesion groups, which is required for navigation. This is reflected in the high precision and recall values in the second column. CoMet observes pedestrians in these datasets for ~ 5 seconds. During this period, it is able to update its initial classification as pedestrians’ trajectories change. For instance, a group initially perceived as high-cohesion may have its members move apart and is thereby classified as a low-cohesion group. Moreover, it is easier to detect features corresponding to proximity and walking speed than interactions between the pedestrians. This sometimes results in CoMet misclassifying medium- and high-cohesion groups in certain datasets. An interesting observation is that human annotators tend to classify groups in extremes, i.e. as either high-cohesion or low-cohesion groups. This leads to a low number of data points for medium-cohesion groups. This ground truth observation affects the effectiveness of our approach.

**Socially-Compliant Navigation:** We first qualitatively compare the trajectories (Fig 5) of DWA [43] (in red), a DWA-Frozone hybrid [6] (in blue), and DWA and CoMet-based navigation (in green) in simulated environments with non-cooperative walking groups of obstacles. We observe that for the same set of dynamic obstacles, our approach computes smaller deviations while also avoiding collisions. DWA and the DWA-Frozone hybrids lead to conservative, and highly sub-optimal deviations from the goal direction.
Fig. 5: Qualitative evaluations of the trajectories generated using our algorithm (green), Frozone [6] (blue) and DWA [43] (red) in scenarios with 50, 75 and 100 moving groups (red dots) that are non-cooperative for collision avoidance. The green and yellow squares represent the sensing regions of Frozone and CoMet-based navigation respectively. We observe that our formulation leads to lower deviation angles and more natural trajectories even in dense environments with tens to hundreds of obstacles. Both Frozone and DWA lead to unnatural trajectories with large deviations as the number of obstacles in the environment increases.

Fig. 6: Qualitative evaluations of the trajectories generated using our algorithm (shown as green) in different scenarios. We also compare with the trajectories generated using Frozone [6] (shown as yellow) and a DRL-based algorithm [29] (shown as orange). Each group’s PFZ is shown as a red region on the floor. We evaluate our method in three different real-world scenarios with tight spaces, with people standing or sitting. Our method differentiates between low (in green) and high cohesion (in red) groups, and navigates only between low cohesion groups. Frozone algorithm behaves in a conservative manner and halts the robot in dense scenarios. DRL [29] prioritizes moving towards the goal and passes through high cohesion groups (see (a)). Overall, our approach results in socially-compliant trajectories.

![Figures showing different scenarios](image)

**TABLE II:** This table shows the precision (first value in each column) and recall (second) values for the three classes when CoMet observes groups in the video for ~5 seconds. CoMet’s parameters have been tuned based on the ground truth in one of the datasets. We observe good accuracy for high-cohesion and low-cohesion groups. The original datasets have fewer occurrences of medium-cohesion groups, which impacts our precision.

| Dataset   | Video      | Low-Cohesion | Medium-Cohesion | High-Cohesion |
|-----------|------------|--------------|-----------------|--------------|
| ADL-Rundle| 1.00/1.00  | 0.50/0.67    | 0.50/0.33       |
| AVG-TownCenter | 1.00/1.00  | 1.00/0.80    | 0.50/0.33       |
| ETH Jelmoli | 1.00/1.00  | 1.00/0.67    | 0.50/0.10       |
| ETH Rämistrasse | 1.00/1.00 | 1.00/0.30    | 0.50/0.10       |
| KIT17-16   | 1.00/1.00  | 1.00/0.60    | 0.50/0.67       |
| KIT17-17   | 1.00/1.00  | 1.00/0.50    | 0.50/0.33       |
| MOT17-08   | 1.00/1.00  | -            | 0.50/0.33       |
| MOT17-11   | 1.00/1.00  | 0.75/0.00    | 0.50/0.33       |
| MOT20-02   | 1.00/1.00  | 0.34/0.33    | 0.50/0.33       |
| TUD        | 1.00/0.857 | -            | 0.50/0.33       |
| Venice     | 0.87/0.875 | -            | 0.93/0.60       |

Although the exact set of obstacles each robot faces could be different depending on the trajectories they take, we observe that our method’s deviations at any instant never exceed the deviations of the other two methods in comparison. These results reinforce proposition [V.1]

We also quantitatively compare the aforementioned three methods in environments with varying numbers (10 – 50) of pedestrians in a corridor-like scenario (which constrains the free space available to the robot). Pedestrians are given random initial locations and velocities, based on which they are classified into groups and $PFZ_{froz}$ and group PFZs ($PFZ_k$) are computed. The robot needs to navigate through the pedestrians to reach its goal. We use the following metrics: (1) Average deviation angle measured relative to the line connecting the start and goal locations; (2) Freezing Rate measured as the number of times the robot halted/froze over the total number of trials; and (3) Normalized Path Length measured as the robot’s path length over the length of the line connecting the start and goal locations.

Our method results in lower values with respect to all these metrics, as compared to DWA and the DWA-Frozone hybrid on the same scenarios. As the crowd size, density, or number of groups increase, Frozone’s conservative formulation makes the robot freeze at a high rate. This is because Frozone forbids the robot from avoiding a robot from in-front of the person, to improve pedestrian-friendliness in low- to medium-density scenes. On the other hand, we observe that our method produces trajectories with high social compliance and naturalness by reducing the occurrence of freezing behavior.

**Real-world Evaluations:** We qualitatively compare our method’s trajectories with Frozone’s and a DRL algorithm’s [29] trajectories. We highlight the differences in Fig. ![Figures showing real-world evaluations](image) Our approach is able to identify low-cohesion groups suc-
TABLE III: We compare different navigation methods (DWA [43], DWA-Frozone hybrid [6] and DWA-CoMet-based navigation algorithm based on three metrics. We observe that our method consistently results in lower values corresponding to all these metrics. This signifies improved naturalness of the robot’s trajectory computed using our approach.

cessfully and navigate through them without interfering with high-cohesion groups. In contrast, Frozone halts the robot completely, since it does not assume pedestrian cooperation in its formulation. The DRL method prioritizes reaching the goal with the minimum path length and therefore navigates through groups regardless of their cohesion. Therefore, the DRL algorithm can generate obtrusive trajectories.

C. Other Improvements using CoMet

Existing high-fidelity simulators for training Deep Reinforcement Learning methods for navigation simulate dynamic pedestrians as individual obstacles [45]. Large-scale simulators [46] use well-known motion models for individual pedestrians, and do not take into account model group behaviors. CoMet can be used to reverse engineer and simulate group behaviors based on different cohesion scores. For instance, simulated groups with high cohesion can have low intra-group member proximities, lower than average walking speeds, and larger group sizes. In 3-D simulators with human models, interactions can be modeled based on body orientations.

From an HRI standpoint, in social-distance monitoring robots [47], CoMet could help predict group properties such as inter-personal relationships between members based on their cohesion. This would help identify the groups that need to be issued warnings regarding maintaining social distancing. This leads to more apt interactions between the robot and the humans.

We present a novel method to compute the cohesion of a group of people in a crowd using visual features. We use our cohesion metric to design a novel robot navigation algorithm that results in socially-compliant trajectories. We highlight the benefits over previous algorithms in terms of the following metrics: reduced freezing, deviation angles, and path lengths. We test our cohesion metric in annotated datasets and observe a high precision and recall.

Our method has some limitations. We model cohesion through a linear relationship between the features, which may not work in all scenarios. In addition, there are other characteristics used to estimate cohesion, including age, gender, environmental context, cultural factors, etc. that we do not consider. Our approach also depends on the accuracy of how these features are detected, which may be affected due to lighting conditions and occlusions. Our navigation assumes that different groups exhibit varying levels of cohesions, which may not hold all the time. As part of future work, we hope to address these limitations and evaluate our approach in crowded real-world scenes.

REFERENCES

[1] P. Trautman and A. Krause, “Unfreezing the robot: Navigation in dense, interacting crowds,” in 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Oct 2010, pp. 797–803.

[2] L. He, J. Pan, S. Narang, and D. Manocha, “Dynamic group behaviors for interactive crowd simulation,” in Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation. Eurographics Association, 2016, p. 139–147.

[3] S. Bandini, A. Gorrini, L. Manenti, and G. Vizzari, “Crowd and pedestrian dynamics: Empirical investigation and simulation,” in Proceedings of Measuring Behavior, 08 2012.

[4] Y. F. Chen, M. Everett, M. Liu, and J. How, “Socially aware motion planning with deep reinforcement learning,” in 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 09 2017, pp. 1343–1350.

[5] L. Tai, J. Zhang, M. Liu, and W. Burgard, “Socially compliant navigation through raw depth inputs with generative adversarial imitation learning,” in ICRA, May 2018, pp. 1111–1117.

[6] A. J. Sathyamoorthy, U. Patel, T. Guan, and D. Manocha, “Frozone: Freezing-free, pedestrian-friendly navigation in human crowds,” IEEE Robotics and Automation Letters, vol. 5, no. 3, pp. 4352–4359, 2020.

[7] Y. F. Chen, M. Liu, M. Everett, and J. P. How, “Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning,” in ICRA. IEEE, 2017, pp. 285–292.

[8] A. Jagan Sathyamoorthy, J. Liang, U. Patel, T. Guan, R. Chandra, and D. Manocha, “DenseC2void: Real-time Navigation in Dense Crowds using Anticipatory Behaviors,” arXiv e-prints, p. arXiv:2002.03038, Feb. 2020.

[9] N. Ashton, M. E. Shaw, and A. P. Worsham, “Affective reactions to interpersonal distances by friends and strangers,” Bulletin of the psychonomic society, vol. 15, pp. 306–308, 1980.

[10] S. Karau and K. Williams, “The effects of group cohesiveness on social loafing and social compensation.” Group Dynamics: Theory, Research, and Practice, vol. 1, pp. 156–168, 1997.

[11] E. Knowles, “Boundaries around group interaction: the effect of group size and member status on boundary permeability,” Journal of personality and social psychology, vol. 26, no. 3, p. 327—331, June 1973. [Online]. Available: https://doi.org/10.1037/h0034464.

[12] E. T. Hall, R. L. Birdwhistell, B. Bock, P. Bohannan, A. R. Diebold, M. Durbin, M. S. Edmonson, J. L. Fischer, D. Hymes, S. T. Kimball, W. L. Barre, J. E. McClelland, D. S. Marshall, G. B. Milner, H. B. Sarles, G. L. Trager, and A. P. Vayda, “Proxemics [and comments and replies],” Current Anthropology, vol. 9, no. 2/3, pp. 83–108, 1968. [Online]. Available: https://doi.org/10.1086/200975.

[13] J. Wagnild and C. M. Wall-Scheffler, “Energetic consequences of human sociality: Walking speed choices among friendly dyads,” PLOS ONE, vol. 8, no. 10, pp. 1–6, 10 2013. [Online]. Available: https://doi.org/10.1371/journal.pone.0076576.

[14] J. Shao, C. C. Loy, and X. Wang, “Scene-independent group profiling in crowd,” in 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 2227–2234.

[15] J. Redmon and A. Farhadi, “Yolo3: An incremental improvement,” CoRR, vol. abs/1804.02767, 2018. [Online]. Available: http://arxiv.org/abs/1804.02767.

[16] R. Chandra, U. Bhattacharya, A. Bera, and D. Manocha, “Densespd: Pedestrian tracking in dense crowds using front-rvo and sparse features,” arXiv preprint arXiv:1906.10313, 2019.

[17] W. Choi, Y.-W. Chao, C. Pantofaru, and S. Savarese, “Discovering groups of people in images,” in Computer Vision – ECCV 2014, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 417–433.

[18] A. K. Chandrahan, L. A. Poh, and P. Vadakkepatt, “Identifying social groups in pedestrian crowd videos,” in 2015 Eighth International Conference on Advances in Pattern Recognition (ICAPR), 2015, pp. 1–6.

[19] R. M. Favaretto, L. L. Dihl, and S. R. Musse, “Detecting crowd features in video sequences,” in 2016 29th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), 2016, pp. 201–208.
