Semantically-Aware Pedestrian Intent Prediction With Barrier Functions and Mixed-Integer Quadratic Programming

Jaskaran Grover*, Yiwei Lyu*, Wenhao Luo†, Changliu Liu*
John Dolan*, Katia Sycara*
*The Robotics Institute, Carnegie Mellon University, USA
(email: {jaskarag, yiweilyu, cliu6, jdolan, sycara}@andrew.cmu.edu
† Department of Computer Science, UNC Charlotte.)

Abstract We develop algorithms for inferring long-term intentions and parameters of local collision-avoidance behavior of agents in a multiagent system from their trajectories. This problem is challenging because an agent's observed trajectory only partially manifests its long-term task; it also contains adjustments made by the agent to ensure collision avoidance with other agents and obstacles in the environment. Since an observer would have no means to determine the magnitude of these adjustments, it is difficult to isolate the task-oriented component from the observed motion. To circumvent this problem, we model the agent's dynamics using a reactive optimization whose objective function captures the long-term task while its constraints capture collision-avoidance behavior. We develop two robust mixed-integer programming algorithms that infer the task and safety related parameters of this optimization problem from the positions and velocities of the agents. These algorithms are validated on synthetic datasets using parameter estimation errors, displacement errors and computation time as metrics. We further test these algorithms on a dataset of real human trajectories. We show that the learned parameters capture the true underlying pedestrian dynamics by rolling out the learned model and showing similarity between the ground truth trajectories and the reconstructed trajectories.

Keywords: Inverse Optimization, Safe Control, Human Motion Prediction, Optimal Control.

1. INTRODUCTION

As robots are envisioned to co-exist with humans, frequent close-proximity interactions amongst robots and humans are inevitable. One critical challenge in programming the behavior of robots around humans is ensuring that the robots' motions are safe, *i.e.*, avoid collisions with humans. This is all the more pressing when the underlying dynamics models of humans are unknown. This is because the safety guarantees of most model-based control design methods assume that the dynamic models of other agents (including humans) are known a-priori. Thus to ensure that such guarantees are respected, it is necessary to develop methods for inferring the dynamics of these agents so that the control engineer can use those models to predict future behaviors of humans for generating safe robot motions.

In this paper, we consider the problem of dynamics inference for a group of heterogeneous agents such as pedestrians. The objective is to identify a set of behavior and safety-related parameters for each agent's dynamics model that describe their (i) long-term intention (such as goal, desired velocity etc.), and (ii) collision avoidance behavior around other agents or walls, (e.g. underlying safety margin, aggressiveness etc). From the perspective of an observer watching these agents, this inference problem is challenging because the motion that the observer watches comes through the filters of goal-directed behavior and safety combined. Because of this, the observer cannot tell the extent to which safety constraints manifest in the agent's motion. Therefore, the observer must learn the safety margins of each agent as well to perform this dynamics disaggregation.

To address this challenge, we take recourse to control barrier function-based quadratic program (CBF-QP) to model the dynamics of each agent Wang et al. (2017). CBF-QP models an agent's intent to follow a nominal task-oriented plan as closely as possible unless this plan causes the agent to come too close to another agent or a wall Ames et al. (2019); Cheng et al. (2020); Luo et al. (2020); Lyu et al. (2021). It also captures the aggressiveness of an agent *i.e.* the extent of its unwillingness to sacrifice optimality for attaining collision-free motions. Finally, it models cooperation among different agents to achieve collision-free behavior.

We develop two mixed-integer quadratic programming (MIQP) based algorithms to learn the parameters of the CBF-QP based model describing each agent's behavior from its observed trajectories. This allows for individually learning the parameters of the task-oriented component and the collision avoidance-related component of the agent. We use stationarity error and prediction error as heuristics to learn parameters that best explain the observed measurements. Our proposed algorithms are decentralized, robust to model mismatch and do not require long

training times (we show empirical results to support these claims). Additionally, by virtue of being model-based, the parameters we learn have an intuitive physics-based interpretation. Because of this feature, our inference approach enables reliable prediction that is generalizable to unseen scenarios, e.g. it describes how an agent will behave when entering a new scenario with a different set of nearby obstacles. Lastly, in addition to learning these parameters, our algorithm automatically learns which obstacles/agents in the environment influence the ego agent's dynamics from its velocity. Identification of these interest entities is what makes our algorithms integer-programs. We validate these algorithms on several simulated datasets and show small values of parameter reconstruction errors and small average and final displacement errors. Next, we evaluate them on a pedestrian dataset called THöR Rudenko et al. (2020a) which contains human motion trajectories recorded in a controlled indoor experiment at Örebro University. These trajectories exhibit social interactions that occur in populated spaces like offices, thus making them suitable for evaluating our algorithms. Our results show that the learned parameters capture pedestrian dynamics accurately, which we demonstrate by showing low values of average and final displacement errors.

2. RELATED WORK

Among all human motion prediction methods, physics based models are the most classic, which generate future human trajectories based on explicit dynamic models Rudenko et al. (2020b). Since this paper focuses on inferring the interactive mechanism among agents, we primarily review literature incorporating local agent interaction models into their inference pipeline Rudenko et al. (2020b). Social Force (SF) model Vasquez et al. (2008); Luber et al. (2010) is widely used to describe attractive forces from a goal and repulsive forces from other agents and obstacles. Grover et al. (2020b,a, 2021) learn task-related parameters of heterogeneous agents but assume known safety margins. In Elfring et al. (2014) a sparse topological map of the dynamic environment consisting of varying state-destination pairs is generated and used to infer the most likely goal. For human-agent interactions in a crowded environment, Oli et al. (2013) proposed a method to classify each human as aware or not aware to the agent based on visual cues, which is then used to describe the sources of the repulsive forces the agent feels from entities in the environment. However, the classification correctness on each human affects the prediction performance greatly. Therefore, a more principled approach is needed that automates the process of identifying which entities in the environment influence the ego agent's dynamics explicitly.

Compared to social forces approaches considering perpetual and long-range repulsion from other agents, our proposed modeling choice is able to automatically filter out influence from agents far away from the ego agent in the ego-agent's dynamics. Additionally, our model incorporates repulsion from only those agents/walls in the ego agent's dynamics that stand in the way of the ego agent en-route to its goal, *i.e.* it implicitly captures directional 'local gaze' of the agents. This computation is completely automated *i.e.* our model figures out these interest agents/walls among all agents/obstacles in the environment automatically.

Approaches based on reachability analysis have also been explored recently to address the trajectory forecasting problem. In Zechel et al. (2019), the reachable space of each pedestrian is determined based on a physical model and pruned by eliminating areas of static obstacles or other agents. However, these approaches are generally computationally expensive. Luo et al. (2019) treats the prediction task as constrained optimization in traffic agents' velocity space and leverages the prediction of human states as an optimization problem of prediction-related parameters to account for different human behaviors. To address joint motion prediction, the concept of linear trajectory avoidance is proposed based on expected point of closest approach between pedestrians Pellegrini et al. (2009). The expected point is used as the driving force to perform collision avoidance between agents. Built upon the idea of linear trajectory avoidance, Yamaguchi et al. (2011); Robicquet et al. (2016) further formulate the trajectory prediction task as an energy function minimization problem, which considers different motion properties, including speed and direction. However, these methods rely on hand-constructed cost functions, which contain scene-dependent variables that needed to be learned. In our method, we use basis functions to capture the nominal task-oriented motion of each agent agent, and we learn the coefficients of those basis functions from trajectory data. By virtue of this, there is a less restrictive requirement on parameter tuning.

The outline of this paper is as follows. In 3, we describe our model of the dynamics of each agent and pose the task + safety constraint inference problem. In 4, we propose our MIQP inference algorithms based on stationarity-residual and predictability-loss minimization. In 5, we validate these algorithms on synthetic data sets, compare them using parameter inference and trajectory reconstruction errors and evaluate their robustness to measurement noise. In 6, we show the results of our algorithms on the THöR dataset. Finally, we conclude in 7 with directions for future work.

3. MULTIAGENT SAFE TASK-BASED CONTROL AND INFERENCE PROBLEM

In this section, we propose our model for the dynamics of each agent in the system. We use CBF-QPs Wang et al. (2017) to capture the task-oriented objective of each agent as well as its safety constraints with other agents and obstacles in the environment. After settling on the dynamic model, we formalize the inference problem with respect to this model.

3.1 Agents' dynamic model

We model each agent as a single integrator that is velocity-controlled. Suppose there are a total of $N_A + 1$ agents in the system. Let the position of the agents be $\boldsymbol{x}_i \in \mathbb{R}^2$ and their velocities be $\boldsymbol{u}_i \in \mathbb{R}^2$ $\forall i \in \{1, \cdots, N_A + 1\}$. We assume that their dynamics are given by

$$\dot{\boldsymbol{x}}_i = \boldsymbol{u}_i \quad \forall i \in \{1, \cdots, N_A + 1\} \tag{1}$$

To keep the discussion simple, we focus on one agent located at $\boldsymbol{x}$. This agent has a primary task and we assume

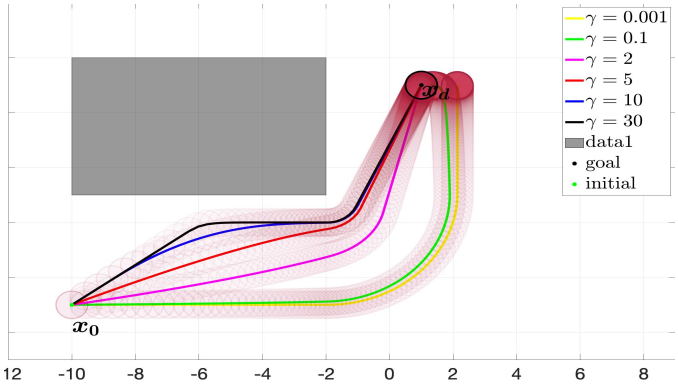


Figure 1. Sample trajectories produced by (13) with one rectangular obstacle as a constraint. Increasing γ makes the trajectory less conservative and makes it follow the reference controller more closely.

it can accomplish this task by using a reference control $\hat{\boldsymbol{u}}_{\boldsymbol{\theta}}$ in (1). We represent this control as

$$\hat{\boldsymbol{u}}_{\boldsymbol{\theta}}(\boldsymbol{x}) = C(\boldsymbol{x})\boldsymbol{\theta} + \boldsymbol{d}(\boldsymbol{x}) \tag{2}$$

Here $\boldsymbol{\theta}$ are the parameters capturing the task of that agent and are in general different for each agent. $C(\boldsymbol{x}), \boldsymbol{d}(\boldsymbol{x})$ are some task-oriented basis functions. This representation is general enough to capture the elementary tasks, for example:

(1) *Reaching a goal position:* Suppose that the primary task is to reach a goal position $\boldsymbol{x}_d$ at an exponentially fast speed governed by a gain k_p. A candidate control that can achieve this is $\hat{\boldsymbol{u}}(\boldsymbol{x}) = -k_p(\boldsymbol{x} - \boldsymbol{x}_d)$. This can be written in a form akin to (2):

$$\hat{\boldsymbol{u}}_{\boldsymbol{\theta}} = \underbrace{\begin{bmatrix} -\boldsymbol{x}_x & 1 & 0 \\ -\boldsymbol{x}_y & 0 & 1 \end{bmatrix}}_{C(\boldsymbol{x})} \underbrace{\begin{bmatrix} k_p \\ k_p \boldsymbol{x}_{d_x} \\ k_p \boldsymbol{x}_{d_y} \end{bmatrix}}_{\boldsymbol{\theta}} + \underbrace{\boldsymbol{0}}_{\boldsymbol{d}(\boldsymbol{x})} \tag{3}$$

(2) *Maintaining a constant velocity*: Suppose the task of the agent is to maintain a constant velocity $\boldsymbol{v}_d$, then we can select $\hat{\boldsymbol{u}}(\boldsymbol{x}) = \boldsymbol{v}_d$. This control expressed akin to (2) is

$$\hat{\boldsymbol{u}}_{\boldsymbol{\theta}} = \underbrace{I}_{C(\boldsymbol{x})} \underbrace{\boldsymbol{v}_{\boldsymbol{d}}}_{\boldsymbol{\theta}} + \underbrace{\boldsymbol{0}}_{\boldsymbol{d}(\boldsymbol{x})} \tag{4}$$

In addition to the task, this agent is also driven by the need to stay collision-free with the remaining agents and obstacles. Next, we describe how we capture these aspects of the agents' motion.

Modeling safety with other agents: We assume that all agents cooperate to achieve collision avoidance amongst one another while performing their respective tasks. Let the other agents be located at positions $\{\boldsymbol{x}_j^o\}\ \forall j \in \{1, 2, \cdots, N_A\}$. The ego agent and agent j are collision-free iff their positions $(\boldsymbol{x}, \boldsymbol{x}_j^o)$ satisfy $\|\Delta \boldsymbol{x}_j\|^2 \geq D_s^2$ where $\Delta \boldsymbol{x}_j \coloneqq \boldsymbol{x} - \boldsymbol{x}_j^o$ and D_s is a desired safety margin. In prior work Wang et al. (2017), control barrier functions were used to derive linear constraints on agents' velocities $\boldsymbol{u}$ for ensuring inter-agent collision-free behavior. We use these constraints as is:

$$A^A \boldsymbol{u} \leq \boldsymbol{b}^A_{\gamma, D_s}, \tag{5}$$

where superscript A denotes that these constraints model safety with other agents. $A^A \in \mathbb{R}^{N_A \times 2}$, $\boldsymbol{b}^A \in \mathbb{R}^{N_A}$ are defined so that the j^{th} row of A^A and j^{th} entry of $\boldsymbol{b}^A_{\gamma, D_s}$ are

$$\boldsymbol{a}_j^T \coloneqq -\Delta \boldsymbol{x}_j^T = -(\boldsymbol{x} - \boldsymbol{x}_j^o)^T \tag{6}$$

$$b_j \coloneqq \gamma(\|\boldsymbol{x} - \boldsymbol{x}_j^o\|^2 - {D_s}^2)\ \forall j \in \{1, 2, \ldots, N_A\} \tag{7}$$

Here $\gamma > 0$ is a hyperparameter unique to each agent. In 1, we illustrate that γ captures the unwillingness of an agent to compromise on task completion to ensure safety.

Modeling safety with walls/obstacles: Let there be N_O obstacles in the environment, which we assume are polytopic. We denote them as $\mathcal{P}_i \coloneqq \{\boldsymbol{y} \in \mathbb{R}^2 | A_i^P \boldsymbol{y} \leq b_i^P\}$ $\forall i \in \{1, \cdots, N_O\}$. To model the agent's safety with the $\mathcal{P}_i$, we assume that the agent (located at $\boldsymbol{x}$) tries to stay D_s distance away from the point on $\mathcal{P}_i$ closest to $\boldsymbol{x}$. This point can be calculated by the following QP

$$\begin{aligned} \boldsymbol{y}_i^O = \arg\min_{\boldsymbol{y}} \quad & \|\boldsymbol{x} - \boldsymbol{y}\|^2 \\ \text{subject to} \quad & \boldsymbol{y} \in \mathcal{P}_i \iff A_i^P \boldsymbol{y} \leq b_i^P \end{aligned} \tag{8}$$

Akin to (5), the following constraints on the ego agent's velocity $\boldsymbol{u}$ account for its safety with each obstacle $\mathcal{P}_i$:

$$A^O \boldsymbol{u} \leq \boldsymbol{b}^O_{\gamma, D_s}, \tag{9}$$

where superscript O denotes that these constraints model safety with obstacles. $A^O \in \mathbb{R}^{N_O \times 2}$, $\boldsymbol{b}^O \in \mathbb{R}^{N_O}$ are defined such that the i^{th} row of A^O and the i^{th} entry of $\boldsymbol{b}_{\gamma, D_s}$ are

$$\boldsymbol{a}_i^T \coloneqq -(\boldsymbol{x} - \boldsymbol{y}_i^O)^T \tag{10}$$

$$b_i \coloneqq \gamma(\|\boldsymbol{x} - \boldsymbol{y}_i^O\|^2 - {D_s}^2)\ \forall i \in \{1, 2, \ldots, N_O\} \tag{11}$$

Constraints (5) and (9) can be combined as

$$\underbrace{\begin{bmatrix} A^A \\ A^O \end{bmatrix}}_{A} \boldsymbol{u} \leq \underbrace{\begin{bmatrix} \boldsymbol{b}^A_{\gamma, D_s} \\ \boldsymbol{b}^O_{\gamma, D_s} \end{bmatrix}}_{\boldsymbol{b}} \tag{12}$$

To combine the collision avoidance requirements (12) with the task completion objective (2), we assume that the agent solves a QP that computes a controller closest to its reference $\hat{\boldsymbol{u}}_{\boldsymbol{\theta}}(\boldsymbol{x})$ and satisfies $N_A + N_O$ constraints (12) as follows:

$$\boxed{\begin{aligned} \dot{\boldsymbol{x}} = \boldsymbol{u}^* \quad = \arg\min_{\boldsymbol{u}} \ & \|\boldsymbol{u} - \hat{\boldsymbol{u}}_{\boldsymbol{\theta}}(\boldsymbol{x})\|^2 \\ \text{subject to} \quad & A\boldsymbol{u} \leq \boldsymbol{b}_{\gamma, D_s} \end{aligned}} \tag{13}$$

Going forward, we will assume that each agent in the multiagent system uses (13) as its underlying dynamic model. The task+constraint parameters $\{\boldsymbol{\theta}, \gamma, D_s\}$ are what distinguish one agent from another. Fig. 1 shows example trajectories computed by this controller for an agent with one collision avoidance constraint relative to a rectangular obstacle. The reference control is $\hat{\boldsymbol{u}}(\boldsymbol{x}) = -k_p(\boldsymbol{x} - \boldsymbol{x}_d)$. To get an intuitive understanding of the parameter γ, we conducted several simulations of (13) with increasing values of γ keeping $D_s, \boldsymbol{x}_0, \boldsymbol{x}_d, k_p$ fixed. It is evident that a large γ makes the trajectory follow the reference controller more closely while still maintaining D_s relative to the obstacle. There are several reasons in favor of using (13) as the candidate model:

(1) The nominal control $\hat{\boldsymbol{u}}_{\boldsymbol{\theta}}(\boldsymbol{x})$ represents the agent's preferred plan should there be no other agents. This represents the self-motivated dynamics of that agent. Thus, having an optimization model the dynamics

automatically encourages the agent's intent to follow this plan as much as possible while also ensuring that safety is respected if and when other agents show up.

(2) The general representation of the task-oriented control (2) allows us to capture both linear and nonlinear behaviors in agent motion. Additionally, we can incorporate previously proposed agent dynamics models into our framework using this representation in (13).

(3) This model implicitly captures the 'local gaze' of the ego agent. In our prior work Grover et al. (2019), we showed that for the goal reaching task using (13), agents that do not lie in the way of the ego agent en-route to its goal do not influence its dynamics *i.e.* they lie outside its 'local gaze'. Thus, this model then automatically filters out agents far away from the ego agent that don't interfere with the ego agent's task.

3.2 Task + Safety Behavior Inference Problem Formulation

We now state the inference problem for the multiagent system. In informal terms, an external observer wishes to understand the parameters of the task + safety constraints of each agent by tracking its positions and velocities. Let's state all assumptions on the observer's knowledge:

Assumption 1. *The observer knows the task functions* $C(\boldsymbol{x}), \boldsymbol{d}(\boldsymbol{x})$ *of* $\hat{\boldsymbol{u}}_{\boldsymbol{\theta}}(\boldsymbol{x})$.

Assumption 2. *The observer knows the form of safety constraints* $A^A, \boldsymbol{b}^A_{\gamma,D_s}, A^O, \boldsymbol{b}^O_{\gamma,D_s}$ *except for* γ, D_s.

Definition 1 (Multiagent Behavior Inference)**.** *The observer's problem is to infer parameters* $\{\boldsymbol{\theta}, \gamma, D_s\}$ *for each agent by monitoring its position* $\boldsymbol{x}(t)$ *and the positions of other agents* $\boldsymbol{x}^o_j(t)$ $\forall j \in \{1, 2, \cdots, N_A\}$ *over some time.*

The observer will use a batch of K signal-response pairs *i.e.* $\Big(\underbrace{(\boldsymbol{x}(k), \{\boldsymbol{x}^o_j(k)\}_{j=1}^{N_A})}_{signal}, \underbrace{\boldsymbol{u}^*(k)}_{response} \Big)$ $\forall k \in \{1, 2, \cdots, K\}$ to compute an estimate of $\boldsymbol{\theta}, \gamma, D_s$. The focus is on inferring these for one agent (*i.e.* the ego agent). The observer can parallelize the algorithm to infer these parameters for other agents simultaneously. In the next section, we propose two algorithms to solve this problem using MIQP.

4. MIQP-BASED ROBUST INFERENCE ALGORITHMS

The general approach for inferring $\boldsymbol{\theta}, \gamma, D_s$ is to pose an empirical risk minimization algorithm that uses a reasonable heuristic as a loss. We propose two algorithms: the algorithm in 4.1 considers the prediction error as a heuristic while the algorithm in 4.2 considers a variant of the KKT loss proposed in Keshavarz et al. (2011) as a heuristic. Both these algorithms rely on the KKT conditions of (13). Thus, we state these conditions first before presenting these algorithms.

Let $(\boldsymbol{u}^*, \boldsymbol{\lambda}^*)$ be the optimal primal-dual solution to (13). The KKT conditions are Boyd and Vandenberghe (2004):

(1) Stationarity: $\boldsymbol{u}^*_{\boldsymbol{\theta}} = \hat{\boldsymbol{u}}_{\boldsymbol{\theta}}(\boldsymbol{x}) - \frac{1}{2} A^T(\boldsymbol{x}) \boldsymbol{\lambda}^*$

(2) Primal Feasibility: $A(\boldsymbol{x})\boldsymbol{u}^* \leq \boldsymbol{b}_{\gamma,D_s}(\boldsymbol{x})$

(3) Dual Feasibility: $\boldsymbol{\lambda}^* \geq \boldsymbol{0}$

(4) Complementary Slackness: $\boldsymbol{\lambda}^* \odot \big(A(\boldsymbol{x})\boldsymbol{u}^* - \boldsymbol{b}_{\gamma,D_s}(\boldsymbol{x})\big) = \boldsymbol{0}$

Complementary slackness can be re-posed with an equivalent formulation by using the big-M approach Dong et al. (2018). This is done by augmenting the lower bounds $\boldsymbol{0} \leq \boldsymbol{b}_{\gamma,D_s}(\boldsymbol{x}) - A(\boldsymbol{x})\boldsymbol{u}^*$ and $\boldsymbol{0} \leq \boldsymbol{\lambda}^*$ with artificial upper bounds as follows:

$$\begin{aligned} \boldsymbol{0} \leq \boldsymbol{b}_{\gamma,D_s}(\boldsymbol{x}) - A(\boldsymbol{x})\boldsymbol{u}^* &\leq M\boldsymbol{z} \\ \boldsymbol{0} \leq \boldsymbol{\lambda}^* &\leq M(\boldsymbol{1} - \boldsymbol{z}) \end{aligned} \tag{14}$$

Here $\boldsymbol{z} \in \{0,1\}^{N_A+N_O}$ are Boolean variables and M is a large number chosen as a hyperparameter. The Boolean variables $\boldsymbol{z}$ are also unknown and will be learned as part of the inference problem in the next section. Given these conditions, we are ready to develop the first inference algorithm.

4.1 Predictability Loss MIQP

The observer assumes that each agent uses (13) as the underlying model. Akin to this model, the observer poses a copy problem in which he treats $\boldsymbol{\theta}, \gamma, D_s$ as tunable knobs. These can be tuned until the *predicted* velocities computed by solving the copy problem match with the measured velocities. This can be done by solving:

$$\begin{aligned} \hat{\boldsymbol{\theta}}, \hat{\gamma}, \hat{D}_s, \{\hat{\boldsymbol{u}}_k\}_{k=1}^K = \underset{\boldsymbol{\theta},\gamma,D_s,\{\boldsymbol{u}_k\}_{k=1}^K}{\arg\min} & \sum_{k=1}^K \|\boldsymbol{u}_k - \boldsymbol{u}_k^{meas}\|^2 \\ \text{such that} \quad \boldsymbol{u}_k \text{ solves (13)} \quad & \forall k \in \{1, \cdots, K\} \end{aligned} \tag{15}$$

The cost function in (15) is the empirical sum of the deviations of the predicted controls $\boldsymbol{u}_k$ from the measured controls $\boldsymbol{u}_k^{meas}$. This is known as the *predictability loss* Aswani et al. (2018). Naturally, it makes sense to minimize this loss only if the observer's predicted controls solve the forward problem (13) which is posed as a constraint in (15). Since (13) is in itself an optimization problem, (15) is a bi-level optimization problem, which is known to be computationally difficult to solve. We convert this to a single level problem by replacing the inner problem with its KKT conditions as follows:

$$\begin{aligned} &\hat{\boldsymbol{\theta}}, \hat{\gamma}, \hat{D}_s, \{\hat{\boldsymbol{u}}_k\}_{k=1}^K, \{\hat{\boldsymbol{\lambda}}_k\}_{k=1}^K, \{\hat{\boldsymbol{z}}_k\}_{k=1}^K, \{\hat{\boldsymbol{\delta}}_k\}_{k=1}^K = \\ &\underset{\substack{\boldsymbol{\theta},\gamma,D_s,\{\boldsymbol{u}_k\}_{k=1}^K, \\ \{\boldsymbol{\lambda}_k\}_{k=1}^K,\{\boldsymbol{z}_k\}_{k=1}^K,\{\boldsymbol{\delta}_k\}_{k=1}^K}}{\arg\min} \quad \sum_{k=1}^K \|\boldsymbol{u}_k - \boldsymbol{u}_k^{meas}\|^2 + \rho \sum_{k=1}^K \|\boldsymbol{\delta}_k\|^2 \\ &\text{subject to} \\ &\quad \begin{aligned} \boldsymbol{0} \leq \boldsymbol{b}_{\gamma,D_s}(\boldsymbol{x}_k) - A(\boldsymbol{x}_k)\boldsymbol{u}_k &\leq M\boldsymbol{z}_k \\ \boldsymbol{0} \leq \boldsymbol{\lambda}_k &\leq M(\boldsymbol{1} - \boldsymbol{z}_k) \\ \{\boldsymbol{z}_k\}_{k=1}^K \in \{0,1\}^{N_A+N_O} & \\ -\boldsymbol{\delta}_k \leq \boldsymbol{u}_k - \hat{\boldsymbol{u}}_{\boldsymbol{\theta}}(\boldsymbol{x}_k) + \frac{1}{2} A^T(\boldsymbol{x}_k)\boldsymbol{\lambda}_k &\leq \boldsymbol{\delta}_k \\ \boldsymbol{\theta}_L \leq \boldsymbol{\theta} \leq \boldsymbol{\theta}_U, \gamma_L \leq \gamma \leq \gamma_U, D_{sL} \leq D_s &\leq D_{sU} \end{aligned} \end{aligned} \tag{16}$$

The cost function in (16) is quadratic. The first term in the cost is the aggregated prediction error and the second penalizes the magnitude of the slack variables. These variables account for how much the stationarity condition is violated. In case the observed measurements of the agent's velocity are noisy, this algorithm tries to find the controls that satisfy the KKT conditions as best

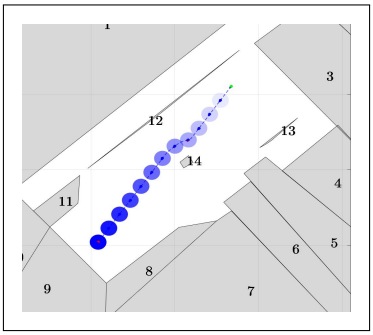


Figure 2. Environment used for generating the simulated trajectories using (13).

as they can while simultaneously ensuring that the inferred/predicted velocities are as close to the measured velocities as possible. Thus including slack variables confers robustness to this algorithm. The constraints are linear in $\boldsymbol{\theta}, \gamma, \gamma D_s^2, \{\boldsymbol{u}_k, \boldsymbol{\lambda}, \boldsymbol{z}_k, \boldsymbol{\delta}_k\}_{k=1}^K$. Since $\{\boldsymbol{z}_k\}_{k=1}^K$ are restricted to be Boolean, the overall problem is a mixed-integer QP. We use Gurobi to solve this problem.

4.2 Stationarity Loss MIQP

Another heuristic that can be used to solve the inference problem is the stationarity loss. This loss quantifies the residual of the stationarity condition evaluated on observed positions and velocities:

$$l_k^{Stat.} = \left\| \boldsymbol{u}_k^{meas} - \hat{\boldsymbol{u}}_{\boldsymbol{\theta}}(\boldsymbol{x}_k) + \frac{1}{2} A^T(\boldsymbol{x}_k) \boldsymbol{\lambda}_k \right\|^2 \tag{17}$$

This residual is quadratic in both $\boldsymbol{\theta}$ and $\boldsymbol{\lambda}_k$. Using K observed signal-response pairs, the observer poses an empirical risk minimization problem that queries for $\boldsymbol{\theta}, \gamma, D_s$, Lagrange multipliers $\{\boldsymbol{\lambda}_k\}_{k=1}^K$ and Boolean variables $\{\boldsymbol{z}_k\}_{k=1}^K$ which minimize the total stationarity loss evaluated on the observed measurements:

$$\begin{aligned} \hat{\boldsymbol{\theta}}, \hat{\gamma}, \hat{D}_s, \{\hat{\boldsymbol{\lambda}}_k\}_{k=1}^K, \{\hat{\boldsymbol{z}}_k\}_{k=1}^K \;=\; & \underset{\substack{\boldsymbol{\theta}, \gamma, D_s, \\ \{\boldsymbol{\lambda}_k\}_{k=1}^K, \{\boldsymbol{z}_k\}_{k=1}^K}}{\arg\min} \; \sum_{k=1}^K l_k^{Stat.} \\ \text{subject to} \quad & \\ \mathbf{0} \;\leq\; & \boldsymbol{b}_{\gamma, D_s}(\boldsymbol{x}_k) - A(\boldsymbol{x}_k)\boldsymbol{u}^* \leq M \boldsymbol{z}_k \\ \mathbf{0} \;\leq\; & \boldsymbol{\lambda}_k \leq M(\mathbf{1} - \boldsymbol{z}_k) \\ & \{\boldsymbol{z}_k\}_{k=1}^K \in \{0,1\}^{N_A + N_O} \\ & \boldsymbol{\theta}_L \leq \boldsymbol{\theta} \leq \boldsymbol{\theta}_U, \gamma_L \leq \gamma \leq \gamma_U, D_{sL} \leq D_s \leq D_{sU} \end{aligned} \tag{18}$$

The constraints in this problem capture all the KKT conditions in addition to bounds on $\boldsymbol{\theta}, \gamma, D_s$ from our prior knowledge. Since the cost is quadratic and all constraints are linear in $(\boldsymbol{\theta}, \gamma, \gamma D_s^2, \{\boldsymbol{u}_k\}_{k=1}^K, \{\boldsymbol{\lambda}_k\}_{k=1}^K, \{\boldsymbol{z}_k\}_{k=1}^K)$, the overall problem is a mixed-integer QP. We use Gurobi to solve this problem.

5. RESULTS: VALIDATION ON SYNTHETIC DATASETS

Before testing these algorithms on the human trajectory dataset, we evaluate them on a simulated dataset. We generated several trajectories of a single agent using (13) by varying its initial position $\boldsymbol{x}(0)$ in each run. The nominal task for this agent is to follow a constant velocity $\hat{\boldsymbol{u}}_{\boldsymbol{\theta}} = \boldsymbol{v}_d^*$. Additionally, the agent has to ensure safety margin of D_s^* with the obstacles with a conservativeness factor of γ^*. The observer's problem is to infer $\boldsymbol{v}_d, \gamma, D_s$ of this agent from the obtained trajectories using the predictability loss MIQP (15) and stationarity loss MIQP (18).

Testing Robustness: To test the robustness of these algorithms, we consider two scenarios: scenario (1) has no noise in the measured velocities and in scenario (2) we add zero mean Gaussian noise with 2 m/s standard deviation to the velocities ($\epsilon \sim \mathcal{N}(0, 2m/s)$). To assess repeatability, we conduct ten simulations with a randomly chosen initial position of the agent. The hyperparameter ρ for the predictability loss MIQP was chosen systematically by tuning performance on a validation dataset.

Warm Starting: Expecting that the computation time of predictability loss MIQP can be longer than stationarity loss MIQP due to more variables, we also compare the performance for predictability loss MIQP with and without a warm start using the result from stationary loss MIQP to see if there is any added benefit.

Error Metrics: The proposed algorithms are compared based on parameter reconstruction accuracy and computation time for parameter identification. In the human dataset, ground truth values of parameters are not available since the underlying model of humans is not (13) necessarily. Therefore, we consider the reconstructed trajectories using inferred parameters for performance evaluation. To this end, comparisons are made based on the average displacement error (ADE) and final displacement error (FDE) of the trajectories generated using the inferred parameters relative to the ground truth trajectories.

Environment: We consider the environment modeled after the map in the THöR dataset. For this, we manually convert the walls and obstacles into polytopes. There are a total of 14 obstacles in this enivornment *i.e.*$N_O = 14$. This is shown in 2.

Analysis of Results: Table 1 shows the parameter reconstruction errors and ADE, FDE errors averaged over ten runs with noiseless demonstrations for the THöR environment. It is evident that when perfect measurements are available to the observer, all the three methods share the same performance in terms of inferred parameters accuracy and ADEs and FDEs. However, we notice that stationary loss MIQP takes much longer for computation. Warm start does reduce the computation time for predictability loss MIQP, but only marginally.
Table 2 shows these errors with noisy demonstrations. From these tables, it is evident that the stationarity loss MIQP exhibits a great amount of robustness to measurement noise. The ADEs and FDEs are much smaller for the stationarity loss MIQP compared to the other algorithms. The computation time for this algorithm is still very large compared to the other two. Additionally, we observe that warm start improves the performance of predictability loss MIQP with lower error in inferred parameters accuracy and as well as ADE and FDE. There is not much reduction in computation time. Overall, stationarity loss MIQP has the best performance and strongest tolerance to noise in

Table 1.
Performance comparison on THoR environment with perfect velocity measurements

Metric / Algorithm	$\left\|\hat{\boldsymbol{v}}_d - \boldsymbol{v}_d^*\right\|$ [m/s]	$\lvert\hat{\gamma} - \gamma^*\rvert$ [1/s]	$\lvert\hat{D}_s - D_s^*\rvert$ [m]	ADE [m]	FDE [m]	Time [s]
Stat. Loss MIQP	0 ± 0	49.71 ± 0.29	0.0238 ± 0.028	0.0012 ± 0.0014	0.044 ± 0.0534	8.17 ± 0.49
Pred. Loss MIQP	0 ± 0	49.71 ± 0.29	0.0238 ± 0.028	0.0012 ± 0.0014	0.044 ± 0.0539	1.63 ± 0.067
Pred. Loss (warm start)	0 ± 0	49.71 ± 0.29	0.0238 ± 0.028	0.0012 ± 0.0014	0.044 ± 0.0539	1.61 ± 0.056

Table 2.
Performance comparison on THoR environment with noisy velocity measurements

Metric / Algorithm	$\left\|\hat{\boldsymbol{v}}_d - \boldsymbol{v}_d^*\right\|$ [m/s]	$\lvert\hat{\gamma} - \gamma^*\rvert$ [1/s]	$\lvert\hat{D}_s - D_s^*\rvert$ [m]	ADE [m]	FDE [m]	Time [s]
Stat. Loss MIQP	0.574 ± 1.558	49.64 ± 0.2	0.0295 ± 0.0164	0.0476 ± 0.1047	0.806 ± 1.432	8.008 ± 0.0827
Pred. Loss MIQP	2.90 ± 2.73	49.64 ± 0.2	0.0295 ± 0.0164	0.3734 ± 0.3718	12.90 ± 12.01	1.6 ± 0.082
Pred. Loss (warm start)	2.93 ± 2.766	49.64 ± 0.2	0.0295 ± 0.0164	0.3317 ± 0.3461	9.67 ± 10.281	1.592 ± 0.048

the velocity measurements. In the next section, we show that the stationarity loss MIQP performs best on the human dataset as well.

6. RESULTS: VALIDATION ON HUMAN DATASETS

In this section, we validate the proposed methods on a human dataset THöR Rudenko et al. (2020a). The trajectories in this dataset exhibit social interactions that occur in populated spaces like offices, malls etc., thus making them suitable for evaluating our algorithms. Our results show that the learned parameters capture pedestrian dynamics accurately, which we demonstrate by showing low ADE and FDE values.

We assume that the nominal task of the humans is to move at a constant velocity. Because of this assumption, we handpicked some scenes in the dataset where the humans approximately moved at a constant velocity. Thus, instead of using the entire datase for parameter identification, we selected 15 handpicked scenarios with selected time intervals of human trajectories where the constant velocity assumption was valid for most of the human pedestrians. Parameter inference is conducted on each single pedestrian independently.

The predictability loss MIQP (16) has a hyperparameter ρ which corresponds to the penalty on the slack variables. We chose this hyperparameter by doing a brute force search on all these scenarios. Since the humans do not necessarily follow (13) as the underlying model, the only way to check whether our inferred parameters rationalize the observed measurements is by rolling out (13) with the learned parameters and comparing the reconstructed trajectories with the ground truth trajectories. We use ADE and FDE as the metrics to do this comparison.

In 3, we plot the mean and standard deviation of the ADE and FDE errors averaged over all 15 scenarios for the predictability loss MIQP without warm start (3(a)), predictability loss MIQP with warm start (3(b)) and the stationarity loss MIQP (3(c)). The y axis denotes these errors and the x axis corresponds the value of the slack penalty hyperparameter ρ (on $\log_{10}$ scale).

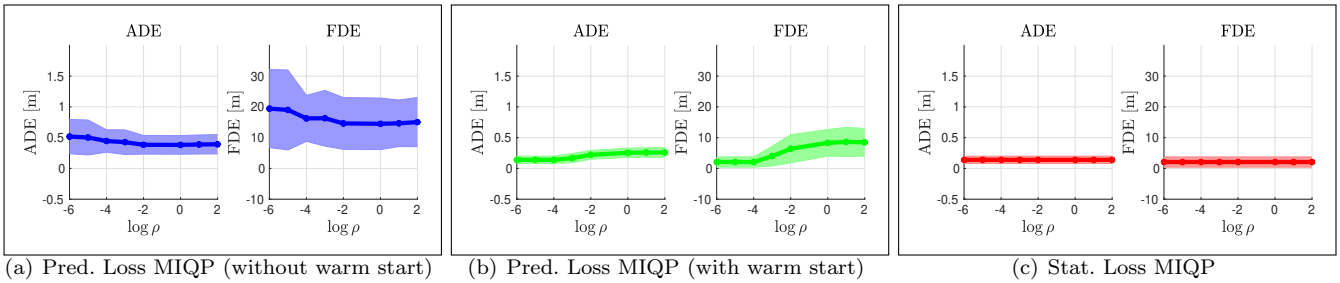
It is observed that predictability loss MIQP with warm start performs better then predictability loss MIQP without warm start, exhibiting both smaller mean and standard deviation in both ADE and FDE. Predictability loss MIQP with warm start performs exactly the same as the stationarity loss MIQP for $\rho < 10^{-4}$. However, the stationarity loss MIQP performs the best, exhibiting the smallest mean ADE and smallest mean FDE with small standard deviation as well.

Thus, we select the stationarity loss MIQP to highlight some results on the human dataset. Figure 4 shows results from six scenarios sampled from our dataset. The black trajectories are the ground truth trajectories of the humans. The green dots indicate their starting positions. Using these trajectories, we learn the $\hat{\boldsymbol{v}}_d, \hat{D}_s, \hat{\gamma}$ with (18) and then roll out (13) with the learned parameters. The obtained trajectories are plotted in red in 4. Since most of the red trajectories almost overlap with the given demonstrations, we conclude that the learned parameters do indeed rationalize the ground truth trajectories. In 4(e), there is one pedestrian whose motion changes abruptly exhibiting a turn and so technically this pedestrian does not satisfy the constant velocity assumption. Yet our algorithm learns the average behavior giving low FDE error for this pedestrian albeit high ADE. We can fix this problem by including some basis functions in $C(\boldsymbol{x}), \boldsymbol{d}(\boldsymbol{x})$ in (2) and relaxing the constant velocity model. Making the nominal task more complex with richer basis functions will allow us to get additional accuracy in trajectory reconstruction.

7. DISCUSSION AND CONCLUSIONS

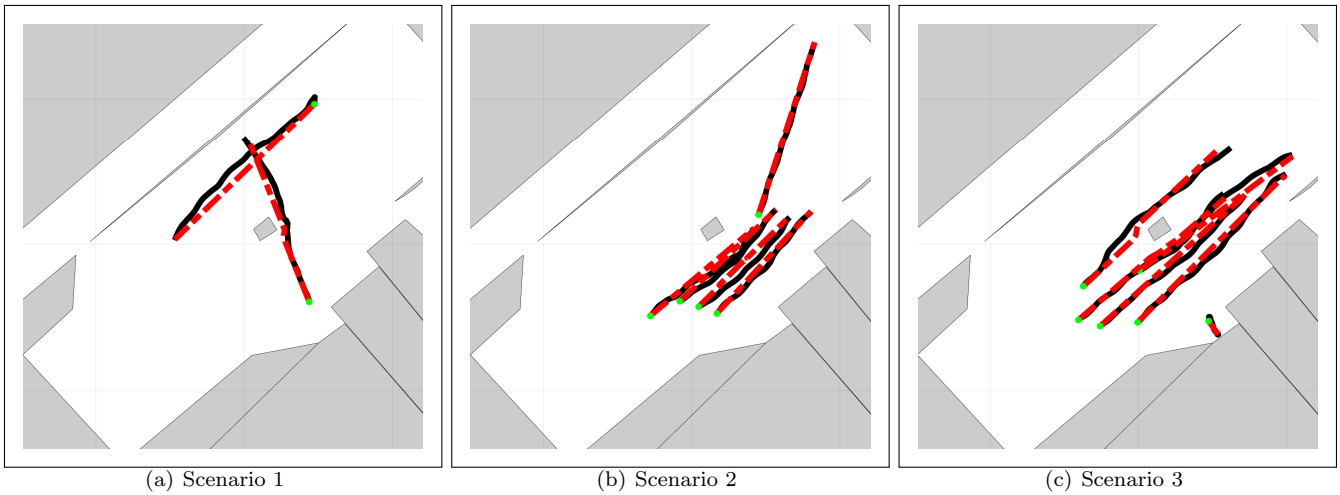
We considered the problem of simultaneously inferring task and safety constraint parameters of individual agents of a multiagent system. We modeled the agents using control barrier functions and developed the predictability loss MIQP and the stationarity loss MIQP to solve the inference problem. We demonstrated that the reconstructed parameters rationalize the observed measurements in a simulated single-agent scenario and also a real human dataset. There are several directions that we would like to take in future work.

- **Richer basis functions** $C(\boldsymbol{x}), \boldsymbol{d}(\boldsymbol{x})$: Sticking to the constant velocity model was a first-order choice and this itself gave us accurate reconstructions. However, we hope that including richer functions that capture

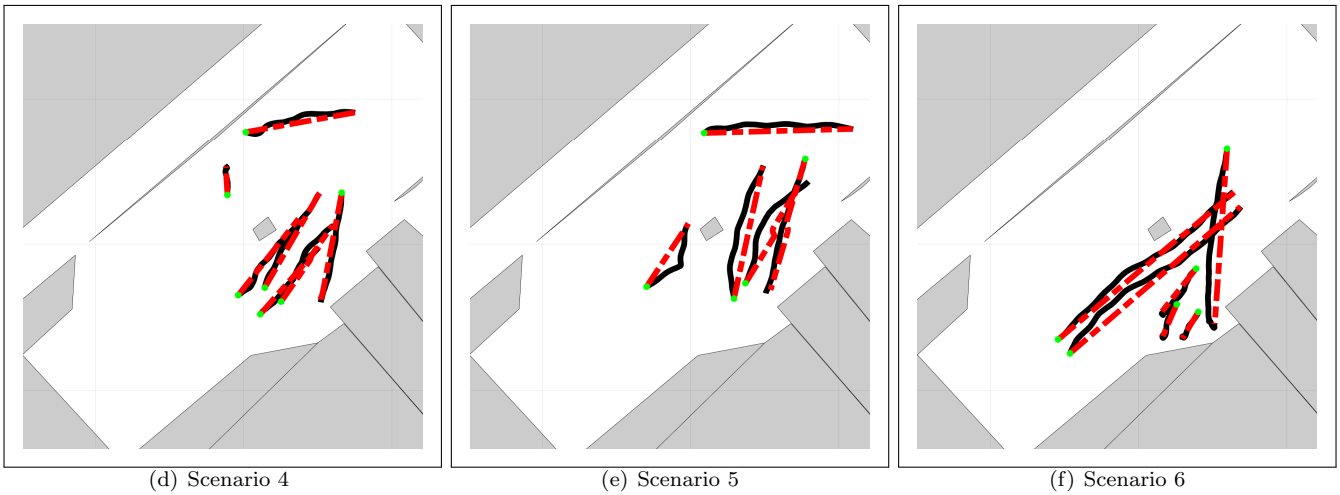


(a) Pred. Loss MIQP (without warm start) (b) Pred. Loss MIQP (with warm start) (c) Stat. Loss MIQP

Figure 3. ADE and FDE Errors averaged over 15 scenarios sampled from the THöR dataset with varying number of pedestrians in each scenario. X axis represents the log of the slack penalty hyperparameter ρ in (16). Among all three algorithms, the stationarity loss MIQP (3(c)) exhibits the lowest mean ADE and mean FDE as well as a low standard deviation.



(a) Scenario 1 (b) Scenario 2 (c) Scenario 3



(d) Scenario 4 (e) Scenario 5 (f) Scenario 6

Figure 4. **Validation on human dataset:** Comparison between the ground truth trajectories (solid black lines) and reconstructed trajectories with learned parameters (dotted red lines) on six selected scenarios. Starting points are marked in green.

the spatial frequencies of motion can allow us to get more accuracy. The overall problem complexity will only increase marginally since as we showed the problem is still an MIQP. This will allow us to capture changes in human trajectory better.

- **Time-varying parameters:** In our framework, we assumed that the task and constraint parameters $\boldsymbol{\theta}, \gamma, D_s$ are constant. However, humans may become more conservative as they come around agents that exhibit uncertain motions. As a result, these parameters may be time-varying. In our future work, we will adjust our code to allow for time-varying parameters. The overall problem will still remain an MIQP, although it will have to learn many parameters.
- **Multimodal Learning:** Currently, our proposed method learns one set of parameters for a human. However, humans may exhibit a distribution of behaviors. In future work, we will explore how to adjust the inference framework so that we can reverse engineer a distribution of these parameters instead of learning one set of parameter values.

8. ACKNOWLEDGEMENT

This material is in part based upon work supported by the National Science Foundation under Grant No. 2144489.

REFERENCES

Ames, A.D., Coogan, S., Egerstedt, M., Notomista, G., Sreenath, K., and Tabuada, P. (2019). Control barrier functions: Theory and applications. In *2019 18th European Control Conference (ECC)*, 3420–3431. IEEE.

Aswani, A., Shen, Z.J., and Siddiq, A. (2018). Inverse optimization with noisy data. *Operations Research*, 66(3), 870–892.

Boyd, S. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.

Cheng, R., Khojasteh, M.J., Ames, A.D., and Burdick, J.W. (2020). Safe multi-agent interaction through robust control barrier functions with learned uncertainties. In *59th IEEE Conference on Decision and Control (CDC)*, 777–783. IEEE.

Dong, C., Chen, Y., and Zeng, B. (2018). Generalized inverse optimization through online learning. *Advances in Neural Information Processing Systems*, 31.

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

Grover, J., Liu, C., and Sycara, K. (2019). Deadlock analysis and resolution in multi-robot systems (extended version).

Grover, J., Liu, C., and Sycara, K. (2020a). Feasible region-based identification using duality (extended version). *arXiv preprint arXiv:2011.04904*.

Grover, J., Liu, C., and Sycara, K. (2021). System identification for safe controllers using inverse optimization. *IFAC-PapersOnLine*, 54(20), 346–353. doi:https://doi.org/10.1016/j.ifacol.2021.11.198. URL https://www.sciencedirect.com/science/article/pii/S2405896321022424. Modeling, Estimation and Control Conference MECC 2021.

Grover, J.S., Liu, C., and Sycara, K. (2020b). Parameter identification for multirobot systems using optimization based controllers (extended version). *arXiv preprint arXiv:2009.13817*.

Keshavarz, A., Wang, Y., and Boyd, S. (2011). Imputing a convex objective function. In *2011 IEEE international symposium on intelligent control*, 613–619. IEEE.

Luber, M., Stork, J.A., Tipaldi, G.D., and Arras, K.O. (2010). People tracking with human motion predictions from social forces. In *2010 IEEE international conference on robotics and automation*, 464–469. IEEE.

Luo, W., Sun, W., and Kapoor, A. (2020). Multi-robot collision avoidance under uncertainty with probabilistic safety barrier certificates. *Advances in Neural Information Processing Systems*, 33, 372–383.

Luo, Y., Cai, P., Hsu, D., and Lee, W.S. (2019). Gamma: A general agent motion prediction model for autonomous driving. *arXiv preprint arXiv:1906.01566*.

Lyu, Y., Luo, W., and Dolan, J.M. (2021). Probabilistic safety-assured adaptive merging control for autonomous vehicles. In *IEEE International Conference on Robotics and Automation (ICRA)*, 10764–10770. IEEE.

Oli, S., L'Esperance, B., and Gupta, K. (2013). Human motion behaviour aware planner (hmbap) for path planning in dynamic human environments. In *2013 16th International Conference on Advanced Robotics (ICAR)*, 1–7. doi:10.1109/ICAR.2013.6766547.

Pellegrini, S., Ess, A., Schindler, K., and Van Gool, L. (2009). You'll never walk alone: Modeling social behavior for multi-target tracking. In *2009 IEEE 12th international conference on computer vision*, 261–268. IEEE.

Robicquet, A., Sadeghian, A., Alahi, A., and Savarese, S. (2016). Learning social etiquette: Human trajectory understanding in crowded scenes. In *European conference on computer vision*, 549–565. Springer.

Rudenko, A., Kucner, T.P., Swaminathan, C.S., Chadalavada, R.T., Arras, K.O., and Lilienthal, A.J. (2020a). Thör: Human-robot navigation data collection and accurate motion trajectories dataset. *IEEE Robotics and Automation Letters*, 5(2), 676–682.

Rudenko, A., Palmieri, L., Herman, M., Kitani, K.M., Gavrila, D.M., and Arras, K.O. (2020b). Human motion trajectory prediction: A survey. *The International Journal of Robotics Research*, 39(8), 895–935.

Vasquez, D., Fraichard, T., Aycard, O., and Laugier, C. (2008). Intentional motion on-line learning and prediction. *Mach. Vision Appl.*, 19(5–6), 411–425. doi:10.1007/s00138-007-0116-9. URL https://doi.org/10.1007/s00138-007-0116-9.

Wang, L., Ames, A.D., and Egerstedt, M. (2017). Safety barrier certificates for collisions-free multirobot systems. *IEEE Transactions on Robotics*, 33(3), 661–674.

Yamaguchi, K., Berg, A.C., Ortiz, L.E., and Berg, T.L. (2011). Who are you with and where are you going? In *CVPR 2011*, 1345–1352. doi:10.1109/CVPR.2011.5995468.

Zechel, P., Streiter, R., Bogenberger, K., and Göhner, U. (2019). Pedestrian occupancy prediction for autonomous vehicles. In *2019 Third IEEE International Conference on Robotic Computing (IRC)*, 230–235. IEEE.