

# SICNav: Safe and Interactive Crowd Navigation using Model Predictive Control and Bilevel Optimization

Sepehr Samavi, James R. Han, Florian Shkurti, Angela P. Schoellig

**Abstract**—Robots need to predict and react to human motions to navigate through a crowd without collisions. Many existing methods decouple prediction from planning, which does not account for the interaction between robot and human motions and can lead to the robot getting stuck. We propose SICNav, a Model Predictive Control (MPC) method that *jointly* solves for robot motion and predicted crowd motion in closed-loop. We model each human in the crowd to be following an Optimal Reciprocal Collision Avoidance (ORCA) scheme and embed that model as a constraint in the robot’s local planner, resulting in a bilevel nonlinear MPC optimization problem. We use a KKT-reformulation to cast the bilevel problem as a single level and use a nonlinear solver to optimize. Our MPC method can influence pedestrian motion while explicitly satisfying safety constraints in a single-robot multi-human environment. We analyze the performance of SICNav in two simulation environments and indoor experiments with a real robot to demonstrate safe robot motion that can influence the surrounding humans. We also validate the trajectory forecasting performance of ORCA on a human trajectory dataset. Code: [github.com/sepsamavi/safe-interactive-crowdnav.git](https://github.com/sepsamavi/safe-interactive-crowdnav.git).

**Keywords:** Social Navigation, Collision Avoidance, Autonomous Vehicle Navigation, Optimization and Optimal Control

## I. INTRODUCTION

For humans, navigating within crowds is considered a trivial task. However, uncertainty in the intent and future motions of pedestrians poses considerable challenges to safe (i.e. collision-free) robotic crowd navigation (e.g. in an environment similar to Fig. 1).

In a classical approach, the robot decouples motion prediction from planning. It first uses a prediction model (e.g. [1]–[3]) in open-loop to forecast the motion of humans, then uses these forecasts to generate robot actions that avoid future collisions (e.g. [4]–[7]). While these approaches are safe by design, e.g. by providing rigorous constraint-satisfaction guarantees [6], [8], they neglect the fact that the humans and the robot form a closed-loop system; the agents influence each other and need to *interactively* coordinate movement. Recent

Manuscript Submitted October XX, 2023; regular submission. (*Corresponding author: Sepehr Samavi.*)

Sepehr Samavi, James R. Han and Florian Shkurti are with the University of Toronto Robotics Institute and the Vector Institute for Artificial Intelligence, Toronto, Canada. Angela P. Schoellig is with the Technical University of Munich, University of Toronto, the Vector Institute for Artificial Intelligence, and the Munich Institute for Robotics and Machine Intelligence (MIRMI), Munich, Germany. (emails: [sepehr@robotics.utoronto.ca](mailto:sepehr@robotics.utoronto.ca), [jamesr.han@mail.utoronto.ca](mailto:jamesr.han@mail.utoronto.ca), [florian@cs.toronto.edu](mailto:florian@cs.toronto.edu), [angela.schoellig@tum.de](mailto:angela.schoellig@tum.de))

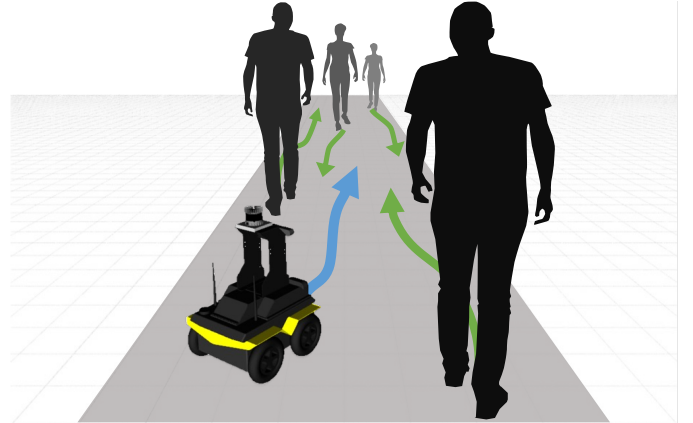


Fig. 1: Our MPC algorithm, SICNav, jointly optimizes robot trajectories (blue) and associated predicted human trajectories (green) for multiple humans while satisfying kino-dynamic and collision-avoidance safety constraints. Video of our work deployed on the pictured robot: [tiny.cc/sicnav\\_overview](https://tiny.cc/sicnav_overview).

work has even shown that, in a decoupled framework, using a prediction model with high accuracy to replace one with lower accuracy, does not lead to better robot navigation performance [9]. Furthermore, these approaches have been shown to exhibit the Freezing Robot Problem (FRP) [6], [10], where predictive uncertainty grows to the point that the robot cannot compute a safe path with sufficient certainty.

Recognizing the limitations of such decoupled strategies has led to the development of more sophisticated closed-loop prediction and planning methods to jointly generate predictions and robot motion. Some methods explicitly model cooperation (e.g. [11], [12]) or competition (e.g. [13]) among agents, while others use Reinforcement Learning (RL) techniques to learn policies that implicitly reason about interactions (e.g. [14]–[16]). Despite these advancements, existing interactive solutions are not designed to satisfy explicit safety constraints, and either couple the often opposing objectives of safety and performance (e.g. in collision-avoiding RL [16]), or need to rely on post-processing to ensure safety (e.g. via safety filters [17]).

Towards addressing this gap, we present Safe and Interactive Crowd Navigation (SICNav), a nonlinear MPC approach for jointly generating closed-loop predictions and planning robot actions (illustrated in green and blue in Fig. 1, respectively). SICNav explicitly models interactions between the robot and human agents, but unlike the aforementioned generative meth-

ods, allows for safety to be formulated as explicit constraints on the state space. We model each human in the environment as an agent following the ORCA algorithm [18]. Although originally intended for decentralized control of a swarm, ORCA is also commonly used to simulate human crowds for training RL agents [14], [19] and testing a variety of other methods [12], [20]. Furthermore, ORCA has even been used to efficiently generate open-loop predictions for humans in multi-agent environments [21], demonstrating comparable predictive accuracy as state-of-the-art methods (e.g. [22], [23]).

Our key insight is to directly incorporate ORCA into a robot planner. We formulate SICNav as a *bilevel* optimization problem. The *upper-level* contains the robot’s objective, safety constraints, and dynamics constraints. We add a set of human prediction constraints (one for each human), each of which is the solution to a *lower-level* ORCA optimization problem. This way, the predictions used for planning are closed-loop: our algorithm jointly generates robot actions and associated predicted human motion by assuming that the humans solve ORCA in reaction to robot motion [24].

The main goal of this paper is to present the theoretical foundations of our formulation and validate that they are sound, in preparation for the key next step of deploying our approach with real humans in the wild. Our contributions are fourfold. We first formulate the bilevel SICNav optimization problem. To solve, we derive a KKT-reformulation of the bilevel problem by replacing the lower-level ORCA problems with their Karush-Kuhn-Tucker (KKT) conditions, which we obtain through Lagrangian duality, and present a theoretical analysis into the equivalence of solving the reformulation versus the original problem. Second, to justify the use of ORCA as a model of pedestrians, we validate its trajectory forecasting performance on real human data in the commonly-used ETH/UCY benchmark [25], [26]. Third, we conduct an experimental analysis of SICNav in two simulation environments and observe interactive behavior between the robot and simulated humans, such as waiting for the other agent to pass and persuading another agent to move out of the way. Finally, we evaluate SICNav on a real robot in a lab environment, demonstrating improved navigation performance compared to a baseline method and making qualitative observations a robot using SICNav to navigate.

## II. RELATED WORK

*Open-loop prediction:* This paradigm *decouples* predictions about the future from motion planning. The robot’s planner takes as input the predicted future trajectories of pedestrians and reactively generates a plan to go around the predicted human motion (e.g. [6], [20], [27]–[29]). The consequence of this decoupling is that the impact of the robot’s motion on the predicted trajectory of the humans is assumed to be static with respect to the robot’s plan. Inaccurate predictions may lead to unsafe behavior, and in dense multi-human scenarios, high predictive uncertainty can result in the FRP [10], where the robot freezes because the planner cannot find a path that is sufficiently certain to be collision-free. Furthermore, following this paradigm neglects the fact that the robot may

be able to influence pedestrians to be cooperative and allow the robot to avoid freezing [24]. In contrast to these methods, we move away from the *predict-then-plan* paradigm by modelling the interactions between the robot and pedestrians. As our optimization algorithm is searching for an optimal solution for the robot’s trajectory, it concurrently optimizes predictions of the optimal human actions (with respect to ORCA) given the current iteration of the robot actions being optimized. Thus, our predictions adapts to the plan being evaluated.

*Cooperative interactions:* Another family of approaches assumes that all agents are cooperative in avoiding collisions. For example, the robot can model all agent trajectories (including its own) as Gaussian Processes (GPs) that represent a distribution of trajectories for each agent, then define an ‘interactive’ cost to minimize the likelihood that trajectory distributions will overlap (i.e. collide) in belief space [11], [12]. However, the assumption that all agents are exclusively cooperative may not always hold true. In many crowded scenarios, such as a densely packed urban sidewalk, pedestrians may compete for space and try to influence each other’s behavior to arrive at their destinations.

*Competitive interactions:* A number of recent methods aimed at self-driving capture a similar competition in highway merging by casting Autonomous vehicle (AV) navigation among human-operated cars as a dynamic zero-sum game, then use a variety of approximations to optimize the robot’s reward without needing to solve for a Nash equilibrium [13] [30]–[32]. The humans are modelled as *myopic* non-cooperative agents, i.e. the human and the robot have different reward functions, but the human is modelled to be following the lead of the robot, i.e. humans calculate the best response to the robot’s actions [13], [33]. This formulation, usually referred to as a Stackelberg game, has demonstrated interesting interactive behavior such as a robot moving in reverse to signal a human to move first at an intersection. However, these works usually focus on one-on-one driving scenarios, which may limit their applicability to navigating pedestrian crowds. First, crowds are often comprised of a variable number multiple agents all interacting with each other. Second, the driving environment is much more structured than pedestrian environments that usually lack explicit navigation rules. Similar to the two-agent game theoretic approaches, our approach models humans as *myopic* optimal agents following an ORCA strategy, but our planner uses bilevel optimization to output locally optimal plans in less structured crowd navigation settings while reasoning about multiple humans.

## III. PROBLEM FORMULATION AND BACKGROUND

The environment consists of a robot, simulated human agents, indexed  $j \in \mathbb{I}_{1:N} = \{1, \dots, N\}$ , and static obstacles in the form of line segments and indexed,  $\tilde{l} \in \{N+1, \dots, N+M\}$ . Fig. 2a illustrates an initial configuration consisting of one robot (in yellow) and  $N = 2$  simulated human agents (in gray). The goal of the robot is to move to a goal location (green star in Fig. 2a),  $g$ , starting from an initial configuration,  $x_0$ , without any collisions with static obstacles or simulated human agents.

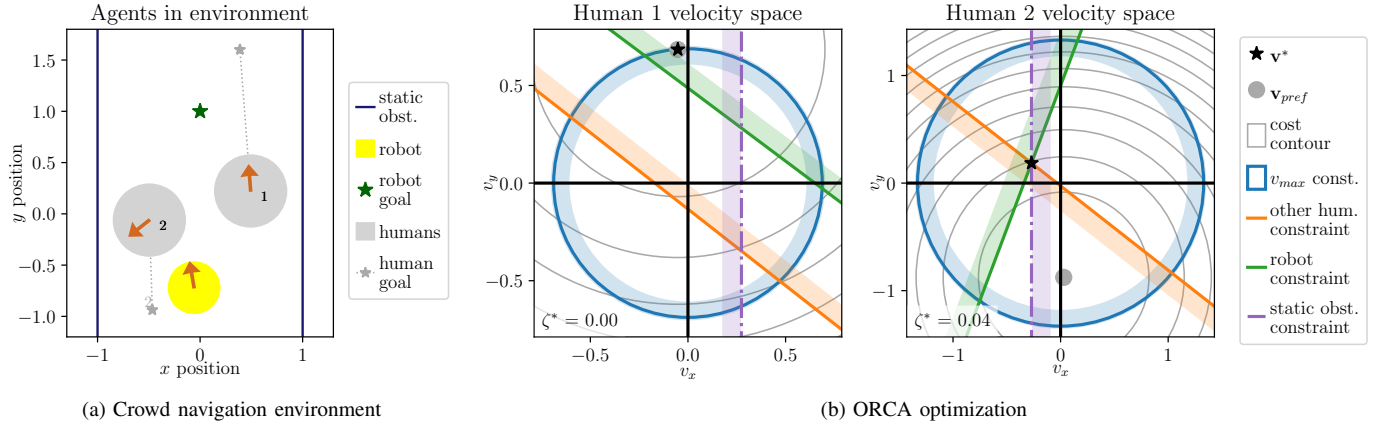


Fig. 2: (a) Exemplary crowd navigation environment state  $\mathbf{x}_t$  with one robot (yellow) and its goal position (green star),  $N = 2$  simulated humans (gray circles) with respective goals (gray stars), agent velocities (red arrows), and  $M = 2$  static obstacles (blue lines). (b) Graphical representation of the associated ORCA velocity space optimization (2) solved by each simulated human (Human 1 left and Human 2 right). The human's preferred velocity (gray circle) points toward their goal, and induces the minimum of the cost (gray contours). The constraints are illustrated as lines (for non-collision constraints) or a circle (for the maximum velocity constraint) with the feasible sides shaded. The optimum velocity,  $\mathbf{v}^*$ , i.e. the closest velocity to  $\mathbf{v}_{pref}$  that is feasible with respect to every constraint, is illustrated as a black star, with the associated optimal value of the slack variable,  $\zeta^*$ , written on the graph.

### A. Robot and Human System Dynamics

The system state is continuous and concatenates the states of all agents, one robot and  $N$  humans. At time step  $t$ , the state is,  $\mathbf{x}_t = (\mathbf{x}_t^{(r)}, \mathbf{x}_t^{(1)}, \dots, \mathbf{x}_t^{(N)}) \in \mathbb{R}^{6+6N}$ , where the state of the robot,  $\mathbf{x}_t^{(r)} \in \mathbb{R}^4$ , consists of its 2D position, heading, linear velocity at  $t-1$ , and the state of each human agent,  $\mathbf{x}_t^{(j)} \in \mathbb{R}^6$ , consists of its 2D position, and 2D velocity. We also assume to know the 2D goal position of each human. The goal of other agents is privileged information that is unavailable in the real-world, in Sec. VII-A we discuss how this assumption can be relaxed when deploying the algorithm.

We separate the dynamics of the system into separate functions for the robot and the humans. The robot dynamics,  $\mathbf{x}_{t+1}^{(r)} = \mathbf{f}(\mathbf{x}_t^{(r)}, \mathbf{u}_t)$ , are modeled as a kinematic unicycle model, where the control input for the system,  $\mathbf{u}_t \in \mathbb{R}^2$ , is a vector of the linear and angular velocity of the robot for time step  $t$ . The dynamics of each human  $\forall j \in \mathbb{I}_{1:N}$ ,  $\mathbf{x}_{t+1}^{(j)} = \mathbf{h}(\mathbf{x}_t^{(j)}, \tilde{\mathbf{u}}_t^{(j)})$ , are modeled as kinematic integrators, where  $\tilde{\mathbf{u}}_t^{(j)}$  denote the human actions. Note that these actions are not an input to the system, but rather predictions of the human actions given the state of the system,  $\mathbf{x}_t$ .

### B. ORCA Model of Human Motion

We obtain these predictions of human actions at each step,  $\tilde{\mathbf{u}}_t^{(j)}$ , by solving the ORCA optimization problem [18]. Before defining the complete optimization problem, we introduce a basic version. For one agent  $j$  at one time step  $t$ , we have the following set of optimal (i.e. minimum-cost feasible) velocities,

$$\begin{aligned} \mathcal{O}(\mathbf{x}_t) := & \arg \min_{\mathbf{v}} \quad \|\mathbf{v} - \mathbf{v}_{pref}(\mathbf{x}_t)\|_2^2 \\ \text{subject to} & \quad \|\mathbf{v}\|_2^2 \leq v_{max}^2 \\ & \quad \mathbf{n}^{(j)l}(\mathbf{x}_t)^\top \mathbf{v} \geq \rho^{(j)l}(\mathbf{x}_t) \end{aligned} \quad (1)$$

where the optimization variable  $\mathbf{v} \in \mathbb{R}^2$  is the velocity of agent  $j$  and the preferred velocity  $\mathbf{v}_{pref} : \mathbb{R}^{6+6N} \rightarrow \mathbb{R}^2$  is a vector toward the agent's goal position. The maximum velocity of the agent is constrained by  $v_{max}$ , and the vectors  $\mathbf{n}^{(j)l} : \mathbb{R}^{6+6N} \rightarrow \mathbb{R}^2$ , which are piece-wise linear functions of  $\mathbf{x}_t$ , and scalar  $\rho^{(j)l} : \mathbb{R}^{6+6N} \rightarrow \mathbb{R}^2$  which are nonlinear functions of  $\mathbf{x}_t$ , define linear collision avoidance constraints between agents  $j$  and agent  $l$ ,  $\forall l \in \{r, 1, \dots, N\} \setminus \{j\}$ . The cost and constraint functions in (1) depend on the positions and velocities of all agents in the system at each time step, therefore, the state of the system,  $\mathbf{x}_t$ , parameterizes this optimization problem. In the remainder of this section, we will reference Fig. 2, which illustrates an exemplary environment state (Fig. 2a) and a graph of the associated ORCA problem solved by each human (Fig. 2b).

*Agent-agent collision avoidance:* The normal vectors,  $\mathbf{n}^{(j)l}(\mathbf{x}_t)$ , and constants,  $\rho^{(j)l}(\mathbf{x}_t)$ , are formulated using a Reciprocal Velocity Obstacle (RVO) approach [18]. Assume agent  $j$  is solving (1). For each other agent,  $l \neq j$  (e.g. the robot and human 2 in Fig. 2), agent  $j$  uses the current relative position and velocity between itself and  $l$  to compute a region of relative velocity space, called a *velocity obstacle* [34], which contains all relative velocities that would cause a collision between the agents within some designated time horizon  $\tau$ . Assuming that both agents follow the same velocity-obstacle protocol, the algorithm finds uses this region to find a half-plane in the velocity space of  $j$  that forbid the agent from choosing a velocity for the next time step that would cause a collision with the other agent within  $\tau$ . In the case of human  $j = 1$  in Fig 2b, the green half plane with the feasible side shaded is induced by the position and velocity of the robot =  $l$ . As illustrated in Fig. 2a, the robot is below and to the left of human 1 and moving upwards ( $y \uparrow$ ), causing the velocity space of human 1 to be forbidden from velocities that move it to the left ( $x \downarrow$ ) or downwards ( $y \downarrow$ ) as that may cause a collision with the robot within  $\tau$ . Likewise, the position and velocity

of human  $l = 2$  induces the orange constraint for human 1.

*Static obstacle collision avoidance:* The algorithm uses a simplified RVO derivation to obtain half-plane constraints that avoid static obstacles defined as line segments, e.g. the static obstacles on the right and left in Fig. 2a. These constraints are similarly represented with a normal vector,  $\mathbf{n}^{(j|\bar{l})}(\mathbf{x}_t)$ , and a scalar,  $\rho^{(j|\bar{l})}(\mathbf{x}_t)$ , for all static obstacle indices  $\forall \bar{l} \in \{N + 1, \dots, N + M\}$ . In the velocity space of human 1 in Fig. 2b, the static obstacle on the right in Fig. 2a induces the dashed purple constraint to forbid human 1 from selecting a velocity towards the right ( $x \uparrow$ ) that would cause a collision with the obstacle.

*Feasible velocities:* For human 1 in Fig. 2, the feasible set of velocities are formed by the aforementioned non-collision constraints for the static obstacle and the robot, together with human 1's maximum permissible velocity (blue circle). In the graph for human 1 in Fig. 2b, we can see that the feasible set of velocities as the intersection of the shaded sides of these three constraints (green, blue and purple). In the case of human 1, the preferred velocity lies in this feasible set and is the optimum.

*Empty feasible set and ORCA relaxation:* As discussed in [18], in dense scenarios i.e. where agents are in close vicinity to each other, the problem (1) may become infeasible, i.e. the feasible set does not contain any points. In this case, the ORCA algorithm [18] switches to a secondary optimization problem that finds the safest-possible feasible velocity by minimizing the maximum distance of the velocity variable  $\mathbf{v}$  to the half-plane  $\mathbf{n}^{(j|l)}(\mathbf{x}_t)^\top \mathbf{v} \geq \rho^{(j|l)}(\mathbf{x}_t)$  (See Sec. 5.2 of [18]). The secondary problem is a relaxation equivalent to moving every pairwise ORCA half-plane outward (i.e. opposite direction to the feasible side) equally, until at least one velocity becomes feasible, then choosing the lowest magnitude velocity. We formulate a similar relaxation by adding a slack variable to the linear collision avoidance constraints in (1). We only add slack variables for the agent-agent collision constraints as only those constraints may result in infeasibility. Putting the relaxation and static obstacle constraints together, we obtain the following optimization problem, which we will use as our prediction model for humans in SICNav.

*Definition 1 (Relaxed ORCA):* Let  $\zeta$  be a slack variable, and let  $M \in \mathbb{R}_{++}$  be some sufficiently large constant. The set of velocities solving the relaxed ORCA problem for agent  $j$  is,

$$\begin{aligned} \mathcal{O}_{rlx}(\mathbf{x}_t) := & \arg \min_{\mathbf{v}, \zeta} \quad \|\mathbf{v} - \mathbf{v}_{\text{pref}}(\mathbf{x}_t)\|_2^2 + M\zeta^2 \\ & \text{subject to} \quad \|\mathbf{v}\|_2^2 \leq v_{\text{max}}^2 \\ & \quad \mathbf{n}^{(j|l)}(\mathbf{x}_t)^\top \mathbf{v} \geq \rho^{(j|l)}(\mathbf{x}_t) - \zeta \\ & \quad \mathbf{n}^{(j|\bar{l})}(\mathbf{x}_t)^\top \mathbf{v} \geq \rho^{(j|\bar{l})}(\mathbf{x}_t) \\ & \quad \zeta \geq 0 \end{aligned} \quad (2)$$

The relaxed problem is a convex three-dimensional Quadratically Constrained Quadratic Program (QCQP). We use an inexact penalty function to relax the agent-agent collision constraints on the cost function. The upper bound of the error tends to 0 as  $M \rightarrow \infty$ . Unlike the original paper [18] which picks the smallest velocity after relaxing the constraints, our approach picks the one closest to the preferred velocity.

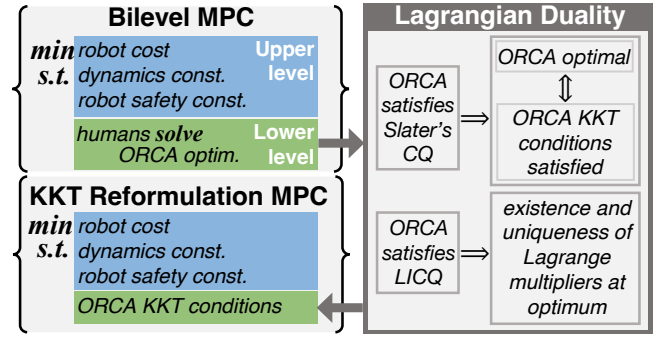


Fig. 3: Overview of our method. In section IV, we formulate a Bilevel MPC problem (3) that solves for robot actions in the upper level (blue) and predicted human actions via a lower level ORCA problems (green). In section IV-A, we use Lagrangian duality analysis to derive the KKT optimality criteria for the lower level problems, and in section IV-B, we reformulate the bilevel problem by replacing the lower level problems with their KKT optimality criteria to get a single level KKT Reformulation MPC problem that we can solve. In section V, we analyze the equivalence between optima of the reformulation and optima of the original bilevel problem.

The slack variable can be visualized as a third dimension protruding out of the page in Fig. 2b. In the case of human 2 there is **no** feasible velocity for  $\zeta = 0$  (not visualized), thus the agent-agent collision constraints are relaxed to  $\zeta^* = 0.04$  (visualized), and we observe the single feasible point (black star) that becomes feasible is selected as the optimum. Since the value of  $M$  is large, increasing the value of  $\zeta$  is severely penalized, meaning that the constraints are relaxed minimally (i.e. least unsafe) in order to obtain at least one feasible velocity. To discern the consequences of the relaxation, recall that ORCA without the relaxation (1) is guaranteed to be collision-free if the agents follow their optimal velocities for some designated time horizon,  $\tau$ . An optimal velocity with  $\zeta > 0$  means that a collision may occur within time  $\tau$ . However, this time horizon is set to be much larger than one typical time step, e.g.  $\tau = 2.0s$  in Fig. 2, therefore it is very unlikely that a collision ever occurs in practice.

#### IV. SICNAV MPC PROBLEM

Fig. 3 provides a graphical overview of our approach described in this section and the next. We formulate an MPC point stabilization controller as a bilevel optimization with the ORCA problem (2) as the lower level.

*Robot cost:* We define the components of the stage cost to penalize deviation from the robot's goal state,  $\bar{\mathbf{x}}_t := \mathbf{x}_t - \mathbf{x}_g$ , and excessive control effort,

$$l(\mathbf{x}_t, \mathbf{u}_t) = \bar{\mathbf{x}}_t^\top \mathbf{Q} \bar{\mathbf{x}}_t + \mathbf{u}_t^\top \mathbf{R} \mathbf{u}_t,$$

with positive semidefinite matrices  $\mathbf{Q}$  and  $\mathbf{R}$ . We also define a terminal penalty as,

$$V_f(\mathbf{x}_T) = \beta \bar{\mathbf{x}}_T^\top \mathbf{Q} \bar{\mathbf{x}}_T,$$

where  $\beta \geq 1$  is selected to be sufficiently large to ensure stability of the controller (See Theorem 2.41 in [35]). In practice, we select the goal state,  $\mathbf{x}_g$  to include the robot's final goal pose and target linear velocity, and we select  $\mathbf{Q}$  to be

diagonal and penalize only the robot's state, not the predicted state of the humans. We select  $\mathbf{R}$  to be diagonal and penalize the control effort of the robot.

*Robot safety constraints:* To avoid collisions between the robot and any human agents we add constraints of the form,

$$\mathbf{x}_t^\top \mathbf{P}_j \mathbf{x}_t \geq d_j^2, \forall j \in \mathbb{I}_{1:N},$$

where,  $\mathbf{P}_j \in \mathbb{R}^{n \times n}$  is a matrix that extracts the positions of the robot and each human agent  $j$  from the state and  $d_j$  is the minimum permissible distance between the two agents. For each static obstacle, we implement a piece-wise function that calculates the closest point on the line segment to the position of the robot and add a quadratic constraint similar to the robot-agent collision constraints. We also bound the input of the system in order to meet the kino-dynamic limits of a real-world robot,

$$\begin{aligned} \mathbf{u}_{\min}^{(r)} &\leq \mathbf{u}_t \leq \mathbf{u}_{\max}^{(r)}, \\ \Delta \mathbf{u}_{\min}^{(r)} &\leq \mathbf{u}_t - \mathbf{u}_{t-1} \leq \Delta \mathbf{u}_{\max}^{(r)}, \end{aligned}$$

With these definitions, the local planning MPC problem for the robot can be formulated as an optimization problem,

$$\begin{aligned} &\underset{\substack{\mathbf{x}_{0:T}, \mathbf{u}_{0:T-1}, \\ \tilde{\mathbf{u}}_{0:T-1}^{(0:N)}}}{\text{minimize}} && \sum_{t=0}^{T-1} l(\mathbf{x}_t, \mathbf{u}_t) + V_f(\mathbf{x}_T) \end{aligned} \quad (3a)$$

$$\text{subject to} \quad \mathbf{x}_0 = \mathbf{x}_0 \quad (3b)$$

$$\mathbf{x}_{t+1}^{(r)} = \mathbf{f}(\mathbf{x}_t^{(r)}, \mathbf{u}_t) \quad (3c)$$

$$\mathbf{u}_{\min}^{(r)} \leq \mathbf{u}_t \leq \mathbf{u}_{\max}^{(r)} \quad (3d)$$

$$\Delta \mathbf{u}_{\min}^{(r)} \leq \mathbf{u}_t - \mathbf{u}_{t-1} \leq \Delta \mathbf{u}_{\max}^{(r)} \quad (3e)$$

$$\mathbf{x}_t^\top \mathbf{P}_l \mathbf{x}_t \geq d_l^2 \quad (3f)$$

$$\tilde{\mathbf{u}}_t^{(j)} \in \mathcal{O}_{rlx}^{(j)}(\mathbf{x}_t) \quad (3g)$$

$$\mathbf{x}_{t+1}^{(j)} = \mathbf{h}(\mathbf{x}_t^{(j)}, \tilde{\mathbf{u}}_t^{(j)}) \quad (3h)$$

where all the constraints (3c)-(3h) are defined for each time step,  $\forall t \in \mathbb{I}_{0:T-1}$ , and the constraints (3g)-(3h) are defined for each human,  $\forall j \in \mathbb{I}_{1:N}$ . The robot collision constraint (3f) is defined for each human agent and each static obstacle in the environment,  $\forall l \in \{1, \dots, N, \dots, N+M\}$ . To solve this bilevel problem, we will reformulate to a single level by replacing the lower level with its KKT optimality conditions. As such, we begin by characterizing the optimality conditions for the ORCA problem (2).

#### A. Lagrangian Duality Analysis of ORCA Problem

To analyze the optimal solutions of the ORCA problem (2), we use Lagrange duality theory. First, we define the Lagrangian of the ORCA problem (2). Recall that each agent  $j$  has its own instance of the problem, however in the remainder of this section we drop the index  $j$  for legibility.

*Definition 2 (Lagrangian Dual):* For the ORCA problem associated to agent  $j$  (2), let the optimization variable be  $\boldsymbol{\nu} = (\mathbf{v}, \zeta) \in \mathbb{R}^3$ , let the cost function be,  $\varphi(\boldsymbol{\nu}; \mathbf{x}_t)$ , let the constraints be summarized by  $\mathbf{c}(\boldsymbol{\nu}; \mathbf{x}_t) \leq \mathbf{0}$ , and let  $\boldsymbol{\lambda} \in \mathbb{R}^{N+M+2}$ . The *Lagrangian* of problem (2) is,

$$\mathcal{L}(\boldsymbol{\nu}, \boldsymbol{\lambda}; \mathbf{x}_t) = \varphi(\boldsymbol{\nu}; \mathbf{x}_t) + \boldsymbol{\lambda}^\top \mathbf{c}(\boldsymbol{\nu}; \mathbf{x}_t) \quad (4)$$

where  $\boldsymbol{\lambda} \geq \mathbf{0}$  are the *Lagrange multipliers*. The *dual optimization* of the problem (2) is,

$$\max_{\boldsymbol{\lambda}} \inf_{\boldsymbol{\nu}} \mathcal{L}(\boldsymbol{\nu}, \boldsymbol{\lambda}; \mathbf{x}_t). \quad (5)$$

For there to exist some  $\boldsymbol{\lambda}^*$  such that the dual optimization is maximized, the problem (2) needs to satisfy the following constraint qualification [36].

*Definition 3 (SCQ): Slater's Constraint Qualification (SCQ)* is satisfied for problem (2) if there exists a point  $\boldsymbol{\nu}$  in the interior of the feasible set such that  $\mathbf{c}(\boldsymbol{\nu}; \mathbf{x}_t) < \mathbf{0}$ .

Problem (2) satisfies SCQ by construction. Starting at any point,  $\boldsymbol{\nu}$ , the value of the slack variable dimension,  $\zeta$ , can be increased such that the pairwise collision constraints are relaxed sufficiently for the feasible set to have an interior.

*Definition 4 (KKT Conditions):* The *Karush-Kuhn-Tucker (KKT)* conditions of (2) are defined as [36],

$$\text{Stationarity:} \quad \nabla_{\boldsymbol{\nu}} \mathcal{L}(\boldsymbol{\nu}, \boldsymbol{\lambda}; \mathbf{x}_t) = \mathbf{0} \quad (6a)$$

$$\text{Complementary Slackness:} \quad \boldsymbol{\lambda}^\top \mathbf{c}(\boldsymbol{\nu}; \mathbf{x}_t) = 0 \quad (6b)$$

$$\text{Primal Feasibility:} \quad \mathbf{c}(\boldsymbol{\nu}; \mathbf{x}_t) \leq \mathbf{0} \quad (6c)$$

$$\text{Dual Feasibility:} \quad \boldsymbol{\lambda} \geq \mathbf{0} \quad (6d)$$

Since the problem is convex and satisfies SCQ, the KKT conditions are necessary and sufficient for global optimality of the problem (2) [36]. Furthermore, note that the cost function in problem (2) is strongly convex with respect to  $\boldsymbol{\nu}$  due to the quadratic form of its cost function. Thus, the set  $\mathcal{O}_{rlx}(\mathbf{x}_t)$  is always singleton, i.e. the solution to the optimization problem is always unique [36]. In the next section, we will make use of the KKT conditions and the uniqueness of solutions reformulate a bilevel optimization, with problem (2) as the lower-level, to a single level problem.

#### B. KKT Reformulation of bilevel problem

In order to solve (3), we replace each instance of the lower level problem with its KKT conditions (6), to obtain the following Mathematical Program with Complementarity Constraints (MPCC),

$$\begin{aligned} &\underset{\substack{\mathbf{x}_{0:T}, \mathbf{u}_{0:T-1}, \tilde{\mathbf{u}}_{0:T-1}^{(0:N)}, \\ \zeta_{0:T-1}, \boldsymbol{\lambda}_{0:T-1}^{(0:N)}}}{\text{minimize}} && \sum_{t=0}^{T-1} l(\mathbf{x}_t, \mathbf{u}_t) + V_f(\mathbf{x}_T) \end{aligned} \quad (7a)$$

$$\text{subject to} \quad \mathbf{x}_0 = \mathbf{x}_0 \quad (7b)$$

$$\mathbf{x}_{t+1}^{(r)} = \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t) \quad (7c)$$

$$\mathbf{u}_{\min}^{(r)} \leq \mathbf{u}_t \leq \mathbf{u}_{\max}^{(r)} \quad (7d)$$

$$\Delta \mathbf{u}_{\min}^{(r)} \leq \mathbf{u}_t - \mathbf{u}_{t-1} \leq \Delta \mathbf{u}_{\max}^{(r)} \quad (7e)$$

$$\mathbf{x}_t^\top \mathbf{P}_l \mathbf{x}_t \geq d_l^2 \quad (7f)$$

$$\nabla_{\tilde{\mathbf{u}}_t^{(j)}, \zeta^{(j)}} \mathcal{L}(\tilde{\mathbf{u}}_t^{(j)}, \zeta^{(j)}, \boldsymbol{\lambda}^{(j)}; \mathbf{x}_t) = \mathbf{0} \quad (7g)$$

$$\boldsymbol{\lambda}_t^{(j)\top} \mathbf{c}^{(j)}(\tilde{\mathbf{u}}_t^{(j)}, \zeta^{(j)}; \mathbf{x}_t) = 0 \quad (7h)$$

$$\mathbf{c}^{(j)}(\tilde{\mathbf{u}}_t^{(j)}, \zeta^{(j)}; \mathbf{x}_t) \leq \mathbf{0} \quad (7i)$$

$$\boldsymbol{\lambda}_t^{(j)} \geq \mathbf{0} \quad (7j)$$

$$\mathbf{x}_{t+1}^{(j)} = \mathbf{h}(\mathbf{x}_t, \tilde{\mathbf{u}}_t^{(j)}) \quad (7k)$$

where the constraints in (7) defined for each human,  $\forall j, \forall l, \forall t$  analogously to (3). Each instance of constraint (3g) has been replaced by its KKT conditions to get (7g)-(7j). Note that the Lagrange multipliers of each instance of (3g) have become optimization variables in (7).

## V. ANALYSIS OF KKT-REFORMULATION

We analyze the equivalence of solutions of the KKT reformulation (7) and solutions of the bilevel problem (3). The authors in [37] present sufficient conditions for equivalence. For globally optimal solution of the reformulated problem (7) to be equivalent to globally optimal solutions of the original (3), it is required that the lower level (2) satisfy SCQ (Definition 3) for any parameterization  $\mathbf{x}_t$  (see Theorem 2.3 in [37]). As discussed in IV-A, the relaxed ORCA optimization (2) satisfies SCQ by construction. Thus, the global optimum of problem (7) is equivalent to the global optimum of (3).

However, problems (7) and the bilevel problem (3) are non-convex, therefore we also need to analyze the equivalence of locally optimal solutions. For locally optimal solutions, equivalence cannot be established without imposing further conditions on the lower level problem. In order for the Lagrange multipliers that optimize the dual problem to be unique, the problem (2) needs to satisfy the following additional constraint qualification [38].

*Definition 5 (LICQ):* For agent  $j$  at some environment state,  $\bar{\mathbf{x}}_t$ , which parameterizes the relaxed ORCA problem (2), let  $\bar{\boldsymbol{\nu}}$  be a locally optimal solution for the problem, and let  $g_i^{(j)}(\boldsymbol{\nu}; \mathbf{x}_t)$  denote the  $i^{\text{th}}$  row of the constraint function,  $\mathbf{c}^{(j)}(\cdot; \cdot)$ . Problem (2) is said to satisfy the *Linear Independence Constraint Qualification (LICQ)* if any subset of the indices of active constraints,  $I_a \subseteq \{i : g_i^{(j)}(\bar{\boldsymbol{\nu}}; \bar{\mathbf{x}}_t) = 0\}$ , the gradients of the subset of active constraints,  $\{\nabla_{\boldsymbol{\nu}} g_i^{(j)}(\bar{\boldsymbol{\nu}}; \bar{\mathbf{x}}_t) : i \in I_a\}$ , where  $\boldsymbol{\nu} = (\mathbf{v}, \zeta)$ , are linearly independent at  $(\bar{\mathbf{x}}_t, \bar{\boldsymbol{\nu}})$ .

Let the set of solutions to the dual of the lower-level problem (5) be defined as,

$$\Lambda(\boldsymbol{\nu}; \mathbf{x}_t) := \arg \max_{\boldsymbol{\lambda}} \inf_{\boldsymbol{\nu}} \mathcal{L}(\boldsymbol{\nu}, \boldsymbol{\lambda}^{(j)}; \mathbf{x}_t). \quad (8)$$

If a candidate solution to (7),  $(\boldsymbol{\nu}; \mathbf{x}_t)$ , results in a parameterization of the lower-level problem (2) that does not satisfy LICQ, then the set  $\Lambda(\boldsymbol{\nu}; \mathbf{x}_t)$  is not singleton, and may even be unbounded [37]. For our lower level problem (2), we may encounter such a scenario if some  $\mathbf{x}_t$  results in two of the linear collision avoidance constraints in (2) becoming collinear, and the solution for the lower level problem (2) is such that these two constraints are active. In the worst case, we will be unable to converge to a solution as there will be infinite potential values for the optimal dual variables,  $\boldsymbol{\lambda}$ . In the case that we can converge to a solution, then  $(\mathbf{x}_t, \boldsymbol{\nu})$  may be a local optimum of the reformulation (7) but not a local optimum of the bilevel problem (3) [37]. Nonetheless, since the KKT conditions are necessary and sufficient for the lower-level problem, we know that any local optimum of  $(\mathbf{x}_t, \boldsymbol{\nu})$  will be feasible with respect to the bilevel problem. In the following section we will discuss the closed loop behavior of the planner and how we use a feasible warm start to mitigate the issue of local optima.

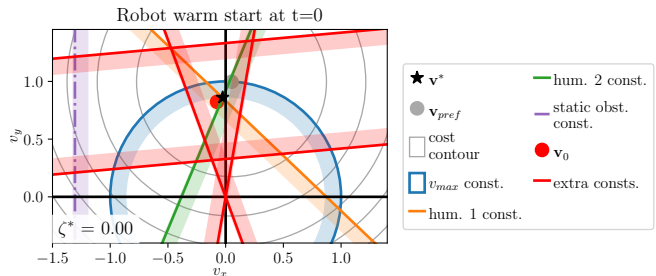


Fig. 4: Illustration of the optimization problem used to solve for robot actions for each step of the warm start for the same environment state as Fig. 2a. The robot’s kino-dynamic constraints are underestimated by the red linear constraints. The extra constraints are identically formulated as the static obstacle constraints in (2).

### A. Feasible Warm Start

We introduce a warm start strategy in order to initialize the optimization of the MPC problem (7) from a feasible trajectory and to mitigate problems caused by scenarios where the lower-level problems do not meet the LICQ. To obtain a feasible trajectory for all agents, we initialize the optimization with trajectories generated from rollouts of the ORCA policy for the robot in addition to the humans. Using ORCA for the robot will allow us to satisfy its safety constraints. In order to also satisfy the robot’s kino-dynamic constraints, we alter the ORCA optimization used for the robot. We illustrate the altered ORCA policy in Fig. 4. Unlike the humans which are modelled as holonomic integrators, the robot has non-holonomic kino-dynamic constraints (7c)-(7e) (unicycle model and limits change in velocity) described in section III-A. To account for these constraints, we add four extra pseudo-collision constraints, illustrated in red in Fig. 4, the first two (crossing lines) bound the change in heading of the robot, and the second two (parallel lines) bound the change in linear velocity of the robot. Since ORCA constraints are linear in the velocity state, we set the constraints to be an underestimate the robot’s actual constraints, hence ensuring that the rollout is feasible, albeit with more conservative constraints.

We use the results of a sequential rollout of this optimization problem as the initial guess for the robot’s actions for the first time we solve the MPC problem. For subsequent MPC optimizations, we bring forward the MPC solution from the previous time step by shifting the trajectory by one time step, and use the altered ORCA policy to update the remaining (final) time step of the MPC horizon. In the case that for some attempted MPC solution we are unable to converge to a trajectory that is feasible and lower-cost than the initial guess, e.g. when we encounter a configuration where one of the lower-level problems does not satisfy LICQ, we use the warm start solution instead.

## VI. EVALUATING ORCA FOR TRAJECTORY FORECASTING

In this paper we argue that our proposed method, SICNav, has the properties of being *interactive* in that the robot is able to explicitly model and optimize its influence on the other agents, and *safe* in that the resulting MPC policy satisfies explicit collision constraints. However, for these properties

TABLE I: Comparison of trajectory prediction methods on (a) ADE/FDE of the *maximum likelihood* prediction, (b) ADE/FDE of the *best-of-20* samples, and (c) mean KDE NLL. Our ORCA-based open-loop predictions do not achieve state-of-the-art performance but are nonetheless competitive with top-performing methods.

Dataset	(a) Maximum likelihood ADE/FDE (m) ↓				(b) Best-of-20 ADE/FDE (m) ↓				(c) Mean KDE NLL ↓		
	CVMM	S-GAN [39]	Traj++ [1]	$\mathcal{O}$ (ours)	Y-Net [2]	S-GAN	Traj++	$\mathcal{O}$ (ours)	S-GAN	Traj++	$\mathcal{O}$ (ours)
ETH	1.15/3.92	1.09/2.34	0.71/1.66	0.97/1.85	0.28/0.33	0.71/1.36	0.39/0.83	0.59/1.00	4.42	1.31	4.76
Hotel	0.23/0.74	0.69/1.44	0.22/0.46	0.28/0.51	0.10/0.14	0.37/0.72	0.12/0.21	0.13/0.21	4.22	-1.29	-0.20
Univ	0.71/0.99	0.71/1.55	0.44/1.17	0.64/1.33	0.24/0.41	0.55/1.16	0.20/0.44	0.33/0.64	5.30	-0.89	1.92
Zara 1	0.58/1.08	0.64/1.43	0.30/0.79	0.51/1.07	0.17/0.27	0.32/0.65	0.15/0.33	0.25/0.48	4.20	-1.13	0.62
Zara 2	0.43/0.34	0.52/1.14	0.23/0.59	0.39/0.82	0.13/0.22	0.30/0.63	0.11/0.25	0.21/0.40	3.62	-2.19	0.45

to hold in the real world we need to demonstrate that the ORCA policy is a sufficiently accurate model for predicting real human trajectories. The authors in [21] have previously shown that ORCA can be used to generate accurate predictions of humans in multi-agent scenarios. In this section, we follow a similar approach to validate the quality of stand-alone trajectory predictions produced by running rollouts of the ORCA algorithm for several time steps on a human trajectory dataset.

In order to generate predictions using a rollout of ORCA, we require a goal-location for each pedestrian, which we use to determine the preferred velocity for the ORCA optimization cost. The method in [21] uses a naive-Bayesian classifier to select a goal for each agent from a finite set of potential goal locations, then, performs a rollout to generate a single deterministic prediction per agent. However, a priori knowledge of such a set of potential goals is not available in many crowd navigation environments. Furthermore, most state-of-the-art trajectory forecasting methods (e.g. [1]–[3]), additionally model the uncertainty of predictions in order to account for the multi-modality of human motion. Thus, in order to fairly compare to these methods, we introduce a method to sample goal positions from a training dataset of human trajectories, then produce predictions with ORCA for each sampled goal. Unlike [21], our method does not require a finite set of a priori known goal positions, we rather sample from the distribution generated from the training split of the dataset. We compare our method with state-of-the-art trajectory forecasting methods on the ETH [26] and UCY [25] pedestrian datasets, a commonly used benchmark for trajectory forecasting.

### A. Methodology

Given a history of states (positions and velocities) with length  $T_h = 3.2s$ ,  $\mathbf{x}_{-T_h:0}^{(j)}$ , for a pedestrian, we want to predict a sequence of future positions with length  $T_p = 4.8s$ ,  $\mathbf{p}_{0:T_p}^{(j)}$ . In order to use roll-outs of the ORCA policy, we first need to predict a goal position for each agent. To this end, we use the trajectories in the training dataset to generate goal distributions to sample. The training set contains full trajectories for each agent,  $\mathbf{x}_{-T_h:T_p}^{(j)}$ . First, we create a 2D histogram of the trajectories in the training set based on average historic velocity and acceleration. In each bin, we obtain the goal location as the final position of the agent at the end of each trajectory in the bin,  $\mathbf{p}_{T_p}^{(j)}$ . For all the goal positions in each bin, we fit a 2D Kernel Density Estimate (KDE) to estimate the density of goal positions.

At test time, we have a scene with  $N$  pedestrians. For each agent in the scene, we use its average historic velocity to select the appropriate goal KDE, and obtain 2000 samples of goal positions for each agent. This way, we are drawing independent goal samples for each agent. However, in order to perform a rollout of the scene using ORCA, we need to perform a rollout for the entire scene, i.e. we need samples from the joint-distribution of goals for every agent. If we naively combine the goals of each agent to get 2000 joint samples, we ignore the fact that a highly likely goal sample for one agent may get matched with a low likely goal for another. To address this issue, we calculate importance weights for the combined goal samples. We sum the log-likelihoods of each individual goal sample over the number of agents in the scene and normalize with respect to the 2000 samples. Now, for each of these 2000 joint samples, we perform an ORCA rollout.

### B. Results

We follow the *leave-one-out* strategy commonly used with the ETH/UCY datasets [1], [2], where one dataset is selected as the test set, and the others are used for training (generating the goal KDEs in our formulation). Following the metrics used in existing literature [1], [2], we evaluate prediction using,

- 1) Average Displacement Error (ADE): the mean Euclidean distance between the ground truth and the maximum likelihood prediction.
- 2) Final Displacement Error (FDE): the Euclidean distance between the maximum likelihood predicted position at the final prediction step,  $T_p$ , and the ground truth.
- 3) Best-of-20: the minimum ADE and minimum FDE from 20 randomly sampled prediction trajectories.
- 4) KDE-based Negative Log Likelihood (NLL) [40]: the mean NLL of the ground truth trajectory under the distribution of a KDE fit to the predicted trajectory samples.

To find the maximum likelihood prediction from our model, we follow a similar approach as calculating the KDE-based NLL metric, however, instead of evaluating the likelihood of the ground truth under the KDEs, we evaluate the likelihoods of the predicted samples themselves. We compare our method against two state-of-the-art prediction models SocialGAN [39], Trajectron++ [1] and Y-net [2]. For the maximum likelihood metrics, we also compare our method to a Constant Velocity Motion Model (CVMM) that simply projects the current velocity of the agent forward in time. We were unable to replicate the results of Y-net, and thus cannot report the KDE-NLL metric.

The results of our evaluation are summarized in Table I. We can observe that our ORCA-based approach does not achieve state-of-the-art performance but is nonetheless competitive with top-performing methods. Our method beats S-GAN and CVMM on almost all metrics for all data splits. Our method has the added benefit of being able to be integrated into downstream planning, which is challenging for the other methods.

## VII. EVALUATION OF SICNAV

### A. Simulation Experiments

*Testing environment:* We evaluate the performance of our planner in a commonly used crowd navigation simulation environment [14], which we extend to simulate static obstacles in addition to the original dynamic agents, and to enable the simulated agents to be controlled by Social Force Model (SFM) [41] in addition to the original ORCA. We conduct two sets of experiments, one where the simulated humans follow ORCA and another where they follow SFM. For each set, we generate 500 random initial and final positions for the humans environments with  $N = 3$  and  $N = 5$  simulated humans to get a total of 2000 runs. The simulated testing area is a  $2m$ -wide corridor with initial positions and goals on either end. In order to ensure a high density of agents in the corridor, we constrain the space using static obstacles, formulated as line-segments, and add a bottleneck that simulates a  $1m$  wide doorway. Fig. 6 highlights an interaction at this doorway. In every scenario, the robot is required to move through the simulated doorway to a goal  $3m$  directly in front its fixed initial position. Furthermore, every agent must cross the doorway in order to arrive at its goal, ensuring that interactions occur in these scenarios.

*Performance metrics:* We adopt several metrics from the literature [10], [14], [28]. As the robot proceeds to the goal, data from the trajectories of the robot and agents is collected to score the behavior in the following ways,

- *Success Rate:* The rate of scenarios in which the robot is able to successfully arrive at its goal within  $90s$ , which measures how the overall success of the robot to arrive at its goal.
- *Average Navigation Time:* The average time to completion, which indicates the robot’s time efficiency.
- *Collision Frequency:* The ratio of time steps that the robot is in a collision state with another agent compared to the total number of time-steps in the 500 scenarios, which measures the safety of the robot.
- *Freezing Frequency:* The frequency of time steps where the robot needed to reduce its velocity to zero, which measures how often the robot can potentially get stuck in a deadlock with the human agents (aka the encountering the FRP).

*Algorithms:* We compare two variants of our algorithm with four baseline methods. The first version of our algorithm, SICNav-p, has access to the ground truth goal of position and ORCA parameters of the human agents. The second version, SICNav-np, does not have access to any privileged information about the agents. In order to estimate the intended goal of the agent, SICNav-np projects the current velocity of the

TABLE II: Quantitative comparison in 500 randomly generated test scenarios with 1 robot and 3 or 5 simulated humans following ORCA. Arrows indicate desired relative values. Our method outperforms baselines across the metrics.

Approach	Success Rate $\uparrow$	Avg Nav Time (s) $\downarrow$	Collision Freq. $\downarrow$	Frozen Freq. $\downarrow$
$N = 3$ humans				
SICNav-p (ours)	<b>1.00</b>	<b>4.24</b>	<b>0.00</b>	<b>0.01</b>
SICNav-np (ours)	<b>1.00</b>	4.37	<b>0.00</b>	<b>0.01</b>
MPC-CVMM	0.98	7.06	0.01	0.05
ORCA [18]	<b>1.00</b>	10.26	0.01	0.07
SARL [14]	0.73	27.21	0.01	0.26
RGL [15]	0.95	8.20	0.01	0.06
DWA [4]	0.57	52.09	0.01	0.55
$N = 5$ humans				
SICNav-p (ours)	<b>1.00</b>	<b>6.35</b>	<b>0.01</b>	<b>0.03</b>
SICNav-np (ours)	<b>1.00</b>	<b>6.27</b>	<b>0.01</b>	<b>0.02</b>
MPC-CVMM	<b>1.00</b>	7.47	0.05	0.07
ORCA [18]	<b>1.00</b>	14.98	0.02	0.09
SARL [14]	0.97	6.45	0.05	0.04
RGL [15]	0.50	46.86	0.03	0.48
DWA [4]	0.68	48.45	0.03	0.50

TABLE III: Quantitative comparison in 500 randomly generated test scenarios with 1 robot and 3 or 5 simulated humans following SFM. Arrows indicate desired relative values. Compared with performance in the ORCA simulations, our method does not show much degradation in any of the metrics.

Approach	Success Rate $\uparrow$	Avg Nav Time (s) $\downarrow$	Collision Freq. $\downarrow$	Frozen Freq. $\downarrow$
$N = 3$ humans				
SICNav-p (ours)	<b>1.00</b>	4.84	0.01	<b>0.01</b>
SICNav-np (ours)	<b>1.00</b>	<b>4.81</b>	0.01	<b>0.01</b>
MPC-CVMM	0.99	6.25	0.02	0.03
ORCA [18]	<b>1.00</b>	6.10	0.01	<b>0.01</b>
SARL [14]	0.64	34.82	0.02	0.36
RGL [15]	0.95	7.49	0.02	0.05
DWA [4]	0.96	16.66	<b>0.00</b>	0.10
$N = 5$ humans				
SICNav-p (ours)	<b>1.00</b>	5.42	0.03	<b>0.02</b>
SICNav-np (ours)	<b>1.00</b>	<b>5.14</b>	0.02	<b>0.01</b>
MPC-CVMM	0.98	7.30	0.05	0.05
ORCA [18]	<b>1.00</b>	7.83	0.03	0.02
SARL [14]	0.92	10.27	0.03	0.08
RGL [15]	0.17	75.56	<b>0.01</b>	0.80
DWA [4]	0.95	20.78	<b>0.01</b>	0.14

agent forward in time, and uses the resulting position as the goal of the human. We compare these two versions of our algorithm with the following baselines: MPC-CVMM has the same setup as SICNav, however with a CVMM instead of the lower-level ORCA model for other agents, i.e. each human is assumed to continue at their current velocity for the complete horizon. We also compare with state of the art RL methods, Socially Attentive Reinforcement Learning (SARL) [14], and Relational Graph Learning (RGL) [15], the robot itself also running ORCA [18], and the classical Dynamic Window Approach (DWA) [4] as a reactive baseline. Note that since ORCA does not have a method to incorporate dynamics constraints by default, the ORCA robot follows a holonomic model without any kino-dynamic (unicycle) constraints, which means that it solves a simpler problem than the other methods.



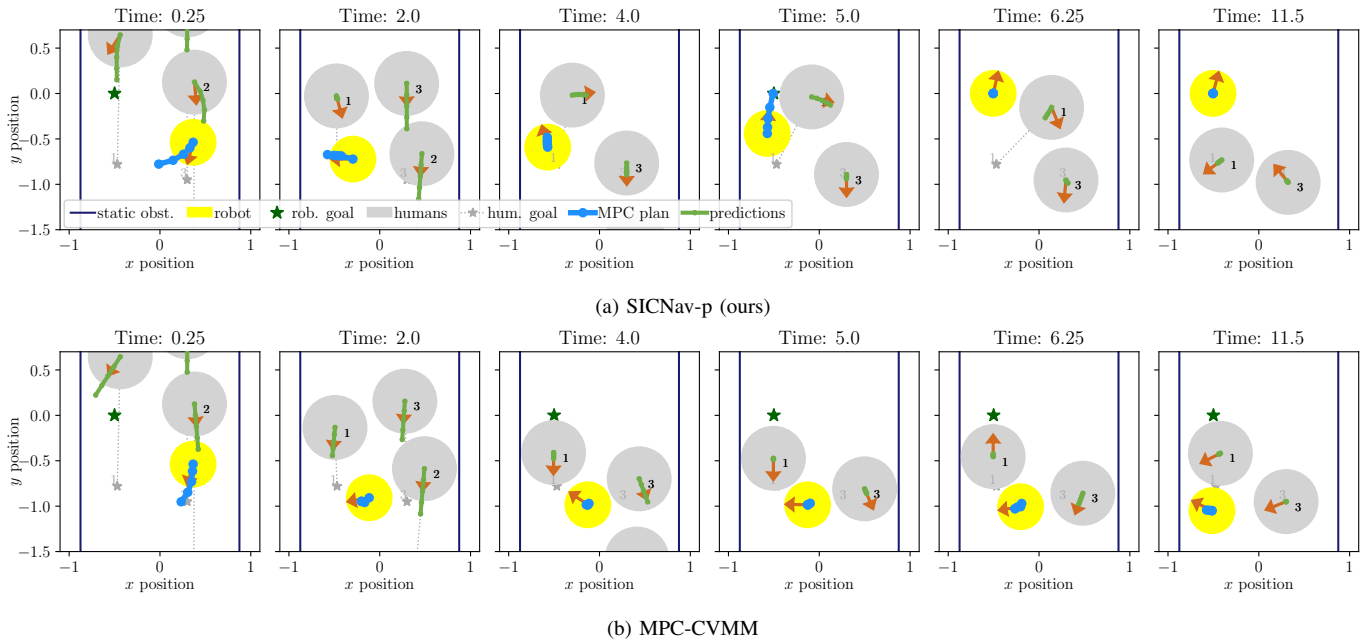


Fig. 5: Snapshots through time of a robot (yellow) using our SICNav approach (a) and the MPC-CVMM baseline approach (b), near a goal location (green star) obstructed by simulated human agents (gray circles). At time 0.25, the robot and the humans are moving downwards, and both the SICNav and MPC-CVMM robot plan to turn to the right towards their goal (MPC plan in blue). Our SICNav robot arrives at the goal by time 6.25 by predicting that human 1 will move out of its way (predictions in green), then proceed to its own goal, arriving at time 11.5. Meanwhile, the robot using the baseline MPC-CVMM method is unable to coordinate with human 1. Animation: [tiny.cc/sicnav\\_fig5](http://tiny.cc/sicnav_fig5).

*Implementation*<sup>1</sup>: We implement SICNav in Python using CasADi [42]. To avoid infeasibility we introduce slack variables on the constraints along with quadratic penalty functions. We use a simulation time step of  $\delta t = 0.25s$  and an MPC horizon of  $T = 1s$ . The RL methods are implemented using Stable Baselines [43]. The cost functions for both RL methods have been altered to include penalties analogous to all components of SICNav’s cost and constraints described in Sec. IV, and the one-step CVMM propagation used to calculate the respective value functions has been replaced with ORCA in order to allow fairer comparisons to our SICNav method, which uses the ORCA model in the lower-level problem. We train the RL algorithms with the same arrangement of static obstacles as the test environment, but with a training set of human initial and goal positions that are not included in the test set. We train separate models for  $N = 3$  and  $N = 5$ . We intend to make the code for the simulator and methods publicly available upon acceptance.

*Quantitative Results and Discussion*: Tables II and III summarize the performance of our method and the baselines in the simulator with ORCA agents and SFM agents, respectively. In the following section we analyze the average performance of the crowd navigation methods. A statistical analysis of the significance of these results is provided in Appendix A.

In simulation scenarios involving three humans simulated by ORCA, the two variants our method, SICNav-p and SICNav-np, achieve a success rate of 1.00, beating all other methods, except ORCA which achieves the same success rate. Accompanying SICNav-p’s success rate are the shortest average

navigation time (4.24s) and the lowest rates of collision (0.00) and robot freezing (0.01). Removing privileged information results in negligibly slower navigation time for SICNav-np (4.37), without affecting collision and freezing rates. Using the ORCA for robot plans, however, demonstrates a much longer navigation time in excess of 10s, and a collision rate of 0.01. This poorer performance compared to SICNav-p is despite the fact that the ORCA method is solving a simpler problem without kino-dynamic constraints. RGL demonstrates an average navigation time of 8.20s and success rate of 0.95. The MPC-CVMM method achieves a slightly higher success rate at 0.98, but requires a longer average navigation time (7.06s). SARL, experiences a lowered success rate (0.73), a high average navigation time (27.21s), and increased rates of collision (0.01) and robot freezing (0.26). DWA records the lowest success rate (0.57), the longest average navigation time (52.09s), and a high rate of robot freezing (0.55).

In the scenarios with five humans simulated by ORCA, all SICNav-p, SICNav-np, MPC-CVMM, and ORCA maintain a success rate of 1.00. Similar to the three human tests, SICNav-p and SICNav-np perform better than all the baselines across all four metrics, indicating our method’s consistent performance under increasing crowd densities. Unlike the three human tests, SICNav-np posts the shortest average navigation time at 6.27s compared to SICNav-p’s 6.35s, with both versions of our method maintaining the lowest rates for collision (0.01) and robot freezing (0.02, and 0.03, respectively). MPC-CVMM and ORCA record longer average navigation times (7.47s and 14.98s) and show substantially higher rates of collisions (0.05 and 0.02) and robot freezing (0.05 and 0.09). In comparison, SARL reached an average navigation time

<sup>1</sup>Code: [github.com/sepsamavi/safe-interactive-crowdnav.git](https://github.com/sepsamavi/safe-interactive-crowdnav.git).

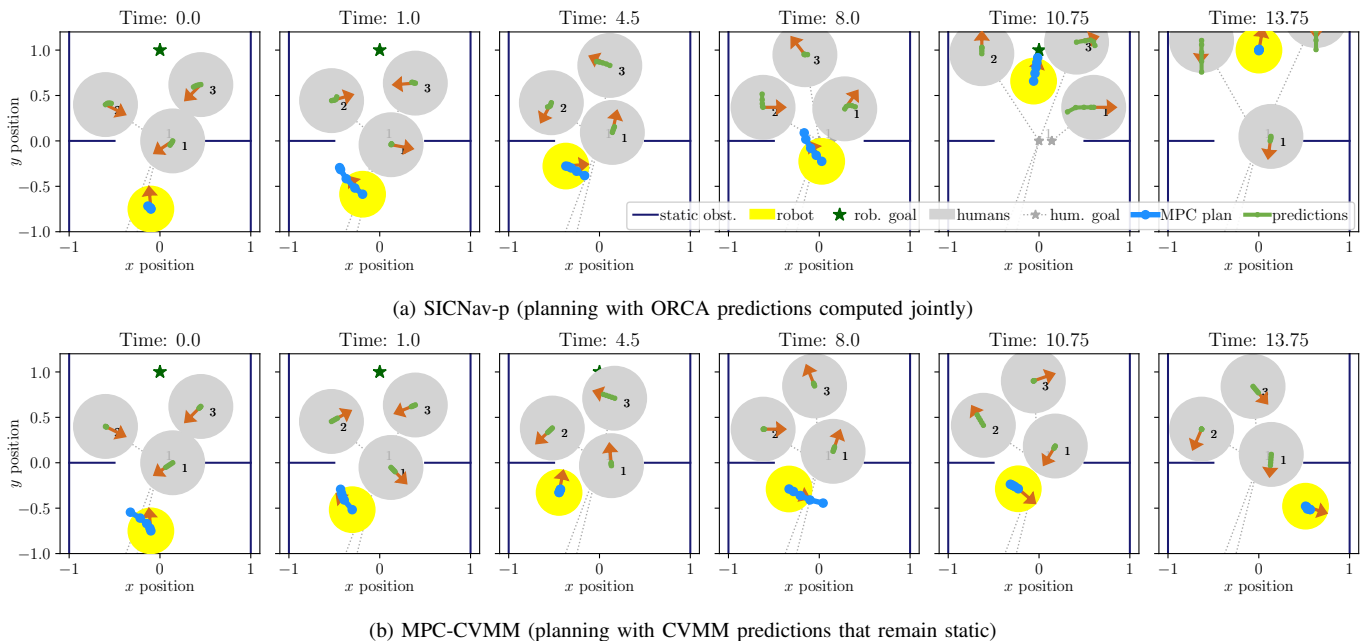


Fig. 6: Snapshots through time of a robot (yellow) using our SICNav-p approach (a) and the MPC-CVMM baseline approach (b), near a goal location (green star) obstructed by simulated human agents (gray circles). The goal position of human 1 is in the way of the robot’s path to its goal. At time 0.0, the robot begins moving towards the doorway. As shown in the images, our SICNav-p robot continues moving to its goal by predicting that agents 1-3 will move out of its way. However, in the same scenario, the robot using the baseline MPC-CVMM method oscillates between positions on the left and right of agent 1, who remains blocking the doorway. Animation: [tiny.cc/sicnav\\_fig6](http://tiny.cc/sicnav_fig6).

of 6.45s, but with a success rate of 0.97 and also with a substantial rate of collisions (0.05). RGL and DWA, under the same conditions, demonstrate lower success rates (0.50 and 0.68 respectively), extended navigation times (46.86s and 48.45s) and increased rates of collisions and robot freezing.

In simulation scenarios involving three humans simulated by SFM, the two variants of our method, SICNav-p and SICNav-np, achieve a perfect success rate of 1.00, maintaining consistent performance with the different human simulation model. SICNav-np, slightly outperforms SICNav-p in terms of average navigation time, recording 4.81s compared to 4.84s with similar collision frequencies of 0.01. The baseline methods exhibit varied performance. For instance, ORCA shows improved navigation time (6.10s) for this human simulation model but with an equivalent collision frequency of 0.01. The MPC-CVMM method achieves a success rate slightly lower at 0.99 and a longer average navigation time (6.25s). SARL significantly struggles with the lowest success rate (0.64) and a much higher average navigation time (34.82s), along with increased collision and freezing rates, suggesting a severe degradation of performance when faced with novel behavior from the simulated humans. With this human simulation model, DWA outperforms our methods in terms of collision frequency (0.00) however at the expense of a lower success rate (0.96), a lengthier navigation times (16.66s), and increased freezing frequency (0.10).

In scenarios with five humans simulated by SFM, both of our models (SICNav-p and SICNav-np) continue to exhibit perfect success rates. SICNav-np presents shorter navigation times (5.14s), slightly better than SICNav-p’s 5.42s. The collision rates for both our models remain low but slightly

increased to 0.02 and 0.03, respectively, indicating our model being strained under higher densities of SFM humans. MPC-CVMM and ORCA demonstrate longer navigation times (7.30s and 7.83s) and higher rates of both collisions and freezing. SARL’s success rate depletes to 0.92 with an inflated navigation time of 10.27s. RGL’s performance notably deteriorates, culminating in a drastically reduced success rate of 0.17 and an extended average navigation time. The degradation in performance of the RL methods in the SFM simulator demonstrates the challenge these methods have when faced with human behavior that deviates from the training data. DWA, however, maintains a high success rate of 0.95 and a low collision frequency of 0.01, but with a longer navigation time (20.78s) and increased freezing frequency (0.14).

The results in with the SFM simulated humans are consistent with the results from the ORCA simulated humans. Our proposed methods, SICNav-p and SICNav-np, do not show much degradation in any of the metrics, demonstrating the robustness of our approach to different human simulation models.

*Qualitative Analysis:* To qualitatively demonstrate the impact of SICNav-p generating robot plans jointly with predictions, in Fig. 5, we illustrate a scenario in which the robot cannot access its goal unless it interacts with other agents blocking its way. Since our SICNav-p method can proactively expect the reaction of the human agents, it slowly approaches the agents in its way, who move to allow the robot to pass in response. Note that the robot is not colliding with the other agents, the agents move out of its way and allow it to pass. On the same scenario we compare the baseline MPC-CVMM, whose cost and constraints are identical to SICNav-p except

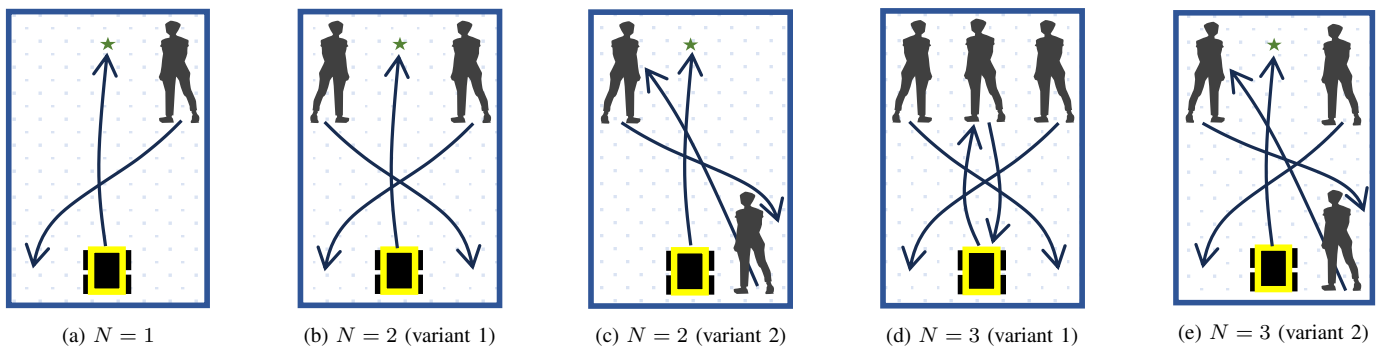


Fig. 7: Types of scenarios that constitute the 80 runs of the robot tested with humans (not to scale). The robot (black and yellow) follows a straight path to the goal (green star). The humans start from the positions shown in the figure and move to the opposite side of the testing area as shown by the solid black arrows. The agents begin moving at the same time. All scenarios were designed to induce a conflict zone in the middle of the room where all agents' paths cross. Video of scenarios: [tiny.cc/sicnav\\_robot\\_scenarios](http://tiny.cc/sicnav_robot_scenarios).

for the predictions of the other agents which use a CVMM model and remain constant in the planning optimization. Starting from an almost identical position, MPC-CVMM gets stuck as it is unable to account for the cooperation of the other agents.

Fig. 6 illustrates a similar but more complex scenario where the robot cannot access its goal unless it interacts with other agents blocking a doorway. In this scenario, the SICNav-p robot is able to move through the doorway by predicting that the agents will move out of its way. However, the baseline MPC-CVMM robot oscillates back and forth around without proceeding through the doorway.

## B. Real Robot Experiments

*Testing Environment:* We also evaluate the performance of our planner, SICNav-np, on a real robot in a controlled environment. We use a Clearpath Jackal differential drive robot pictured in Fig. 1, which weighs  $20kg$  and measures  $46cm$  W  $\times$   $60cm$  L  $\times$   $50cm$  H. Our operating environment is an indoor space measuring  $5m \times 9m$ . We conduct a total of 80 runs from five scenarios with a varying number of humans  $N \in \{1, 2, 3\}$ . The scenarios are illustrated in Fig. 7. For each run of a particular scenario, all agents were instructed to move from their starting positions (where the agents are illustrated in Fig. 7) to the opposite corner of the room (illustrated by the arrows in Fig. 7). In the scenarios with multiple humans, we tested two variants: one where all the humans start from the opposite side of the room than the robot (variant 1) and another where one human moved in the same direction as the robot (variant 2). These scenarios were designed to cause a conflict zone in the middle of the room where all agents need to interact. We repeated 20 runs for each scenario with  $N \in \{1, 2\}$  humans and 10 runs for each scenario with  $N = 3$ . We use a VICON motion capture system to measure the positions of the robot and human agents in the environment and use an Extended Kalman Filter (EKF) and Kalman Filters (KFs) to track the positions and velocities of the robot and humans, respectively.

*Algorithms:* We compare the performance our SICNav-np method with MPC-CVMM as a baseline. As noted in Sec. VII-A, SICNav-np, does not have access to any privileged

information about the agents. In order to estimate the intended goal of the agent, SICNav-np projects the current velocity of the agent forward in time, and uses the resulting position as the human's goal. MPC-CVMM has the same setup as SICNav-np, however with a CVMM instead of the lower-level ORCA model for other agents, i.e. each human is assumed to continue at their current velocity for each step of the MPC horizon. By comparing our method with MPC-CVMM, we aim to isolate our analysis specifically to the primary contribution of this paper: the effect of using ORCA to model the humans and jointly produce robot plans with predictions.

*Implementation:* The robot used the Robot Operating System (ROS) 1 middleware. The robot accepts linear and angular velocity commands, which are then executed by its low level controller. We run the algorithms under analysis on an 8-core computer using Intel<sup>(R)</sup> Core<sup>(TM)</sup> i7-9700K CPU @ 3.6 GHz. We use the same Python implementation of our algorithm as the simulation experiments (Sec. VII-A), which uses CasADi as the modelling language. However, we replace the solver with Acados [44], which generates C++ code from the CasADi models and uses Just-in-time Compilation to allow us to run our algorithm in real-time at 10 Hz. We keep the same discretization step as the simulation of  $\delta t = 0.25s$  and use an MPC horizon of  $T = 2s$ .

*Quantitative Results and Discussion:* We evaluate the robot's path efficiency by measuring the average distance of the robot from the shortest path to its goal in each run. In Fig. 8a, we observe that in scenarios with  $N \in \{2, 3\}$ , SICNav-np has a lower average deviation from the shortest path, with  $0.13m$  and  $0.17m$ , compared to MPC-CVMM, with  $0.30m$  and  $0.27m$ . Furthermore, we observe that the distribution of values for MPC-CVMM is larger than for SICNav-np. These results indicate that SICNav-np is more efficient in terms of the path it follows, and more consistently chooses the shorter path.

We evaluate the robot's safety by measuring the total duration of time the robot is too close to any human (body of robot is  $< 0.35m$  to centroid of human) in each run. In Fig. 8b we observe that in the scenarios with  $N \in \{1, 2\}$  there is not much difference in the average amount of time the robot is too close to a human between the two algorithms,

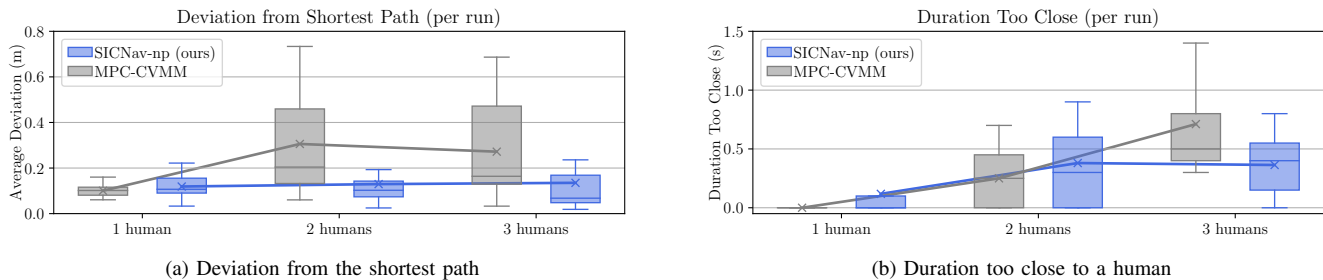


Fig. 8: Box-and-whisker plots of performance metrics obtained when running SICNav-np (blue) and MPC-CVMM (gray) in scenarios with varying numbers of humans. Median values are illustrated by the line in the box-and-whisker plots and mean values are indicated by the  $\times$ . We plot (a) the robot’s average deviation from the shortest path in each run, and (b) the total duration of time that the robot is too close to any human in each run. As crowd size increases, using our method instead of the baseline results in a lower deviation from the shortest path, demonstrating better path efficiency, and a lower duration too close to a human, demonstrating better safety.

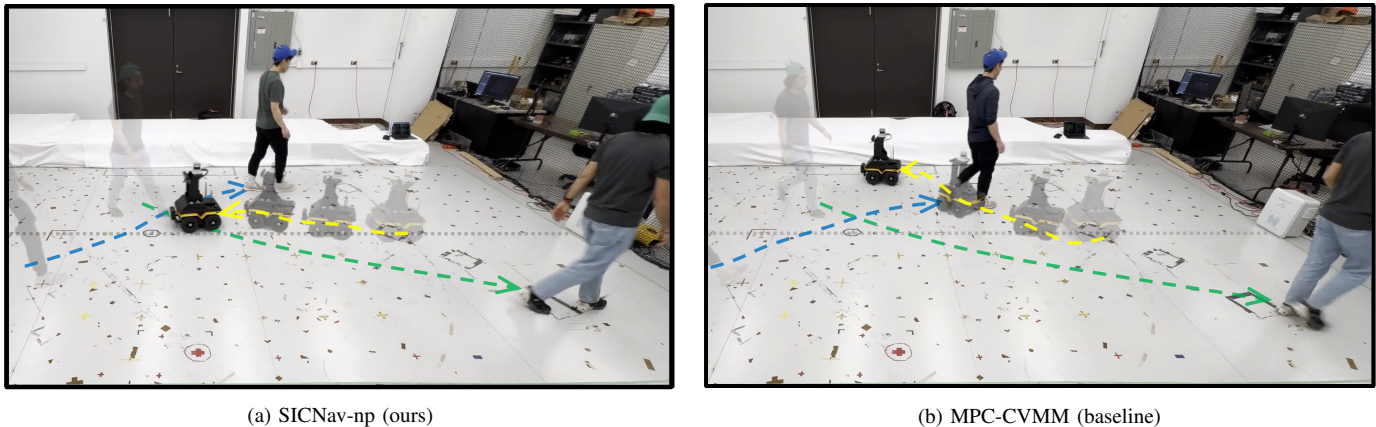


Fig. 9: Videos of one robot and two humans rendered as images. Dashed lines indicate paths traversed by each agent (yellow for robot, green and blue for humans). Transparent images illustrate older frames of the video. For both SICNav-np (a) and the baseline (b), the robot begins moving towards its goal in the center of the room (gray dotted line). Our SICNav robot’s path (yellow dashed line in (a)) deviates less from the center of the room (gray dotted line) than the path of the robot using the baseline MPC-CVMM method (yellow dashed line in (b)). Video: [tiny.cc/sicnav\\_fig9](https://tiny.cc/sicnav_fig9).

with an average total time of  $0.12s$  and  $0.38s$  for SICNav-np compared to  $0.00s$  and  $0.26s$  for MPC-CVMM. However, in the case with  $N = 3$  humans, we can see that SICNav-np has a lower average duration too close to a human, with  $0.36s$ , compared to  $0.77s$  for MPC-CVMM. While SICNav-np and MPC-CVMM perform similarly with respect to scenarios with two or fewer humans, in scenarios with a higher number of humans, SICNav-np maintains roughly the same amount of duration too close to any human, while the time for MPC-CVMM almost doubles, indicating that SICNav-np is more successful in avoiding close interactions with humans. We can also observe that while the size of the interquartile range is similar between the two methods for cases with  $N \in \{1, 2\}$ , in the cases with  $N = 3$ , the interquartile range is smaller for SICNav-np, indicating that SICNav-np is more consistent in avoiding close interactions with humans. *Qualitative Analysis:* Reviewing the videos of the runs, we observe that the robot using SICNav-np would often slow down and wait for the human to cross its path before proceeding. We illustrate this observation in Fig. 9 (Video: [tiny.cc/sicnav\\_fig9](https://tiny.cc/sicnav_fig9)). The SICNav-np robot predicts that the humans will alter their paths to pass each other and with the robot. As such the robot slows down. In contrast, the robot using MPC-CVMM,

predicts that the humans would continue at their current speed and heading. Thus, the MPC-CVMM robot turns to the right to avoid the human’s path leading to a larger deviation from the shortest path than the SICNav-np robot. The observations from this scenario are consistent with the results in Fig. 8a, where we observe that SICNav-np has a lower average deviation from the shortest path than MPC-CVMM.

In Fig. 10, we illustrate a scenario where the robot is confronted by humans acting adversarially towards the robot (Video: [tiny.cc/sicnav\\_fig10](https://tiny.cc/sicnav_fig10)). Fig. 10a illustrates the beginning of this scenario: as the robot proceeds towards its goal (green star), the humans with the blue and red hats walk towards the robot without attempting to avoid it. Snapshots of the top-view are illustrated in the first three plots in Fig. 10c, starting with the humans walking toward the robot at time  $00:02$ . At times  $00:02$  and  $00:03$ , we observe that the predictions produced by SICNav-np (green lines) expect the humans to go around the robot. In the associated robot MPC plans (blue lines) we can see the robot planning to reverse away from the humans in the first few time steps of the MPC horizon, then proceed forward in the latter part of the horizon. However, in this scenario, the humans do not follow the robot’s predictions that they will deviate their path to avoid the robot,

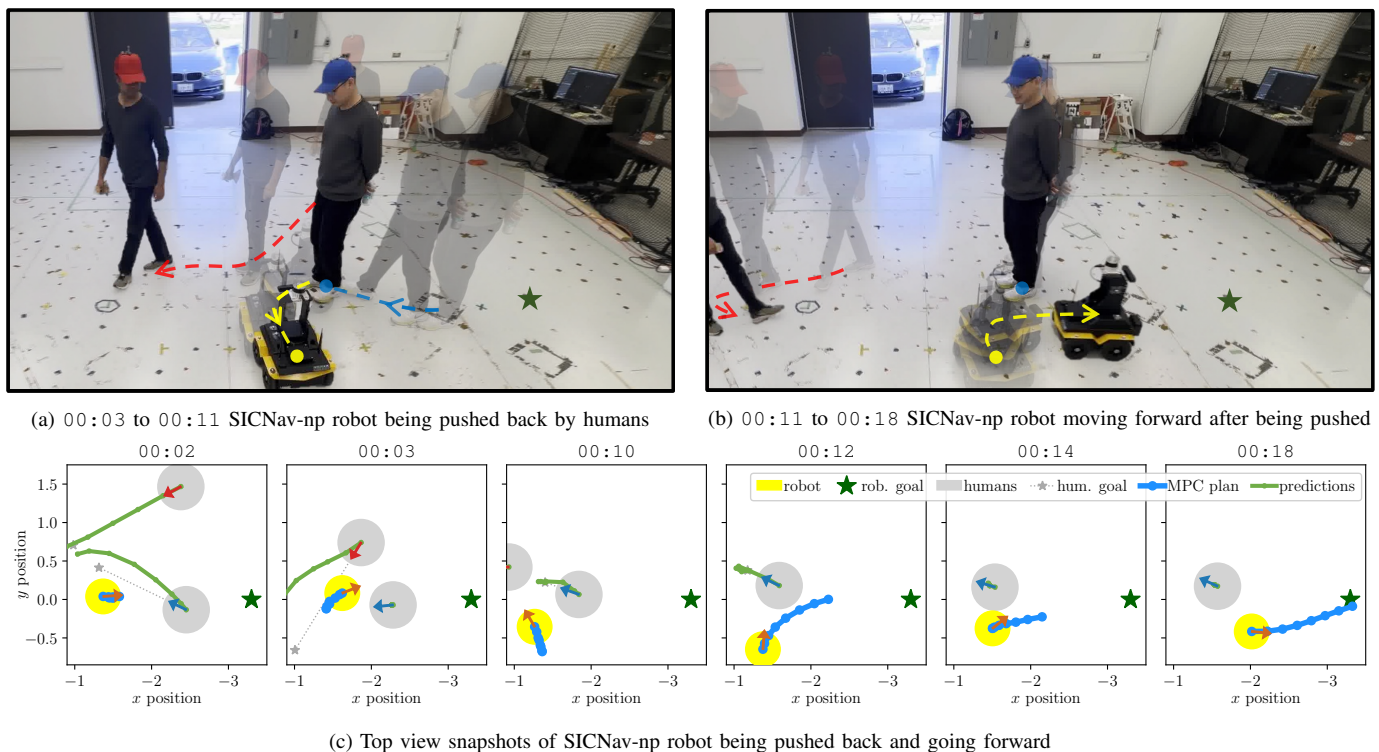


Fig. 10: Videos of one robot and two humans rendered as images. Dashed lines indicate paths traversed by each agent (yellow for robot, blue and red for humans). Transparent images illustrate older frames of the video. The robot goal is illustrated as the green star. (a) As the robot proceeds towards its goal, the humans with the blue and red hats walk towards the robot without attempting to avoid it, resulting in the robot being pushed back. (b) After the humans stop pushing, the robot comes to a stop and moves forward around the human with the blue hat to proceed to its goal. In (c), we illustrate snapshots of the top-view visualization corresponding to (a) and (b). Video: [tiny.cc/sicnav\\_fig10](https://tiny.cc/sicnav_fig10).

instead they proceed toward the robot. This human behavior deviates from our assumption that the humans follow ORCA. However, since the robot follows SICNav-np in receding horizon fashion, re-planning at 10Hz, new plans take into account the fact that the humans are getting closer to the robot than the SICNav-np prediction expected. As such, subsequent robot plans for the robot to reverse at the beginning of the MPC horizon in order to continue avoiding the humans. Thus, we observe the robot being pushed back, as illustrated in Fig. 10a. Once the humans stop behaving adversarially, at time 00:11 in this scenario, the robot predictions become more accurate (e.g. human with the blue hat will remain roughly stationary at 00:14), and the robot proceeds to its goal as illustrated in Fig. 10b. In this scenario, we observe the robustness of our method to adversarial behavior from the humans. In such scenarios, the robot only needs to avoid collisions for one time step of the MPC horizon. If its predictions are incorrect, as long as the robot can avoid the humans for one time step, it can re-plan and continue avoiding collisions.

### C. Limitations and Future Work

Our evaluation of SICNav in simulation and with a real robot operating in a lab setting show promising results. Our algorithm runs in real time and enables the robot to interact with other agents while avoiding collisions in tight scenarios with multiple agents. Nonetheless, our paper has several limitations that we intend to address in future work.

*Knowledge of Human Intents:* SICNav requires information about the human agents' intended goals in order to predict their future trajectory. In SICNav-np, which does not require knowledge of human goals, we project each agent's current velocity forward in time to estimate their goal. While this method shows promising performance in our real-robot experiments (Sec. VII-B), in future work, we plan to use a learning-based short-term goal prediction (e.g. [2]) to generate goal estimates that we can then use in the ORCA models integrated into SICNav. To train, we intend to combine a goal predictor with open-loop ORCA rollouts and use a standard prediction loss (e.g. [1], [2]), leveraging the implicit function theorem to differentiate the ORCA  $\arg \min\{\cdot\}$  as a function. We can even extend this approach to infer the other ORCA parameters from data as well. Our evaluation of ORCA rollouts as an open-loop prediction model (Sec. VI) has shown promising results, even with a rudimentary empirical goal sampling method and ORCA parameters fixed. We also intend to investigate incorporating uncertainty of the predicted ORCA parameters into the SICNav bilevel optimization, e.g. by following a robust bilevel optimization approach [45].

*Evaluation Scope:* In this work, we evaluate SICNav in simulation and on a real robot operating indoors. In our robot experiments, we use a VICON motion tracking system to obtain position measurements (mm error) of the humans head position in the scene, and use KFs to estimate their positions and velocities. The first limitation of this setup is that we

would not have access to such motion tracking infrastructure in a deployment in the wild. Nonetheless, recent human motion estimation work (e.g. [46]) has shown promising results in estimating and predicting human motion from only image data to centimeter accuracy. Furthermore, our estimates of the human centroid did not meet the mm accuracy possible VICON: a second limitation of our experimental setup is that we only tracked the location of each agent's head. Thus, our estimates of the person's location were noisy when the person swayed or leaned, e.g. to initiate or stop walking. Previous work using images and lidar [47], has shown position tracking Root Mean Squared Error (RMSE) of less than 15cm for nearby agents. Toward the key next step of deploying SICNav on a real robot in the wild, we intend to evaluate our algorithm's performance when using the onboard lidar and camera sensors to track the positions of the agents in the scene.

*Number of Interactive Humans:* In this work, we demonstrate the real-time (10Hz) performance of our algorithm in environments with up to  $N = 3$  humans modeled using ORCA for a horizon of  $T = 8$  time steps. The dimension of variables for the MPC optimization problem (7) associated with the KKT-reformulation scale quadratically in the number of humans times the control horizon,  $O(N^2T)$ . As a result, solving for control may become slower than real time for scenarios with many agents ( $\gg 3$ ). In future work, we intend to limit the number of agents modeled interactively using ORCA to a neighborhood of the 3–4 most 'important' humans each time we solve (7) and use the solution in receding horizon fashion, similar to [6], [11].

## VIII. CONCLUSION

In this paper we presented SICNav, a novel MPC planner for robot navigation in crowded environments. We formulated the problem as a bilevel optimization problem, where the robot plans its trajectory while jointly predicting the trajectories of other agents using ORCA. To evaluate our approach, we first validated the effectiveness of ORCA in generating open-loop predictions of human motion on a benchmark dataset. Then, we demonstrated the effectiveness of our planning method in simulation environments with ORCA-simulated humans and SFM-simulated humans, and showed that our method outperforms state-of-the-art methods in terms of success rate, navigation time, and collision avoidance. We also qualitatively evaluated the ability of our method to interact with other agents in tight simulation scenarios. Finally, we evaluated our method on a real robot in a lab environment, demonstrating our methods improved navigation performance when compared to a baseline. We also qualitatively evaluated our method in cases where the humans deviate from behaving as expected under ORCA to demonstrate our method's robustness.

## ACKNOWLEDGEMENT

We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC).

Computing resources used in preparing this research were provided, in part, by the Province of Ontario, the Government of Canada through CIFAR, and companies sponsoring the Vector Institute for Artificial Intelligence.

## REFERENCES

- [1] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, "Trajectron++: Dynamically-Feasible Trajectory Forecasting With Heterogeneous Data," in *2020 European Conference on Computer Vision (ECCV)*, 2020. [Online]. Available: <http://arxiv.org/abs/2001.03093>
- [2] K. Mangalam, Y. An, H. Girase, and J. Malik, "From Goals, Waypoints & Paths To Long Term Human Trajectory Forecasting," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Montreal, QC, Canada: IEEE, Oct. 2021, pp. 15 213–15 222. [Online]. Available: <https://ieeexplore.ieee.org/document/9709992/>
- [3] J. Yue, D. Manocha, and H. Wang, "Human Trajectory Prediction via Neural Social Physics," in *Proceedings of the European Conference on Computer Vision (ECCV)*. arXiv, July 2022, arXiv:2207.10435 [cs]. [Online]. Available: <http://arxiv.org/abs/2207.10435>
- [4] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, Mar. 1997. [Online]. Available: <doi.org/10.1109/100.580977>
- [5] M. Khatib, H. Jaouni, R. Chatila, and J. Laumond, "Dynamic path modification for car-like nonholonomic mobile robots," in *Proceedings of International Conference on Robotics and Automation*, vol. 4. Albuquerque, NM, USA: IEEE, 1997, pp. 2920–2925. [Online]. Available: <http://ieeexplore.ieee.org/document/606730/>
- [6] N. E. Du Toit and J. W. Burdick, "Robot Motion Planning in Dynamic, Uncertain Environments," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 101–115, Feb. 2012. [Online]. Available: <doi.org/10.1109/TRO.2011.2166435>
- [7] C. Rösmann, F. Hoffmann, and T. Bertram, "Integrated online trajectory planning and optimization in distinctive topologies," *Robotics and Autonomous Systems*, vol. 88, pp. 142–153, Feb. 2017. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0921889016300495>
- [8] D. Mayne, J. Rawlings, C. Rao, and P. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, June 2000. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0005109899002149>
- [9] S. Poddar, C. Mavrogiannis, and S. S. Srinivasa, "From Crowd Motion Prediction to Robot Navigation in Crowds," in *IEEE/RSSJ International Conference on Intelligent Robots and System (IROS)*, Detroit, MI, USA, Mar. 2023. [Online]. Available: <http://arxiv.org/abs/2303.01424>
- [10] P. Trautman and A. Krause, "Unfreezing the robot: Navigation in dense, interacting crowds," in *International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2010, pp. 797–803. [Online]. Available: <https://doi.org/10.1109/IROS.2010.5654369>
- [11] P. Trautman, J. Ma, R. M. Murray, and A. Krause, "Robot navigation in dense human crowds: Statistical models and experimental studies of human-robot cooperation," *International Journal of Robotics Research*, vol. 34, no. 3, pp. 335–356, 2015. [Online]. Available: <doi.org/10.1177/0278364914557874>
- [12] M. Sun, F. Baldini, P. Trautman, and T. Murphey, "Move Beyond Trajectories: Distribution Space Coupling for Crowd Navigation," in *Robotics: Science and Systems XVII*. Robotics: Science and Systems Foundation, July 2021. [Online]. Available: <doi.org/10.15607/RSS.2021.XVII.053>
- [13] D. Sadigh, S. Sastry, S. A. Seshia, and A. D. Dragan, "Planning for Autonomous Cars that Leverage Effects on Human Actions," in *Robotics: Science and Systems XII*. Robotics: Science and Systems Foundation, 2016.
- [14] C. Chen, Y. Liu, S. Kreiss, and A. Alahi, "Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning," in *International Conference on Robotics and Automation (ICRA)*, 2019, pp. 6015–6022. [Online]. Available: <doi.org/10.1109/ICRA.2019.8794134>
- [15] C. Chen, S. Hu, P. Nikdel, G. Mori, and M. Savva, "Relational Graph Learning for Crowd Navigation," in *2020 IEEE/RSSJ International Conference on Intelligent Robots and Systems (IROS)*. Las Vegas, NV, USA: IEEE, Oct. 2020, pp. 10 007–10 013. [Online]. Available: <https://ieeexplore.ieee.org/document/9340705/>
- [16] M. Everett, Y. F. Chen, and J. P. How, "Collision avoidance in pedestrian-rich environments with deep reinforcement learning," *IEEE Access*, vol. 9, pp. 10 357–10 377, 2021.
- [17] R. Cheng, G. Orosz, R. M. Murray, and J. W. Burdick, "End-to-End Safe Reinforcement Learning through Barrier Functions for Safety-Critical Continuous Control Tasks," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 3387–3395,

- July 2019. [Online]. Available: <https://ojs.aaai.org/index.php/AAAI/article/view/4213>
- [18] J. van den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-Body Collision Avoidance," in *Robotics Research*, B. Siciliano, O. Khatib, F. Groen, C. Pradalier, R. Siegwart, and G. Hirzinger, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, vol. 70, pp. 3–19, series Title: Springer Tracts in Advanced Robotics. [Online]. Available: [http://link.springer.com/10.1007/978-3-642-19457-3\\_1](http://link.springer.com/10.1007/978-3-642-19457-3_1)
- [19] Y. Liu, Q. Yan, and A. Alahi, "Social NCE: Contrastive Learning of Socially-Aware Motion Representations," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, Montreal, QC, Canada, Oct. 2021, pp. 15 118–15 129. [Online]. Available: <https://doi.org/10.1109/ICCV48922.2021.01484>
- [20] K. D. Katyal, G. D. Hager, and C. M. Huang, "Intent-Aware Pedestrian Prediction for Adaptive Crowd Navigation," in *International Conference on Robotics and Automation*, 2020, pp. 3277–3283, iSSN: 10504729. [Online]. Available: [doi.org/10.1109/ICRA40945.2020.9197434](https://doi.org/10.1109/ICRA40945.2020.9197434)
- [21] Y. Chen, F. Zhao, and Y. Lou, "Interactive Model Predictive Control for Robot Navigation in Dense Crowds," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 4, pp. 2289–2301, Apr. 2022. [Online]. Available: <https://ieeexplore.ieee.org/document/9328889/>
- [22] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social LSTM: Human Trajectory Prediction in Crowded Spaces," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [23] B. Ivanovic, E. Schmerling, K. Leung, and M. Pavone, "Generative Modeling of Multimodal Multi-Human Behavior," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Madrid: IEEE, Oct. 2018, pp. 3088–3095. [Online]. Available: <https://ieeexplore.ieee.org/document/8594393/>
- [24] A. D. Dragan, "Robot planning with mathematical models of human state and action," 2017. [Online]. Available: <https://arxiv.org/abs/1705.04226>
- [25] A. Lerner, Y. Chrysanthou, and D. Lischinski, "Crowds by Example," *Computer Graphics Forum*, vol. 26, no. 3, pp. 655–664, Sept. 2007. [Online]. Available: <https://onlinelibrary.wiley.com/doi/10.1111/j.1467-8659.2007.01089.x>
- [26] S. Pellegrini, A. Ess, K. Schindler, and L. van Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," in *2009 IEEE 12th International Conference on Computer Vision*. Kyoto: IEEE, Sept. 2009, pp. 261–268. [Online]. Available: <http://ieeexplore.ieee.org/document/5459260/>
- [27] R. Kummerle, M. Ruhnke, B. Steder, C. Stachniss, and W. Burgard, "A navigation system for robots operating in crowded urban environments," in *International Conference on Robotics and Automation (ICRA)*, 2013, pp. 3225–3232, iSSN: 10504729. [Online]. Available: [doi.org/10.1109/ICRA.2013.6631026](https://doi.org/10.1109/ICRA.2013.6631026)
- [28] H. Kretzschmar, M. Spies, C. Sprunk, and W. Burgard, "Socially compliant mobile robot navigation via inverse reinforcement learning," *International Journal of Robotics Research*, vol. 35, no. 11, pp. 1352–1370, 2016. [Online]. Available: [doi.org/10.1177/0278364915619772](https://doi.org/10.1177/0278364915619772)
- [29] H. Thomas, M. G. De Saint Aurin, J. Zhang, and T. D. Barfoot, "Learning Spatiotemporal Occupancy Grid Maps for Lifelong Navigation in Dynamic Scenes," in *2022 International Conference on Robotics and Automation (ICRA)*. Philadelphia, PA, USA: IEEE, May 2022, pp. 484–490. [Online]. Available: <https://ieeexplore.ieee.org/document/9812297/>
- [30] Y. Ren, S. Elliott, Y. Wang, Y. Yang, and W. Zhang, "How Shall I Drive? Interaction Modeling and Motion Planning towards Empathetic and Socially-Graceful Driving," in *IEEE International Conference on Robotics and Automation*, 2019, pp. 4325–4331, arXiv: 1901.10013. [Online]. Available: <http://arxiv.org/abs/1901.10013>
- [31] J. F. Fisac, E. Bronstein, E. Stefansson, D. Sadigh, S. S. Sastry, and A. D. Dragan, "Hierarchical Game-Theoretic Planning for Autonomous Vehicles," *IEEE International Conference on Robotics and Automation*, pp. 9590–9596, 2019, arXiv: 1810.05766 ISBN: 9781538660263. [Online]. Available: <http://arxiv.org/abs/1810.05766>
- [32] K. Kedia, P. Dan, and S. Choudhury, "A Game-Theoretic Framework for Joint Forecasting and Planning," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Detroit, MI, USA: IEEE, Oct. 2023, pp. 6773–6778. [Online]. Available: <https://ieeexplore.ieee.org/document/10341265/>
- [33] W. Schwarting, A. Pierson, J. Alonso-Mora, S. Karaman, and D. Rus, "Social behavior for autonomous vehicles," *Proceedings of the National Academy of Sciences (PNAS)*, pp. 1–7, 2019. [Online]. Available: <http://www.pnas.org/lookup/doi/10.1073/pnas.1820676116>
- [34] P. Fiorini and Z. Shiller, "Motion Planning in Dynamic Environments Using Velocity Obstacles," *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, July 1998. [Online]. Available: <http://journals.sagepub.com/doi/10.1177/027836499801700706>
- [35] J. B. Rawlings, D. Q. Mayne, and M. M. Diehl, *Model Predictive Control: Theory, Computation, and Design*, 2nd ed. Nob Hill Publishing, LLC, Feb. 2019.
- [36] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004.
- [37] S. Dempe and J. Dutta, "Is bilevel programming a special case of a mathematical program with complementarity constraints?" *Mathematical Programming*, vol. 131, no. 1-2, pp. 37–48, Feb. 2012. [Online]. Available: <http://link.springer.com/10.1007/s10107-010-0342-1>
- [38] G. Wachsmuth, "On LICQ and the uniqueness of Lagrange multipliers," *Operations Research Letters*, vol. 41, no. 1, pp. 78–80, Jan. 2013. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0167637712001459>
- [39] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social GAN: Socially Acceptable Trajectories with Generative Adversarial Networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Salt Lake City, UT: IEEE, June 2018, pp. 2255–2264. [Online]. Available: <https://ieeexplore.ieee.org/document/8578338/>
- [40] L. Thiede and P. Brahma, "Analyzing the Variety Loss in the Context of Probabilistic Trajectory Prediction," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. Seoul, Korea (South): IEEE, Oct. 2019, pp. 9953–9962. [Online]. Available: <https://ieeexplore.ieee.org/document/9010632/>
- [41] D. Helbing and P. Molnár, "Social force model for pedestrian dynamics," *Physical Review E*, vol. 51, no. 5, pp. 4282–4286, May 1995, helbing1995sfm. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevE.51.4282>
- [42] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.
- [43] A. Hill, A. Raffin, M. Ernestus, A. Gleave, A. Kanervisto, R. Traore, P. Dhariwal, C. Hesse, O. Klimov, A. Nichol, M. Plappert, A. Radford, J. Schulman, S. Sidor, and Y. Wu, "Stable baselines," <https://github.com/hill-a/stable-baselines>, 2018.
- [44] R. Verschueren, G. Frison, D. Kouzoupis, J. Frey, N. van Duijkeren, A. Zanelli, B. Novoselnik, T. Albin, R. Quirynen, and M. Diehl, "acados – a modular open-source framework for fast embedded optimal control," *Mathematical Programming Computation*, Oct 2021. [Online]. Available: <https://doi.org/10.1007/s12532-021-00208-8>
- [45] Y. Beck and M. Schmidt, "A robust approach for modeling limited observability in bilevel optimization," *Operations Research Letters*, vol. 49, no. 5, pp. 752–758, Sept. 2021. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0167637721001188>
- [46] K. Xie, T. Wang, U. Iqbal, Y. Guo, S. Fidler, and F. Shkurti, "Physics-based Human Motion Estimation and Synthesis from Videos," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. Montreal, QC, Canada: IEEE, Oct. 2021, pp. 11 512–11 521. [Online]. Available: <https://ieeexplore.ieee.org/document/9711215/>
- [47] K. Burnett, S. Samavi, S. L. Waslander, T. D. Barfoot, and A. P. Schoellig, "aUToTrack : A Lightweight Object Detection and Tracking System for the SAE AutoDrive Challenge," in *Conference on Computer and Robot Vision (CRV)*, 2019. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/8781613>



**Sepehr Samavi** is a Ph.D. Candidate at the University of Toronto Institute for Aerospace Studies, Canada. He received the B.A.Sc. in Engineering Science and M.A.Sc from the University of Toronto in 2018 and 2021. His research interests lie at the intersection of control, optimization, and machine learning, for robotics. His goal is to empower safe navigation in unstructured, dynamic, and human-centered environments. His research has been supported by the Alexander Graham Bell Canada Graduate Scholarship (CGSD) by the Natural Sciences and Engineering Research Council of Canada (NSERC).



**James R. Han** is a M.A.Sc student at the University of Toronto Institute for Aerospace Studies, Canada. He earned his B.A.Sc. in Engineering Science with a focus on Machine Intelligence and Robotics from the University of Toronto in 2023. His academic pursuits are centered at the intersection of Reinforcement Learning, Deep Learning, and Robotics, aiming to bring about real-world robots. This work was conducted during his undergraduate studies at the University of Toronto.



**Florian Shkurti** is an Assistant Professor in the Robotics Group at the Department of Computer Science, University of Toronto, Canada. He directs the Robot Vision and Learning (RVL) lab. His research centers around robotics and spans machine learning, perception, planning and control, with the aim of developing methods that enable robots to perceive, reason, and act effectively and safely, particularly in dynamic environments and alongside humans. He received his PhD in 2018 from McGill University.



**Angela P. Schoellig** is an Alexander von Humboldt Professor for Robotics and Artificial Intelligence at the Technical University of Munich, Germany. She is also an Associate Professor at the University of Toronto Institute for Aerospace Studies and a Faculty Member of the Vector Institute in Toronto, Canada. Angela conducts research at the intersection of robotics, controls, and machine learning with the goal of enhancing the performance, safety, and autonomy of robots by enabling them to learn from experience. She received her Dr. sc. (Ph.D.) from

ETH Zurich in 2013, where she was awarded the ETH Medal and the Dimitris N. Chorafas Foundation Award.



TABLE IV: Shapiro-Wilk Test for Normality across Metrics and Algorithms tested in ORCA simulations. Arrows indicate desired values ( $\uparrow$  indicates desired higher p-value toward rejecting  $H_0$ , i.e. accepting normality).

Approach	Navigation Time p-value $\uparrow$	Collision Frequency p-value $\uparrow$	Frozen Frequency p-value $\uparrow$
$N = 3$ humans			
SICNav-p	$5.59e - 43$	$2.35e - 42$	$3.25e - 41$
SICNav-np	$3.53e - 31$	$1.02e - 40$	$1.53e - 39$
MPC-CVMM	$7.74e - 41$	$4.94e - 39$	$1.82e - 38$
ORCA	1.00	1.00	1.00
DWA	1.00	1.00	1.00
SARL	$2.57e - 33$	$1.16e - 39$	$6.38e - 33$
RGL	$1.57e - 40$	$4.07e - 38$	$1.09e - 39$
$N = 5$ humans			
SICNav-p	$1.44e - 32$	$4.20e - 40$	$4.22e - 38$
SICNav-np	$3.95e - 28$	$1.17e - 35$	$3.46e - 37$
MPC-CVMM	$4.52e - 28$	$5.49e - 31$	$2.26e - 31$
ORCA	1.00	1.00	1.00
DWA	1.00	1.00	1.00
SARL	$2.21e - 41$	$1.96e - 33$	$5.46e - 41$
RGL	$6.97e - 31$	$1.08e - 34$	$3.64e - 30$

## APPENDIX

### A. Statistical Analysis of Simulation Results

In this section, we perform a series of statistical tests to verify that the crowd navigation simulation experiments presented in Section VII-A show statistically significant improvements in performance by using our proposed method.

1) *Selection of Statistical Tests*: In order to determine whether to use parametric or non-parametric tests, we first perform Shapiro-Wilk tests to check for normality of the data for each metric (Navigation Time, Collision Frequency, and Frozen Frequency) that is measured over the 500 test scenarios for each algorithm (SICNav, SICNav-np, MPC-CVMM, ORCA, SARL, RGL, DWA) in each environment ( $N = 3$  humans,  $N = 5$  humans). We conduct a total of 42 Shapiro-Wilk tests ( $3$  metrics  $\times$   $7$  algorithms  $\times$   $2$  environments). The hypotheses in each test are,

- $H_0$ : The performance metrics normally distributed.
- $H_1$ : The data is not normally distributed.

We use a significance level of  $\alpha = 0.05$  for all tests.

The results of the Shapiro-Wilk test are summarized in Tables IV and V. We observe that the p-values for the Shapiro-Wilk tests are all far below  $\alpha$  for all algorithms and metrics, except for all metrics in ORCA and DWA in the ORCA simulation environments. Since these results indicate that the performance results are not normally distributed for many algorithms, we need to use non-parametric tests.

2) *Mann-Whitney U Test for Pair-Wise Comparisons*: We use the Mann-Whitney U test to evaluate the significance of the differences observed in the performance metrics evaluated on the 500 evaluation scenarios across pairs of algorithms in each environment.

We begin by comparing the two variants of our method, SICNav and SICNav-np. We conduct a total of 6 Mann-Whitney U tests ( $3$  metric  $\times$   $2$  environments) for this pair of algorithms. The hypotheses in each test are,

TABLE V: Shapiro-Wilk Test for Normality across Metrics and Algorithms tested in SFM simulations. Arrows indicate desired values ( $\uparrow$  indicates desired higher p-value toward rejecting  $H_0$ , i.e. accepting normality).

Approach	Navigation Time p-value $\uparrow$	Collision Frequency p-value $\uparrow$	Frozen Frequency p-value $\uparrow$
$N = 3$ humans			
SICNav-p	$9.30e - 43$	$2.08e - 34$	$4.42e - 41$
SICNav-np	$1.31e - 41$	$4.00e - 35$	$1.37e - 41$
MPC-CVMM	$4.28e - 42$	$9.83e - 37$	$4.27e - 40$
ORCA	$3.04e - 18$	$7.35e - 38$	$7.60e - 37$
DWA	$6.71e - 33$	$1.03e - 43$	$5.72e - 33$
SARL	$1.11e - 31$	$4.97e - 40$	$8.27e - 30$
RGL	$5.14e - 41$	$1.13e - 38$	$2.98e - 40$
$N = 5$ humans			
SICNav-p	$1.31e - 29$	$4.79e - 36$	$2.05e - 37$
SICNav-np	$1.94e - 28$	$2.90e - 35$	$2.88e - 38$
MPC-CVMM	$2.62e - 41$	$1.15e - 32$	$4.39e - 39$
ORCA	$1.87e - 35$	$1.70e - 31$	$2.58e - 38$
DWA	$2.42e - 30$	$5.85e - 43$	$3.05e - 30$
SARL	$2.14e - 39$	$1.62e - 37$	$5.83e - 39$
RGL	$2.77e - 36$	$9.66e - 41$	$2.24e - 35$

TABLE VI: Pairwise Mann-Whitney U Test Results between SICNav-p and SICNav-np tested in ORCA simulations. Arrows indicate the desired outcome toward failing to reject the null hypothesis, signifying no significant differences between SICNav-p and SICNav-np.

Metric	$N = 3$ humans		$N = 5$ humans	
	Stat. $\uparrow$	p-value $\uparrow$	Stat. $\uparrow$	p-value $\uparrow$
Nav. Time	121221.5	0.402296	119372.5	0.216947
Coll. Freq.	120554.0	0.009947	110152.0	$1.30e - 07$
Frozen Freq.	124767.0	0.919510	123475.0	0.609077

- $H_0$ : The distributions of the performance metrics are identical between SICNav and SICNav-np.
- $H_1$ : The distributions of the performance metrics are not identical between SICNav and SICNav-np.

As before, we use a significance level of  $\alpha = 0.05$ . Tables VI and VII summarize the results of the Mann-Whitney U tests comparing SICNav-p and SICNav-np in the ORCA and SFM simulations, respectively. We observe that the test statistics for all metrics and environments are high,

From the pairwise tests in the SFM environment, we observe from the results that for navigation time and frozen frequency, the p-values exceed the significance level. This indicates no significant difference in the distributions of these performance metrics between SICNav-p and SICNav-np in both environments (i.e. failure to reject  $H_0$ ). Conversely, for collision frequency, the observed p-values ( $0.009947$  in the  $N = 3$  humans environment and  $1.30e - 07$  in the  $N = 5$  humans environment) fall below the significance threshold. This suggests that the distributions of collision frequencies between SICNav-p and SICNav-np are significantly different (i.e. reject  $H_0$ ).

From the pairwise tests in the SFM environment, we also observe that for navigation time and frozen frequency, the p-values ( $0.104038$  and  $0.422128$  for  $N = 3$  humans;  $0.065717$  and  $0.013285$  for  $N = 5$  humans) generally exceed the significance level, except for the frozen frequency in the

TABLE VII: Pairwise Mann-Whitney U Test Results between SICNav-p and SICNav-np tested in SFM simulations. Arrows indicate the desired outcome toward failing to reject the null hypothesis, signifying no significant differences between SICNav-p and SICNav-np.

Metric	$N = 3$ humans		$N = 5$ humans	
	Stat. $\uparrow$	p-value $\uparrow$	Stat. $\uparrow$	p-value $\uparrow$
Nav. Time	117642.5	0.104038	133368.0	0.065717
Coll. Freq.	126088.5	0.741520	128207.0	0.380315
Frozen Freq.	126975.5	0.422128	132179.0	0.013285

$N = 5$  humans scenario, indicating no significant difference in these performance metrics between SICNav-p and SICNav-np in most cases. In contrast, for collision frequency, the p-values (0.741520 for  $N = 3$  humans and 0.380315 for  $N = 5$  humans) exceed the significance level in the SFM environment, further indicating no significant difference for collision frequency in this environment.

These tests provide compelling evidence that the absence of privileged information in SICNav-np does not significantly affect the navigation time and frozen frequency in both ORCA and SFM environments, but does significantly affect the collision frequency in the ORCA environment. However, the average difference in collision frequency between SICNav-p and SICNav-np in the ORCA environment are small (see Table II). Nonetheless, for the sake of completeness, we proceed to compare both SICNav-p and SICNav-np against each of the baseline algorithms.

For each of the two variants of our proposed method, SICNav-p and SICNav-np, we conduct a total of 30 Mann-Whitney U tests (3 metrics  $\times$  5 baseline algorithms  $\times$  2 environments). The hypotheses in each test are,

- $H_0$ : The distributions of the performance metrics are identical between our proposed algorithm and the baseline algorithm.
- $H_1$ : The distributions of the performance metrics are not identical between our proposed algorithm and the baseline algorithm.

As before, we use a significance level of  $\alpha = 0.05$ .

Tables VIII and IX summarize the results of the Mann-Whitney U test for the comparison of SICNav-p against each of the baseline algorithms for the ORCA simulation and SFM simulation, respectively. We observe that almost all the p-values for the Mann-Whitney U test are low for all metrics and navigation scenarios, indicating that there are significant differences between SICNav-p and each of the baseline algorithms. The only exceptions are the frozen frequency metric in the  $N = 5$  humans environment for the comparison of SICNav-p against SARL in both the ORCA and SFM simulations, where the p-value exceeds  $\alpha = 0.05$ . This indicates that there are no significant differences between SICNav-p and SARL for the frozen frequency metric in the  $N = 5$  humans environment in these simulations.

Tables X and XI summarize the results of the Mann-Whitney U test for the comparison of SICNav-np against each of the baseline algorithms for the ORCA simulation and SFM simulation, respectively. We can make the same observations

TABLE VIII: Pairwise Mann-Whitney U Test Results of SICNav-p against Baseline Algorithms tested in ORCA simulations. Arrows in the table ( $\uparrow$  for Stat. and  $\downarrow$  for p-value) represent our analytical desire to uncover significant differences between SICNav-p and each baseline, indicating SICNav-p's distinct performance advantages.

Metric	$N = 3$ humans		$N = 5$ humans	
	Stat. $\uparrow$	p-value $\downarrow$	Stat. $\uparrow$	p-value $\downarrow$
SICNav-p vs. MPC-CVMM				
Nav. Time	85827.0	$4.06e - 18$	108493.0	0.000292915
Coll. Freq.	111530.0	$1.31e - 09$	85984.0	$2.58e - 30$
Frozen Freq.	110833.5	$4.07e - 07$	103446.0	$3.14e - 10$
SICNav-p vs. ORCA				
Nav. Time	40075.0	$3.48e - 77$	26466.5	$6.54e - 103$
Coll. Freq.	112631.5	$3.10e - 08$	100827.5	$1.50e - 14$
Frozen Freq.	84343.0	$3.06e - 31$	79141.0	$4.70e - 32$
SICNav-p vs. DWA				
Nav. Time	4656.0	$5.80e - 155$	10950.5	$1.53e - 137$
Coll. Freq.	115431.5	$3.88e - 06$	71260.5	$5.47e - 46$
Frozen Freq.	24000.0	$1.03e - 127$	18861.5	$4.25e - 132$
SICNav-p vs. SARL				
Nav. Time	164736.0	$1.42e - 18$	215377.0	$5.50e - 88$
Coll. Freq.	109332.5	$1.54e - 11$	93854.5	$4.88e - 22$
Frozen Freq.	95514.0	$1.43e - 20$	128810.5	0.176
SICNav-p vs. RGL				
Nav. Time	179211.5	$4.60e - 33$	107574.0	0.000119
Coll. Freq.	109442.0	$1.71e - 11$	75747.5	$6.22e - 41$
Frozen Freq.	116993.5	0.002	66160.0	$4.91e - 51$

TABLE IX: Pairwise Mann-Whitney U Test Results of SICNav-p against Baseline Algorithms tested in SFM simulations. Arrows in the table ( $\uparrow$  for Stat. and  $\downarrow$  for p-value) represent our analytical desire to uncover significant differences between SICNav-p and each baseline, indicating SICNav-p's distinct performance advantages.

Metric	$N = 3$ humans		$N = 5$ humans	
	Stat. $\uparrow$	p-value $\downarrow$	Stat. $\uparrow$	p-value $\downarrow$
SICNav-p vs. MPC-CVMM				
Nav. Time	100911.5	$1.10e - 07$	129735.0	0.298018
Coll. Freq.	123682.5	0.691238	113669.5	0.003175
Frozen Freq.	112400.0	$2.06e - 05$	117566.5	0.022995
SICNav-p vs. ORCA				
Nav. Time	59444.5	$4.82e - 47$	58916.0	$1.39e - 47$
Coll. Freq.	136718.5	0.000103	120294.5	0.216113
Frozen Freq.	120162.0	0.077268	111429.5	0.000104
SICNav-p vs. DWA				
Nav. Time	6776.5	$3.91e - 148$	12364.5	$1.80e - 134$
Coll. Freq.	145570.0	$1.16e - 13$	153497.5	$1.10e - 19$
Frozen Freq.	72485.5	$9.05e - 44$	71533.0	$1.43e - 40$
SICNav-p vs. SARL				
Nav. Time	140532.5	0.000622	221024.0	$6.51e - 100$
Coll. Freq.	128256.0	0.312171	137374.0	0.000310
Frozen Freq.	73315.5	$2.10e - 44$	127930.0	0.323385
SICNav-p vs. RGL				
Nav. Time	221622.0	$1.76e - 100$	37619.5	$7.47e - 88$
Coll. Freq.	135752.5	0.000300	144991.0	$4.08e - 09$
Frozen Freq.	120759.0	0.112056	20241.5	$2.21e - 130$

as for SICNav-p, where almost all the p-values for the Mann-Whitney U test are low for all metrics and navigation scenarios, indicating that there are significant differences between SICNav-np and each of the baseline algorithms. The only exceptions are the frozen frequency metric in the  $N = 5$  humans environment in the ORCA simulation and the collision frequency metric in the  $N = 3$  humans environment in the SFM simulation for the comparison of SICNav-np against SARL, where the p-value exceeds  $\alpha = 0.05$ . This indicates

TABLE X: Pairwise Mann-Whitney U Test Results of SICNav-np against Baseline Algorithms in ORCA simulations. Arrows in the table ( $\uparrow$  for Stat. and  $\downarrow$  for p-value) represent our analytical desire to uncover significant differences between SICNav-np and each baseline, indicating SICNav-np’s distinct performance advantages.

Metric	$N = 3$ humans		$N = 5$ humans	
	Stat. $\uparrow$	p-value $\downarrow$	Stat. $\uparrow$	p-value $\downarrow$
SICNav-np vs. MPC-CVMM				
Nav. Time	89875.0	$7.78e - 15$	112744.5	0.007183724
Coll. Freq.	115875.0	0.000162293	97726.0	$1.62e - 13$
Frozen Freq.	111069.0	$6.80e - 07$	104256.5	$2.08e - 09$
SICNav-np vs. ORCA				
Nav. Time	41541.5	$1.55e - 74$	25943.5	$5.60e - 104$
Coll. Freq.	117143.5	0.001753694	116393.0	0.018405429
Frozen Freq.	84537.0	$7.19e - 31$	79896.5	$1.10e - 30$
SICNav-np vs. DWA				
Nav. Time	5706.5	$3.43e - 152$	10793.5	$6.73e - 138$
Coll. Freq.	120028.0	0.035176261	88519.0	$1.51e - 19$
Frozen Freq.	24003.5	$1.42e - 127$	18627.0	$5.13e - 132$
SICNav-np vs. SARL				
Nav. Time	164838.5	$1.19e - 18$	216830.0	$9.26e - 91$
Coll. Freq.	113773.0	$7.73e - 06$	105817.0	$6.84e - 08$
Frozen Freq.	95694.5	$2.90e - 20$	130540.0	0.053446485
SICNav-np vs. RGL				
Nav. Time	180210.5	$3.35e - 34$	108649.0	0.000304544
Coll. Freq.	113542.0	$4.59e - 06$	94052.5	$2.32e - 15$
Frozen Freq.	117233.5	0.002776	67023.0	$4.91e - 51$

TABLE XI: Pairwise Mann-Whitney U Test Results of SICNav-np against Baseline Algorithms in SFM simulations. Arrows in the table ( $\uparrow$  for Stat. and  $\downarrow$  for p-value) represent our analytical desire to uncover significant differences between SICNav-np and each baseline, indicating SICNav-np’s distinct performance advantages.

Metric	$N = 3$ humans		$N = 5$ humans	
	Stat. $\uparrow$	p-value $\downarrow$	Stat. $\uparrow$	p-value $\downarrow$
SICNav-np vs. MPC-CVMM				
Nav. Time	105022.0	$1.55e - 08$	128497.0	0.223828
Coll. Freq.	130855.5	0.381082	125318.0	0.121516
Frozen Freq.	119577.0	$1.40e - 05$	116662.5	0.014878
SICNav-np vs. ORCA				
Nav. Time	57897.0	$2.42e - 38$	66466.5	$2.15e - 37$
Coll. Freq.	108882.0	$4.65e - 09$	141355.5	0.274905
Frozen Freq.	144661.0	0.960146	127993.5	0.011287
SICNav-np vs. DWA				
Nav. Time	13089.0	$1.79e - 104$	12064.0	$6.03e - 89$
Coll. Freq.	137814.0	0.724094	122684.0	0.015292
Frozen Freq.	84822.0	$1.63e - 80$	79184.0	$8.21e - 63$
SICNav-np vs. SARL				
Nav. Time	164449.5	$5.09e - 25$	212318.0	$2.07e - 66$
Coll. Freq.	127842.5	0.148963	112235.5	$3.13e - 05$
Frozen Freq.	128716.5	0.235351	129313.5	0.073007
SICNav-np vs. RGL				
Nav. Time	198035.0	$1.45e - 46$	101080.0	$9.73e - 06$
Coll. Freq.	126678.0	0.107036	146464.0	0.946688
Frozen Freq.	107908.5	$1.78e - 10$	88899.0	$8.11e - 63$

that there are no significant differences between SICNav-np and SARL for these metrics and environments.

3) *Discussion and Conclusion:* In this section we performed a series of statistical tests to verify the significance of the crowd navigation performance results that we present in Section VII-A. In the first series of tests, we compared the two variants of our algorithm, SICNav-p and SICNav-np, with each other. We found that the absence of privileged information in SICNav-np does not significantly affect most metrics. This

finding led us to compare each variant of our method with each of the baseline algorithms in pairwise fashion. Combining our findings from these pairwise statistical comparisons with the average values of metrics presented in Section VII-A, we show that for all metrics and navigation scenarios both variants of our method significantly outperformed all the baselines. The only exception was the frozen frequency metric in the  $N = 5$  humans environment for the comparisons of the variants of our method with SARL. Our analysis provides compelling evidence that the proposed method is significantly better than the baseline algorithms for crowd navigation in both the  $N = 3$  humans and  $N = 5$  humans environments for both ORCA and SFM simulators.