Abstract—We focus on the problem of planning safe and efficient motion for a ballbot (i.e., a dynamically balancing mobile robot), navigating in a crowded environment. The ballbot’s design gives rise to human-readable motion which is valuable for crowd navigation. However, dynamic stabilization introduces kinematic constraints that severely limit the ability of the robot to execute aggressive maneuvers, complicating collision avoidance and respect for human personal space. Past works reduce the need for aggressive maneuvering by motivating anticipatory collision avoidance through the use of human motion prediction models. However, multiagent behavior prediction is hard due to the combinatorial structure of the space. Our key insight is that we can accomplish anticipatory multiagent collision avoidance without high-fidelity prediction models if we capture fundamental features of multiagent dynamics. To this end, we build a model predictive control architecture that employs a constant-velocity model of human motion prediction but monitors and proactively adapts to the unfolding homotopy class of crowd-robot dynamics by taking actions that maximize the pairwise winding numbers between the robot and each human agent. This results in robot motion that accomplishes statistically significantly higher clearances from the crowd compared to state-of-the-art baselines while maintaining similar levels of efficiency, across a variety of challenging physical scenarios and crowd simulators.

I. INTRODUCTION

Readable robot motion may reduce the mental load and planning effort for co-navigating humans [5]. This has motivated work on legible motion for mobile robots [20, 22, 34]. However, generating human-readable motion with just 2 or 3 degrees of freedom can be challenging. Humans leverage a variety of modalities to signal intent in navigation, such as eye gaze, gestures or body posture [12, 13, 41]. Inspired by humans, roboticists have incorporated modalities such as blinking lights [1, 11], robot arm signals [15] or body leaning [10, 27] into robot motion design.

In this paper, we focus on the special case of a robot design with the ability of body leaning: the ballbot. A ballbot [26] is a mobile robot that dynamically balances on a single spherical wheel joint (see Fig. 1). The ballbot’s self-balancing dynamics enables human-readable [29], dynamically agile maneuvers, especially valuable for operation in crowded human environments. This type of dynamics is also amenable to contact with humans [35], allowing for fast, smooth, and compliant yielding when accidental contact occurs.

Besides these important benefits, the ballbot design also introduces special challenges. The requirement for dynamic stabilization severely limits the ability of the ballbot to execute agile and aggressive planar maneuvers. On statically stable robot designs, such maneuvers are often important for respecting the human personal space and avoiding imminent collisions in situations involving complex crowd dynamics in confined spaces. Existing state-of-the-art frameworks [6, 7, 9] often rely on such maneuvers to ensure human safety.

To enable a ballbot to safely maneuver around humans, we design a framework for anticipatory collision avoidance. Anticipatory motion is generally important for any robot but especially crucial for constrained platforms like the ballbot. Generating anticipatory maneuvers involves incorporating models of human motion prediction into decision making [2, 39, 43]. However, multiagent behavior prediction is fundamentally hard due to the combinatorial structure of the underlying space [8], and a topic of ongoing research [38]. Further, human response to robot motion is not well understood, and driven by novelty effects, limited human mental models, and context-dependency.

Fig. 1: Still from our experiments. Honda’s experimental ballbot [17] navigates next to three users in a lab workspace.
Our key insight is that we can produce anticipatory collision-avoidance maneuvers even without high-fidelity human motion prediction if we integrate fundamental properties of crowd dynamics into the robot’s decision making strategy. To this end, we employ a simplistic constant-velocity human motion prediction model but leverage a topological representation that abstracts crowd-robot dynamics over a horizon into a tuple of topological invariants (winding numbers), modeling pairwise interactions between the robot and human agents (Fig. 2). We then design a cost functional that drives robot motion towards the unfolding homotopy class of crowd-robot dynamics. We incorporate this cost and the constant-velocity human motion prediction model into a model predictive control (MPC) architecture (Fig. 3).

We evaluated the ability of our framework to generate safe ballbot motion in simulated and real crowded domains. Our simulated investigation considered challenging multiagent scenarios involving human agents simulated using multiple crowd simulation engines. Our framework exhibited statistically significantly safer behaviors than a state-of-the-art baseline while maintaining similar efficiency. A deployment of our framework on a real ballbot navigating in a tight workspace next to 3 users demonstrated real-world transfer of attributes observed in simulation. Snippets from our experiments can be found at https://youtu.be/Y-YGhcOxYFU.

II. RELATED WORK

Recent work in crowd navigation leverages the expectation of human cooperation and rationality [32]. For instance, Trautman et al. [39] developed a Gaussian-process-based human motion prediction model that estimates future human trajectories under the assumption of cooperative and goal-directed human behavior. With a similar goal, Ziebart et al. [43] employed inverse reinforcement learning (IRL) as a method for estimating humans’ goal-directed motion towards planning robot motion that does not distract human paths, whereas Kretzschmar et al. [24] and Kim and Pineau [19] used IRL as a technique to recover robot policies for socially compliant navigation. Inspired by this line of work, our approach formalizes cooperation as a low-dimensional abstraction of multiagent dynamics which is explicitly incorporated into a model predictive controller.

Some works leverage cooperation implicitly by using crowd simulators [16, 40] to recover policies for collision avoidance via deep reinforcement learning (RL) [6, 9, 28]. Such approaches tend to require large amounts of data. The high dimensionality of crowd navigation, and the challenge of acquiring realistic human-robot interaction data or realistically simulating human crowds limits their applicability and generalization to new domains. To address such complications, recent works have been combining data-driven methods with model based techniques. For instance, Brito et al. [3] employ an interaction-aware RL-based subgoal recommendation model to guide a MPC to smoothly guide the unfolding interactions with other agents, enabling informed decision making.

A. Topological Models of Multiagent Dynamics

Another thread employs representations from topology to model the multiagent dynamics of crowd navigation domains. Kretzschmar et al. [24] employ a topology-aware featurization in their IRL framework to incorporate preferences over passing sides. Cao et al. [4] employ elements of homotopy theory to enhance a global planner with an understanding of local multiagent dynamics. Mavrogiannis and Knepper [33] abstract multiagent dynamics into topological braids, and leverage vortex dynamics in a Hamiltonian form to generate multiagent motion primitives [30].

Our approach is closest to recent work that abstracts multiagent collision avoidance as a superposition of rotations [34, 37] but moves beyond in a few important ways. Instead of angular momentum [34], we make use of the winding number, a topological invariant that encodes global properties of motion. This enables the transition to a MPC architecture with long-horizon actions (i.e., trajectories), which
yields smoother behavior. While Roh et al. [37] also employ the winding number to define modes of intersection crossing, they use them at a binary level (right or left); in contrast, we also leverage the absolute value, which is indicative of progress in a pairwise collision avoidance maneuver.

B. Ballbots

First proposed by Lauwers et al. [26], a ballbot consists of a mechanical body that dynamically balances on top of an omnidirectional ball that makes single-point contact with the ground (see Fig. 1). A set of actuators enable the ball to move independently along all three directions of motion on the plane. A few works have proposed similar designs such as BallIP [25], Rezero [10], Kugle [18] or Honda’s experimental ballbot [17]. Going beyond the design, past work with ballbots has looked at trajectory planning and control [36], physical human-robot interaction [35], and human perceptions of pedestrian avoidance strategies [29].

However, not much attention has been placed to the problem of multiagent collision avoidance on a ballbot. The stabilization constraints effectively limit the repertoire of actions that a ballbot can execute. In safety-critical tasks, this could complicate collision avoidance and yield unsafe behaviors.

III. PROBLEM STATEMENT

Consider a workspace $W \subseteq \mathbb{R}^2$ where a robot navigates amongst $n$ other dynamic agents. Denote by $s$ the state of the robot and by $s^i \in W$, $i \in \mathcal{N} = \{1, \ldots, n\}$, the state of agent $i$. The robot is navigating from a state $s_0$ towards a destination $s_T$ by executing controls $u$ from a space of controls $\mathcal{U}$, subject to dynamics constraints $\dot{s} = g(s,u)$. Agent $i \in \mathcal{N}$ is navigating from $s^i_0$ towards a destination $s^i_T$ by executing controls $u^i$ from a space of controls $\mathcal{U}^i$. The robot is not aware of agent $i$’s destinations $s^i_T$ or policy. However, we assume that the robot is perfectly observing the complete world state $(s, s^{1:n})$. In this paper, our goal is to design a policy $\pi(s, s^{1:n}) \rightarrow u$ that enables the robot to navigate from $s_0$ to $s_T$ safely and efficiently.

IV. A TOPOLOGICAL ABSTRACTION OF MULTIAGENT DYNAMICS

We describe an abstraction of multiagent dynamics that emphasizes pairwise collision-avoidance intentions.

A. A Topological Signature of Pairwise Collision Avoidance

Define by $x^i_k$ a vector that connects the robot’s position to the position of agent $i$ at timestep $k$, and by $\theta^i_k = \angle x^i_k$ the angle of that vector with respect to a fixed global frame. From time $k$ to $k+1$, agents’ displacement from $x^i_k$ to $x^i_{k+1}$ results in a rotation $\Delta \theta^i_{k+1} = \theta^i_{k+1} - \theta^i_k$. Over a horizon of $N$ timesteps, the relative accumulated rotation of the two agents generates a topological signature that can be captured by the winding number:

$$\lambda^i(s, s^i) = \frac{1}{2\pi} \sum_{k=0}^{N-1} \Delta \theta^i_{k+1}. \tag{1}$$

The magnitude of the winding number represents the number of times that the robot and agent $i$ revolved around each other throughout the period $N$, whereas its sign indicates the direction of this rotation, i.e., on which side the two agents passed each other. Right-side passing corresponds to a clockwise rotation yielding a positive winding number ($\lambda^i > 0$), whereas left-side passing corresponds to counterclockwise rotation yielding a negative winding number ($\lambda^i < 0$). Figs. 2a, 2b show examples of winding-number computations in a simple two-agent scenario, highlighting the property of topological invariance: for any pair of trajectories $(s, s^i)$ between the same start $(s_0, s^i_0)$ and goal locations $(s_T, s^i_T)$ for which passing sides between agents are the same, $\lambda^i(s, s^i)$ is constant.

B. Crowd-Robot Dynamics as a Tuple of Winding Numbers

In a scene where the robot navigates alongside $n$ other agents, we can abstract the unfolding multiagent dynamics into a tuple $\Lambda$ containing the winding numbers representing the “passing relationships” formed between the robot and all other agents:

$$\Lambda = (\lambda^1, \lambda^2, \ldots, \lambda^n). \tag{2}$$

By monitoring $\Lambda$, the robot may proactively adapt to the collision-avoidance intentions (intended passing sides) of humans, thus reducing the need for aggressive maneuvering. See Fig. 2c for an example of $\Lambda$ in a multiagent scene.

C. A Topology-Enforcing Cost Functional

Based on the definition of $\Lambda$, we develop a cost functional $J(s, s^{1:n}) \rightarrow \mathbb{R}$ that can be used to steer a planner/controller towards topology-informed decision making when considering an ego trajectory $s$ given an estimate of other agents’ trajectories $s^{1:n}$:

$$J_I(s, s^{1:n}) = -\frac{1}{n} \sum_{i=1}^{n} \lambda^i(s, s^i)^2. \tag{3}$$

Minimization of this cost corresponds to maximization of the absolute value of the winding numbers defined between the robot and all other agents. This promotes robot trajectories that attempt to align as much as possible with the unfolding collision-avoiding maneuvers of other agents towards accomplishing the goal of anticipatory collision avoidance (see Fig. 2c). The $J_I$ cost considers only non-stationary agents within the robot’s field of view to prevent disengaged agents from influencing the robot’s perception of multiagent dynamics. Finally, note that the robot can only explicitly decide on $s - s^{1:n}$ is decided by other agents.

V. TOPOLOGY-INFORMED MODEL PREDICTIVE CONTROL

We describe T-MPC, a model predictive controller for navigation in crowds that leverages our topological representation of multiagent dynamics described in Sec. IV.
A. Model Predictive Control for Navigation next to Humans

A MPC for navigation in a multiagent environment can be formulated as the following optimization problem:

\[
\mathbf{u}^* = \arg \min_{\mathbf{u}_{0:N-1}} \mathcal{J}(s_{1:N}, s^1_{1:N}, \ldots, s^n_{1:N})
\]

\[
\text{s.t. } s_{k+1} = g(s_k, u_k)
\]

\[
s^i_{k+1} = f(s_{k-h:k}, s^i_{k-h:k}),
\]

where the current system state \((s_0, s^1_0, \ldots, s^n_0)\) is known, \(\mathbf{u}^* = u_{0:N-1}\) is the optimal trajectory of robot controls over a horizon \(N\) with respect to a cost functional \(\mathcal{J}\), and \(f\) is a state transition model that takes as input the system state history up to \(h\) timesteps in the past.

A Vanilla implementation of MPC for navigation in human environments (V-MPC) encodes specifications such as safety and efficiency. We define \(\mathcal{J}\) as a weighted sum

\[
\mathcal{J}(s, s^{1:n}) = a_g \mathcal{J}_g(s) + a_d \mathcal{J}_d(s, s^{1:n}),
\]

where \(s = s_{1:N}, s^{1:n} = (s^1, \ldots, s^n)\) and \(s^i = s^i_{1:N}\). The term

\[
\mathcal{J}_g(s) = \sum_{k=0}^{N-1} (s_{k+1} - s_T)^T Q_g (s_{k+1} - s_T),
\]

is a goal-tracking cost penalizing trajectories taking the robot further from its goal, where \(Q_g\) is a weight matrix. The term

\[
\mathcal{J}_d(s, s^{1:n}) = \sum_{k=0}^{N-1} \sum_{i=1}^n A^2_d(s_k, s^i_{k+1}),
\]

is a cost penalizing violations to agents’ personal space \([14]\) through the Asymmetric Gaussian Integral Function \(A_2\) of Kirby \([21]\). The weights \(a_g, a_d\) encode the relative importance of cost terms. Finally, we approximate agents’ state transition in (4) by adopting a constant-velocity motion model of the form \(s^i_{k+1} = f(s^i_{k-h:k})\), and setting \(h = 1\). Via finite differencing, the model propagates the current state of agent \(i\) one timestep \(dt\) into the future, without accounting for interactions between agents’ behaviors.

B. T-MPC: Proactively Adapting to Multiagent Dynamics

We incorporate the functional of (3) into V-MPC (5) to derive the topology-informed MPC (T-MPC):

\[
\mathcal{J}(s, s^{1:n}) = a_g \mathcal{J}_g(s) + a_d \mathcal{J}_d(s, s^{1:n}) + a_t \mathcal{J}_t(s, s^{1:n}),
\]

where and \(a_t\) is a weight of relative significance. T-MPC accounts for aligning with the unfolding multiagent dynamics while respecting agents’ personal space and making progress towards its destination. In conjunction, this formulation is designed to motivate goal-oriented, anticipatory collision avoidance. Fig. 3 illustrates the T-MPC framework.

VI. EVALUATION

We evaluate T-MPC through a simulation study involving ballbot navigation in challenging crowded scenes. A. Ballbot model

We employ the model depicted in Fig. 4 following the design of the experimental ballbot by Honda \([17]\). We only consider the position of the ball on the plane and not its orientation. We define the robot state as \(s = (x_1, y_1, \dot{x}_1, \dot{y}_1, \theta_1^1, \theta_1^2)\), where \((x, y)\) is its position, \((\dot{x}, \dot{y})\) its velocity, \((\theta_1^1, \theta_1^2)\) its body inclination, \((\dot{\theta}_1^1, \dot{\theta}_1^2)\) the rate of change of its inclination. For body stabilization and velocity control, we employ the controller of Yamane and Kurosu \([42]\), which consists of a state feedback controller, augmented with an integral control layer to account for steady-state errors through due to e.g., floor conditions, robot design parameters etc. This controller induces closed-loop dynamics of the form \(s_{k+1} = g(s_k, u_k)\), where \(u_k\) represents a reference velocity control for the ballbot position at time \(k\). The controller will attempt to reach \(u\) by generating accelerations \((\dot{\theta}_1^2, \dot{\theta}_1^2)\) on the the ball.
B. Experimental Setup

We considered a rectangular workspace of area $3.6 \times 4.5m^2$. We partitioned the workspace into six zones of equal area $1.8 \times 1.5m^2$, and defined three scenarios involving 3, 4, and 5 humans respectively as shown in Fig. 5. In each scenario, humans move between start and goal zones, selected to give rise to challenging collision-avoidance instances. For each scenario, we generate 100 trials by sampling start and goal coordinates for agents, uniformly at random from their assigned zones. Across all trials, the robot’s start and goal coordinates are fixed at $(0, 0)$ and $(3.6, 4.5)$, respectively. The preferred speeds for both the robot and human agents are set to $0.8m/s$ which was found experimentally to be a natural walking speed for the dimensions of this workspace.

We evaluate T-MPC in terms of: a) Safety, defined as the minimum robot distance to human agents throughout a trial, $D(m)$; b) Efficiency, defined as the time taken by the robot to reach its goal, $T(s)$. We compare T-MPC’s performance against three main baseline robot policies:

CADRL [9] is a recent collision-avoidance framework based on deep reinforcement learning. The original formulation considers a differential-drive robot. To accommodate a ballbot, we modified their formulation as follows. We kept the original observation space, but augmented the state transition model. We initialized the network with the original weights from the implementation of [9] and included the ballbot dynamics defined in VI-A as part of the state transition model. We modified their formulation as follows. We kept the original weights from the implementation of [9] and subsequently trained it in two stages, first with 2-4 agents and then with 2-10 agents.

ORCA [40] is a standard baseline in crowd navigation literature. Under homogeneous settings (identical agents), ORCA guarantees collision avoidance for a finite time by modeling agents as obstacles of size depending on their velocity (velocity obstacles). We use the ORCA configuration of Chen et al. [6].

MPC: We solve a discretized version of (4); at every cycle, we evaluate $m$ rollouts $u_1, \ldots, u_m$ of horizon $N$. These rollouts start from the robot’s current state to a set of $m$ subgoals. For an in-depth investigation, we consider three types of rollouts: a) a constant-velocity propagation mechanism (CV); b) ORCA [40]; c) CADRL [9]. Thus, we instantiate 6 MPC variants: 1) T-MPC-CV; 2) T-MPC-ORCA; 3) T-MPC-CADRL; 4) V-MPC-CV; 5) V-MPC-ORCA; 6) V-MPC-CADRL.

All MPC variants consider $m = 10$ subgoals, spanning $[0, 2\pi)$ in $\pi/5$ intervals at a distance $8m$ from the robot. For each variant, we generate these rollouts by executing a policy $\pi_{sim} \in \{CV, ORCA, CADRL\}$ into the future for 10 timesteps of size $dt = 0.1s$. We conducted a parameter sweep for the weights $a_g$ and $a_d$ of the V-MPC cost functional, optimizing with respect to Safety and Efficiency over 30 trials for each scenario. T-MPC shares the same cost weights but also incorporates a weight $a_t$ which was acquired through a similar parameter sweep over Safety. The weights used throughout our evaluations are $a_g = 5$, $a_d = 1$, $a_t = 5$.

We conducted our evaluation in two Gazebo [23] worlds: one in which humans are simulated as ORCA agents, and one as CADRL agents. ORCA is a standard evaluation environment in crowd simulation and social navigation literature [6, 6, 28, 32]. CADRL world serves as a way to investigate the robustness of T-MPC. Across all simulations, humans are represented as spheres of $0.3m$ radius whereas the robot’s body is modeled as a cylinder of radius $0.2m$.

C. Hypotheses

We investigate the following hypotheses:

H1: MPC controllers with ORCA or CADRL rollouts outperform controllers with CV rollouts in terms of Safety across all scenarios and worlds.

H2: T-MPC outperforms a V-MPC with identical rollouts across all scenarios (3, 4, 5 agents) and worlds (ORCA, CADRL) in terms of Safety.
TABLE I: Performance of robot policies with respect to Safety (D) and Efficiency (T) across all worlds and scenarios. Each entry contains a mean and a standard deviation over 100 trials. Bold entries indicate the best-performing controller per column. Red, green and blue entries indicate scenarios in which a baseline was outperformed by the best-performing controller at a significance level corresponding to a p-value < 0.001, < 0.01, and < 0.05 respectively (U test).

| Scenario | Three Humans | Four Humans | Five Humans |
|----------|--------------|-------------|-------------|
| Metric   | D(m)         | T(s)        | D(m)        | T(s)        | D(m)        | T(s)        |
| ORCA     | 0.65 ± 0.11  | 0.71 ± 0.21 | 0.62 ± 0.13 | 0.69 ± 0.24 | 0.71 ± 0.29 | 0.72 ± 0.30 |
| CADRL    | 0.62 ± 0.10  | 0.69 ± 0.21 | 0.60 ± 0.12 | 0.67 ± 0.24 | 0.68 ± 0.21 | 0.69 ± 0.22 |

**H3:** T-MPC outperforms both CADRL and ORCA across all scenarios and worlds in terms of Safety.

**D. Analysis**

Table I contains the performance of all policies across all scenarios and worlds. Fig. 6 depicts the differences between the T-MPC and V-MPC variants in the form of % improvement over the world baseline (i.e., ORCA or CADRL) in terms of Safety. Fig. 7 highlights the performance of T-MPC compared to CADRL and ORCA in terms of % Safety improvement over each world’s baseline.

**H1** was supported. As we see in Fig. 6, informed rollouts (i.e., ORCA or CADRL), tend to enable a MPC (V-MPC or T-MPC) to perform better in terms of Safety, and in most cases to a statistically significant extent, regardless of scenario or world (Mann–Whitney U test).

**H2** was supported. In an ORCA world (Fig. 6a), T-MPC statistically significantly outperforms V-MPC with identical rollouts in terms of Safety (Mann–Whitney test). The result is more pronounced when using informed rollouts, i.e., CADRL or ORCA. In a CADRL world (Fig. 6a), we see the same pattern, with even more pronounced differences.

**H3** was supported. As shown in Table I, T-MPC variants with informed rollouts tend to perform best in terms of Safety to a statistically significant extent across almost all scenarios and worlds. Fig. 7 gives a deeper insight of this result: we see that in each world, the best performing T-MPC variant contributes a significantly higher Safety improvement compared to its baselines. Crucially, this result is even more pronounced in challenging scenarios with 4 or 5 humans.

In terms of efficiency, we see (Table I) that ORCA and CADRL tend to dominate, with the exception of the 5-humans scenario in a CADRL world, in which T-MPC-CV does best while maintaining a good Safety level. In fact, we see that T-MPC-CV often performs comparably to CADRL, see for example the 3-humans scenario in the ORCA world, or the 4-humans scenario in the ORCA and CADRL worlds.

**E. Discussion**

We saw that the selection of rollouts affects performance significantly (H1). While CV rollouts might enable a MPC to accomplish Efficiency comparable to the baselines, we saw that it cannot deliver the Safety of informed rollouts. Further, we saw that for identical rollouts rollouts, T-MPC consistently outperforms V-MPC across all scenarios and worlds in terms of Safety (H2). Crucially, we showed that T-MPC outperforms its two main baselines, ORCA and CADRL in terms of Safety (H3) and generally performs comparably in terms of Efficiency.

A holistic look at the results shows that the ORCA world is generally tighter to navigate: all controllers generally achieve lower clearances than in the CADRL world. We also see that the rollout-world match tends to make a difference; across all scenarios, T-MPC-ORCA dominates in an ORCA world and similarly, T-MPC-CADRL dominates in a CADRL world. Overall, our investigation demonstrated the robustness of our framework in handling different numbers and types of agents.

**VII. REAL-WORLD EXPERIMENTS**

We conducted a pilot study involving navigation of a ballbot alongside three members of our research team in a
Fig. 7: Scatter plots of % change in Safety over the world baseline. The plots compare the best-performing T-MPC for each world, i.e., T-MPC-ORCA in (a) and T-MPC-CADRL in (b) against CADRL and ORCA respectively.

lab workspace of size $4 \times 5m^2$. We used Honda’s experimental ballbot [17], which is 1 m tall, and weighs 20kg (Fig. 1). Agents were tracked using an Optitrack motion capture system of twelve overhead cameras operating at 120Hz. The system captured agents’ positions by tracking reflective markers placed on the top of the robot body and on construction hats that users wore. Users participated in three interactions; in each interaction, the robot navigated with a different policy in the following order: a) CV (the robot drives to the goal with constant velocity, without avoiding collisions), b) CADRL, c) T-MPC-CADRL. CV is a low-performance reference whereas CADRL has been shown to perform robustly [9] in the real world. For the same reason, we also employed the T-MPC-CADRL variant. Each interaction consisted of 20 trials. In each trial, users navigated between opposing corners of the workspace in the formation of Fig. 5a. They were told to navigate with normal walking speed and to treat the robot as a walking human.

Fig. 1 shows a still from our experiments whereas Fig. 9 depicts paths followed by the robot and the users under the 3 robot policies. Fig. 8 depicts the cumulative performance of the robot policies across all interactions. We see that the trends of simulation transfer to the real world: T-MPC-CADRL statistically significantly outperforms CV and CADRL in terms of Safety, while exhibiting similar Efficiency to CADRL. We observed a tendency of T-MPC-CADRL to keep greater clearances from users by proactively deviating from its direction to goal, as shown in Fig. 9c, and as confirmed in Fig. 8a. After the study, users informally confirmed that the behavior differences between policies were noticeable: they found CV to be the least preferred and commended the expressiveness of T-MPC-CADRL but also the predictability of CADRL. Snippets from our experiments can be found at https://youtu.be/Y-YGhcOxYFU.

VIII. DISCUSSION

Our evaluation focused on Safety due to the safety-critical nature of the domain. However, higher-order properties of robot motion such as smoothness or acceleration are also known to influence users’ perceptions [31] and it would be important to account for them. Overall, it would be important to collect users’ feedback, perceptions and preferences. To this end, we plan on conducting a more extensive study, building up on past work on benchmarking in social navigation [31]. Finally, while a simple constant-velocity human motion prediction model appeared to be sufficient for a T-MPC in the considered domains, we would be interested to understand how it would compare against a baseline featuring a more involved trajectory prediction model [38]. Relatedly, another direction of future work involves investigating alternative subgoal generation models [3] for facilitating goal-directed motion.
