Multi-Agent Path Topology in Support of Socially Competent Navigation Planning

Christoforos I. Mavrogiannis\(^1\) and Ross A. Knepper\(^2\)

Abstract
We present a navigation planning framework for dynamic, multi-agent environments, where no explicit communication takes place among agents. Inspired by the collaborative nature of human navigation, our approach encodes the concept of coordination into an agent’s decision making through an inference mechanism about collaborative strategies of collision avoidance. Each such strategy represents a distinct avoidance protocol, prescribing a distinct class of navigation behaviors to agents. We model such classes as equivalence classes of multi-agent path topology, using the formalism of topological braids. This formalism may naturally encode any arbitrarily complex, spatiotemporal, multi-agent behavior, in any environment with any number of agents into a compact representation of dual algebraic and geometric nature. This enables us to construct a probabilistic inference mechanism that predicts the collective strategy of avoidance among multiple agents, based on observation of agents’ past behaviors. We incorporate this mechanism into an online planner that enables an agent to understand a multi-agent scene and determine an action that not only contributes progress towards its destination but also reduction of the uncertainty of other agents regarding the agent’s role in the emerging strategy of avoidance. We evaluate our approach by comparing against a greedy baseline that only maximizes individual efficiency. Simulation results of statistical significance demonstrate that our planner results in a faster uncertainty decrease that facilitates the decision making process of co-present agents. The algorithm’s performance highlights the importance of topological reasoning in decentralized, multi-agent planning and appears promising for real-world applications in crowded human environments.

Keywords
Motion Planning, Navigation, Multi-Agent Systems, Inference, Topology, Braids

1 Introduction
The traffic flow in human environments, such as crowded hallways, sidewalks, and rooms may often be characterized as unstructured and even unpredictable, as a result of the lack of formal rules to control traffic and the lack of explicit communication among agents. Nonetheless, humans are capable of traversing such environments with remarkable efficiency, without hindering one another’s motion. Human navigation not only achieves collision avoidance; it does so while respecting several social considerations, such as the passing preference of others and their personal space (Hall 1990). This behavior has largely been attributed to trust, with pedestrians trusting that others will behave competently, according to Wolfinger (1995). This form of trust enables humans to infer the intentions of others, under the assumption of rational action (Csibra and Gergely 2007) but also effectively communicate their own intentions by leveraging various implicit communication channels, broadcasting and receiving information through their path shape, their body posture, their gaze etc. This exchange of information through nonverbal, implicit communication, enables humans to negotiate and agree on a joint strategy of avoidance. Complying with this strategy yields a socially acceptable outcome for everyone in the scene.

\(^1\)Sibley School of Mechanical & Aerospace Engineering, Cornell University, Ithaca NY, 14853, USA.
\(^2\)Department of Computer Science, Cornell University, Ithaca NY, 14853, USA.

Corresponding author: Christoforos I. Mavrogiannis, Upson Hall, Room 568, 124 Hoy Road, Ithaca, NY 14853.
Email: cm694@cornell.edu
Inspired by the outlined cooperative mechanisms that humans employ towards ensuring collision-free and socially-compliant encounters, we design a navigation planning framework for artificial agents, navigating crowded environments. We explicitly incorporate the concept of cooperation into an agent’s decision making process by constructing an inference mechanism that reasons about joint strategies of avoidance. A joint strategy essentially corresponds to an avoidance protocol that all agents follow, while making progress towards their destinations (see Fig. 1).

Once a consensus over a joint strategy is established, the agents may be trusted to finesse the details of their path plans, in a compliant fashion. In order to accelerate the convergence to a state of consensus, the agents select behaviors that (1) communicate their own intentions but also (2) communicate acknowledgment of the intentions of others. This results in a faster decrease of uncertainty, which not only facilitates everyone’s decision making but also often leads to overall efficiency improvement.

In this work, we represent joint strategies as topological patterns of agents’ trajectories, which we model as topological braids (Birman 1975) by employing the construction of braid groups from low-dimensional topology. Based on this representation, we design an analytical belief distribution over joint strategies that allows an agent to estimate what strategies are mutually acceptable by everyone and approximate the effect of its actions to the belief of others. We then employ an information-theoretic approach to manipulate this belief towards a state of low information entropy, which corresponds to higher confidence over the emerging evolution of the scene. We encode this specification into the agent’s decision making policy, which determines an action by compromising between personal efficiency and contribution to consensus establishment.

In contrast to the majority of existing approaches that are either too myopic, only focusing on local collision avoidance resolution (e.g. (Helbing and Molnár 1995; van den Berg et al. 2009) and related frameworks), or too specific, reproducing demonstrated behaviors in specific contexts (e.g. (Kretzschmar et al. 2016; Kim and Pineau 2016; Chen et al. 2017)), our approach enables agents to execute local collision avoidance maneuvers with a global, multi-agent collision-avoidance horizon in mind. More specifically, we contribute: (1) a formal model that may represent and characterize topologically any – arbitrarily complex – multi-agent behavior in any environment with any number of agents; (2) a human-inspired inference mechanism for predicting collective, multi-agent navigation behaviors; (3) an online decision making mechanism for decentralized, collaborative navigation planning that accelerates the rate of convergence to a mutually beneficial joint strategy; (4) simulation results of statistical significance that demonstrate the importance of incorporating a collective, global spatiotemporal understanding in the planning process. Our framework was designed according to the insights of sociology studies on pedestrian behavior (Wolfinger 1995) and psychology studies on action interpretation (Csibra and Gergely 2007), reflecting our scope of employing it on a mobile robot platform for navigation in crowded human environments. The topological structure that our model offers to the motion planning process is expected to reduce the emergence of undesired situations such as deadlocks and livelocks that are frequently observed in practice in human-robot interaction.
pedestrian encounters. With this work, we aim at providing a thorough, proof-of-concept exploration of the aforementioned fundamental idea in a case study on a simplified, discretized board game setup that represents an abstracted version of the highly-complex real world problem.

This paper extends our past work (Mavrogiannis and Knepper 2016) with the incorporation of the following contributions: (1) a more thorough literature review; (2) a more rigorous presentation of the topological foundations of our approach; (3) advancements and improvements to our modeling representations; (4) an extended discussion on the inference mechanism and a more principled inference construction that incorporates new heuristics and handles uncertainty over agents’ destinations; (5) a more detailed presentation of the algorithm design that includes a discussion of the algorithm’s complexity; (6) an extensive simulated evaluation, demonstrating results with verified statistical significance.

2 Related Work

Our work focuses on the development of a navigation planning framework with the end goal of being employed by robots operating in multi-agent human environments. As a result, it draws principles from several fields, such as action interpretation, human and robotic navigation, crowd simulation and tracking and topology. In this section, we review relevant literature from these communities, discuss works that influenced our approach and highlight the features that make our framework unique.

2.1 Human Navigation

Understanding and modeling human navigation accurately has been the focus of researchers from various fields for a long time. Karp et al. (1977), in their definition of the Mini-Max Hypothesis of Urban Life, specify that “urbanites seek to minimize involvement and to maximize social order”. This idea is also present in Wolfinger’s definition of the Pedestrian Bargain (Wolfinger 1995), a concise, high-level protocol of foundational social rules that regulate pedestrian navigation: (1) “people must behave like competent pedestrians” and (2) “people must trust copresent others to behave like competent pedestrians”. Trust to the rules of the bargain constitutes the basis of smooth co-navigation in human environments as it enables pedestrians to plan with the expectation that others will also behave competently and thus cooperate to resolve potential conflicts. This is enabled through a sophisticated mechanism of perception and action, enabled through information exchange mostly via path shape, body posture and gaze (Goffman 1966) that has been widely studied from a number of different fields. For example, Carton et al. (2016a) studied the trajectory planning horizon of humans in locomotion tasks towards informing the design of models for the prediction of human walking behaviors. In the field of psychology, Warren (2006) proposed a model that may describe organization in human behavior in a number of tasks by treating an agent and its environment as a pair of coupled interacting dynamical system. More broadly, Csibra and Gergely (2007) highlighted the teleological nature of human inference, suggesting that humans tend to attribute goals to observed actions.

Unlike human navigation which is heavily based on a multi-modal information exchange, our approach only considers the modality of path shape. Our planning framework aims at generating motion that maximizes social order through a local, collision-free action selection with a global lookahead. This is made possible with a principled design of a goal-driven inference that connects individual and collective behavior towards enabling artificial agents to understand the effects of their actions on the behaviors of others.

2.2 Multi-Agent Simulation

The problem of generating smooth, collision-free, multi-agent simulations has been central in a number of applications, ranging from city planning to the study of evacuation scenarios and even computer game design. To this end, a class of works models agents as interacting particles, attracted to their destinations and repulsed by others. Within this class, the Social Force model (Helbing and Molnár 1995) has been one of the first and most influential approaches, whereas several works have employed similar models with additional considerations such as discomfort fields (Treuille et al. 2006), local predictive processes (Hoogendoorn and Bovy 2003; Karamouzas et al. 2009) and cognitive heuristics (Moussaid et al. 2011; Farina et al. 2017). Some works have employed data-driven techniques to learn the parameters of human navigation in different contexts from simulated (Henry et al. 2010) or real world demonstrations (Karamouzas et al. 2014). Finally, van den Berg et al. (2009) and Knepper and Rus (2012) have proposed decentralized motion planners that explicitly leverage the expectation of cooperation.

Our planner is also cooperative by design. However, in contrast to most of the aforementioned approaches that either treat agents as moving obstacles or make purely local motion prediction, our inference mechanism enables agents to understand that they are
part of a crowd of intelligent agents that cooperate to reach consensus over a collision avoidance protocol.

2.3 Multi-Agent Tracking

From a different perspective, the computer vision community has contributed a number of methods focused on tracking the local or global behavior of pedestrians in different environments. For example, Scovanner and Tappen (2009), Pellegrini et al. (2009), Alahi et al. (2016) and Ma et al. (2017) present data-driven models for local short-term trajectory predictions for interacting pedestrians, that make use of models of social interactions, whereas Zhou et al. (2012) predict large-scale, global pedestrian interactions. These contributions are of particular significance both for offline labeling of pedestrian datasets but also for online tracking of multi-agent behaviors for robotics applications.

Similarly to some of the aforementioned approaches, our proposed inference mechanism enables agents to perceive the global nature of observed actions. Under the assumption of rationality (agents move efficiently towards intended destinations), our prediction model essentially scores the set of possible multi-agent collision avoidance strategies that the agents could follow to reach their destinations in a collision-free fashion. In a recent work we showed how such a model can be learned from deep neural sequence-to-sequence architectures (Mavrogiannis et al. 2017).

2.4 Social Robot Navigation

Humans are intelligent agents, reacting to observed behaviors upon making inferences about their own future behaviors. Thus, enabling robots to navigate seamlessly alongside humans in pedestrian environments requires an incorporation of our understanding of human navigation and decision making into the robot’s motion planning processes.

To this end, a class of works have proposed optimization-based planners that consider different criteria related to social compliance. Sisbot et al. (2007) presented a cost-based planner that incorporates considerations of human comfort and context-specific social conventions, whereas Park et al. (2012) proposed an online model-predictive control framework that generates locally optimal, smooth, collision-free trajectories for autonomous robotic wheelchair navigation. Mead and Matarić (2017) consider the incorporation of psychophysical proxemics theory into a cost based planner, aiming at driving the robot to configurations that facilitate speech and gesture-based social interactions between a human and a robot.

Another class of works has focused on reproducing observed human behaviors. Bennewitz et al. (2005), Sehestedt et al. (2010) and Ziebart et al. (2009) proposed planning frameworks that generate socially compliant robot motion through the use of human motion prediction models, learned from demonstrations of human trajectories in specific environments. Trautman et al. (2015) developed a planning framework that enables the generation of competent robot motion through dense human crowds via the consideration of a learned inference mechanism over cooperative collision avoidance strategies. Kim and Pineau (2016) and Kretzschmar et al. (2016) proposed planning frameworks for generating humanlike robot motion, based on cost functions that model human preferences as observed in datasets collected from human demonstrations or from robots teleoperated in crowded environments. Chen et al. (2017) learn a model for generating motion that respects culture-specific social norms such as passing from the right hand side. Finally, Bera et al. (2017) learn a probabilistic planning and prediction model that incorporates psychological and social constraints, based on identifying personality traits of pedestrians. For an extensive discussion of the state of the art in social robot navigation, the reader may refer to Thomaz et al. (2016).

The main contribution of this paper is an online, cost-based navigation planner for the generation of socially competent robot motion in multi-agent environments. Our framework aims at enabling a robot to interact (passively) with other agents, by socially avoiding them and does not consider the task of (active) social engagement. In contrast to some of the aforementioned data-driven approaches that imitate demonstrated human behaviors or social norms and are thus domain and context-specific by design, our framework focuses on the underlying topological foundation of multi-agent collision avoidance, which represents an inherently generalizable basis across domains and contexts.

Our work leverages the foundational idea that — under the assumption of rationality — the collision avoidance constraints (with the environment and with others) introduce a topological structure on agents’ global behaviors. To the best of our knowledge, this work is the first to incorporate the existence of this structure into agents’ inference mechanisms. This is done through the introduction of an inherently-generalizable representation of collective behavior, based on the braid group, a group of topological objects with a dual geometric and algebraic nature. This representation enables agents running our algorithm to reason about the topology of the spatiotemporal
pattern of everyone’s future trajectories instead of reasoning individually about the detailed trajectories of other agents.

Conceptually, the frameworks of Trautman et al. (2015) and Kretzschmar et al. (2016) are similar in that they also incorporate models of joint strategies into their planning processes. However, their models are not as expressive as our model of multi-agent collision-avoidance behaviors, based on topological braids. Braids allow our agents to understand the effect of their actions on the decision making of others and make decisions of global outlook that serve the social welfare.

2.5 Intention-Aware and Intent-Expressive Motion Planning

Over the past few years, a significant amount of research has been devoted to the design of planning frameworks for explicit or implicit human-robot interaction. Ensuring safe, natural, seamless and efficient interaction under this setting often necessitates the development of mechanisms for intention recognition and nonverbal communication. Towards this goal, roboticists have been inspired by the insights of studies on the principles underlying action interpretation by humans. Representative works include the studies of Csibra and Gergely (1998, 2007), Baker et al. (2009) and Wiese et al. (2012) who have highlighted the teleological nature of human inference, i.e., their tendency to interpret observed actions as a model that captures the entanglement of particle trajectories in a fluid (Thiffeault 2010); in a similar fashion, we employ braids to model the entanglement of the trajectories of navigating agents.

Based on these insights, Dragan and Srinivasa (2014) formalized the concepts of Legibility and Predictability as properties of motion that allow an observer to make a correct inference of another agent’s goal, given observation of its action and of its future action given knowledge of its goal, respectively, under the principle of rational action. These formalisms may be used for encoding intentions into robot motion; intent-expressive robot motion has been shown to lead to increased efficiency in human-robot collaboration but also to reduced planning effort for humans (Carton et al. 2016b). Generalizing these concepts to any human-robot collaboration task, involving any type of modality, Knepper et al. (2017) developed a formal framework for planning implicitly communicative actions.

Finally, the recent interest in autonomous driving has led to several works specializing on a street navigation context. For example, Bandyopadhyay et al. (2012) and Ferguson et al. (2015) presented navigation frameworks for planning autonomous car navigation, based on models for predicting pedestrian intentions, whereas Sadigh et al. (2016) proposed a planning framework that reasons about human drivers’ mental models to plan actions that influence their decision making towards desired outcomes.

Our planner aims at generating intent-expressive robot behaviors in dynamic multi-agent environments. In particular, it aims at manipulating an agent’s path shape to convey its intentions to others. Our inference mechanism allows our agents to monitor the uncertainty over future multi-agent behaviors and take information-rich or legible actions that aim at reducing this uncertainty. In contrast to existing approaches that generate legible robot motion in static and structured environments (e.g. (Dragan and Srinivasa 2014)), our framework is the first to address the problem of planning legible motion in a dynamic, multi-agent environment.

2.6 Braids in Robotics

The foundation of this work is the topological construction of braids from the field of low-dimensional topology (Kassel and Turaev 2008). The formalism of braids, first presented by Artin (1947a,b) and extensively studied by Birman (1975) has been an inspiration for applications in various disciplines. Our approach is specifically inspired by the use of braids as a model that captures the entanglement of particle trajectories in a fluid (Thiffeault 2010); in a similar fashion, we employ braids to model the entanglement of the trajectories of navigating agents.

The idea of employing braids in planning problems in robotics is not new. It may be traced at least as back as to Ghrist (1999), who drew a parallel between braids and configuration spaces of robotic systems. Later, Diaz-Mercado and Egerstedt (2017) developed a framework for centralized multi-robot navigation, in which the agents are assigned trajectories that contribute to a topological pattern corresponding to a specified braid. Although we are also making use of braids to model multi-robot behaviors, the scope of our approach is inherently different, since our target application concerns navigation in dynamic environments where no explicit communication takes place among agents. In our case the agents do not follow a pre-specified braid pattern, but rather employ a braid-based probabilistic reasoning to reach a topological consensus that best complies with everyone’s intentions or preferences. For our purposes, braids provide an abstraction of the complex spatiotemporal multi-agent dynamics of interaction among agents. This abstraction enables artificial agents to reason about uncertainty in a principled fashion,
as the dual geometric and algebraic representation of braids enables them to enumerate a set of diverse, topologically distinct scene evolutions. As a result, our algorithm generates socially competent behaviors, i.e., behaviors that explicitly take into consideration the social welfare of the system of all agents.

To showcase the benefits of braids for multi-agent planning problems in robotics, we consider a case study on an abstracted, discretized version of the real-world, multi-agent navigation problem. Simulation results demonstrate that our agents learn to coordinate faster than purely greedy agents. Recent work of ours (Mavrogiannis et al. 2017, 2018) has demonstrated the potential of the proposed planning architecture for real-world environments by considering continuous workspaces and comparing against other widely used baselines such as the Social Force model (Helbing and Molnár 1995) and the Optimal Reciprocal Collision Avoidance (ORCA) framework (van den Berg et al. 2009). Ongoing work involves validation of our framework on an autonomous telepresence robot platform, designed to navigate around humans in crowded workspaces.

3 Foundations

Consider a set of \( n \geq 2 \) agents \( N = \{1, \ldots, n\} \) navigating a workspace \( \mathcal{Q} \subset \mathbb{R}^2 \). Denote by \( q_i \in \mathcal{Q} \) the configuration of agent \( i \in N \). Agent \( i \) starts from an initial configuration \( q_i^0 \in \mathcal{Q} \) at time \( t = 0 \) and reaches a final configuration \( q_i^d \) at time \( t = T_i \). The final configuration \( q_i^d \) corresponds to a landmark \( d_i \) from a set of landmarks \( D \subset \mathcal{Q} \) (we assume that \( d_i \neq d_j \) for any two agents \( i, j \in N \)). The path agent \( i \) follows to reach its destination is a function \( \xi_i : [0, T_i] \rightarrow \mathcal{Q} \).

The agents do not explicitly exchange any kind of information with each other but are assumed to be acting rationally, which in our context means that (1) they always aim at making progress towards their destinations and (2) they have no motive for acting adversarially against other agents (e.g. blocking their paths or colliding with them). The notion of rationality is in line with the concept of competence as described by Wollinger (1995) in his definition of the Pedestrian Bargain and also with the concept of teleological reasoning that appears to be foundational for human inference (Csibra and Gergely 2007).

3.1 Game - Theoretic Setup

Inspired by Wollinger’s observations on the cooperative nature of human navigation, we approach the problem of robotic navigation in multi-agent, dynamic environments as a finitely repeated coordination game of imperfect information and perfect recall. The game is repeated a finite number of rounds \( K \), which is unknown a priori and corresponds to the round at which the slowest agent reached its destination. At each round \( k \in \{1, \ldots, K\} \), each agent \( i \) decides on an action \( a_i^k \) from a set of available actions (actions that could potentially lead to collisions and actions that violate the agent’s dynamics are excluded) \( A_i^k \) by minimizing a cost function \( u_i \). The agents are simultaneously selecting their actions and therefore they have no access to other agents’ plans (imperfect information); we assume however that they maintain a history of all previous rounds (perfect recall). The result of all agents’ decision making at round \( k \) is the strategy profile \( A^k = \{a_1^k, \ldots, a_n^k\} \). The sequence of strategy profiles of all rounds, from the beginning to the end of the game, \( A_1 \ldots A_K \), forms a global joint strategy \( \tau \) that the agents engaged in to avoid each other, while making progress towards their destinations.

Although the agents do not explicitly coordinate with each other to decide on a joint strategy, their strategy profiles at every round gradually reinforce and contribute to one. Imbuing artificial agents with an understanding of the collective dynamics of a multi-agent scene may allow them to make informed and socially competent decisions that contribute to the avoidance of undesired situations, such as hindering others’ paths, deadlocks and livelocks. For this reason, besides personal efficiency, it is important that agents’ cost functions incorporate a model of multi-agent efficiency, reflecting the social welfare of the system of agents.

In this paper, we develop a topological model of joint strategies, employing the formalism of braids (Birman 1975), which we use to develop a human-inspired inference mechanism, supported by studies on human action interpretation. Our mechanism provides a principled prediction over the scene evolution that allows agents to take into consideration the effect of their decision making on any observers.

3.2 A Topological Model of Joint Strategies

Let us collect the state of the system of all agents in a tuple \( Q = (q_1, \ldots, q_n) \in \mathcal{Q}^n \). The system state evolves from a starting configuration \( Q^s = (q_1^s, \ldots, q_n^s) \) to a final configuration \( Q^d = (q_1^d, \ldots, q_n^d) \), by following a path \( \Xi : [0, T] \rightarrow \mathcal{Q}^n \), from the space of system paths \( \mathcal{Z} \), starting from \( Q^s \) and ending at \( Q^d \). The system path is a function \( \Xi : [0, T] \rightarrow \mathcal{Q}^n \), where \( \Delta = \{Q = (q_1, q_2, \ldots, q_n) \in \mathcal{Q}^n : q_i = q_j \) for some \( i \neq j \in N \} \) is the set of all system states with agents in collision and \( T = \max_{i \in N} T_i \) (it is assumed that agents remain at their destinations until
everyone reaches their own). Naturally $\Delta$ partitions the space of system paths $Z$ into a set of classes of homotopically equivalent system paths. Each such class has distinct topological properties which indicate a distinct joint strategy that the agents followed to reach their destinations. To enumerate such classes of joint strategies but also to characterize topologically the collective behavior of a system of agents, we develop a model of joint strategies using the concept of braids (Birman 1975). In the following paragraphs, we provide a primer on braids, establish a correspondence between braids and collective navigation behaviors and define a topological model of joint navigation strategies.

3.2.1 Background on Braids

Braids are topological objects with algebraic and geometric presentations. We first introduce them as geometrical entities, following a presentation based on Artin (1947b) and continue with a discussion of their algebraic presentation and their group formation.

Denote by $x, y, z$ the cartesian coordinates of a Euclidean space $\mathbb{R}^2 \times I$. A braid string is a curve $X(z) : I \rightarrow \mathbb{R}^2$ that increases monotonically in $z$, i.e., has exactly one point of intersection $X(z) = (x, y)$ with each plane $z \in I$. A braid on $n$-strings or $n$-braid (see Fig. 2) is a set of $n$ strings $X_i(z)$, $i \in N = \{1, \ldots, n\}$ for which:

1. $X_i(z) \neq X_j(z)$, for $i \neq j \forall z \in \mathbb{R}$
2. $X(0) = (i, 0)$ and $X(1) = (p(i), 0)$,

where $p(i)$ is the image of an element $i \in N$, through a permutation $p : N \rightarrow N$ from the set of permutations of $N$, $\text{Perm}(N)$, defined as:

$$ p = \begin{pmatrix} 1 & 2 & \cdots & n \\ p(1) & p(2) & \cdots & p(n) \end{pmatrix}. $$

This geometric representation of a braid is commonly referred to as a geometric braid. More formally, a geometric braid is often represented with a braid diagram, a projection of the braid onto the plane $\mathbb{R} \times 0 \times I$ (see e.g. Fig. 3).

The set of all braids on $n$ strings, along with the composition operation, form a group $B_n$. The group may be generated from a set of $n-1$ elementary braids $\sigma_1, \sigma_2, \ldots, \sigma_{n-1}$ (see Fig. 3), called the generators of $B_n$, that satisfy the following relations:

$$ \begin{align*}
\sigma_j \sigma_k &= \sigma_k \sigma_j, & |j - k| > 1, \\
\sigma_j \sigma_k \sigma_j &= \sigma_k \sigma_j \sigma_k, & |j - k| = 1.
\end{align*} $$

A generator $\sigma_i$, $i \in \{1, 2, \ldots, n-1\}$ can be described as the crossing pattern that emerges upon exchanging the $i$th string (counted from left to right) with the $(i+1)$th string, such that the initially left string passes over the initially right one, whereas the inverse element, $\sigma_i^{-1}$, implements the same string exchange, with the difference that the left string passes under the right (see Fig. 4). An identity element, $e$, is a braid with no string exchanges.

Two braids $b_1, b_2 \in B_n$ may be composed through the composition operation ($\cdot$), which is algebraically denoted as a product $b_1 \cdot b_2$. Geometrically, this composition results in the pattern that emerges upon attaching the lower endpoints of $b_2$ to the upper endpoints of $b_1$ and shrinking each braid by a factor of 2, along the $z$ axis (see Fig. 3, Fig. 4). Any braid can

\begin{figure}
\centering
\includegraphics[width=\textwidth]{figure2.png}
\caption{A set of 4 particles are initially $(z = 0)$ arranged along the $x$-axis, on the points 1, 2, 3 and 4. Through a sequence of rearrangements, the particles finally $(z = 1)$ reach a final arrangement on the points 2, 3, 1 and 4 respectively. The pattern of their trajectories is a geometric braid.}
\end{figure}

\begin{figure}
\centering
\includegraphics[width=\textwidth]{figure3.png}
\caption{The generators of the Braid Group $B_n$.}
\end{figure}

\begin{figure}
\centering
\includegraphics[width=\textwidth]{figure4.png}
\caption{Example of the Composition operation $\sigma_1 \cdot \sigma_2^{-1}$ for $\sigma_1, \sigma_2^{-1} \in B_n$.}
\end{figure}
be written as a product of generators and generator inverses. This representation is commonly referred to as an algebraic braid or a braid word (Fig. 4).

### 3.2.2 Abstracting Joint Strategies Using Braids

Denote by $f_x : \mathcal{Q}^n \to \text{Perm}(N)$ a function that takes as input the system state $Q \in \mathcal{Q}^n$ and outputs a permutation $p \in \text{Perm}(N)$ corresponding to the arrangement of all agents in order of increasing $x$-coordinates. As the agents move towards their destinations, they employ navigation strategies – maneuvers to avoid collisions. These contribute to a system path $\Xi$ which corresponds to a path $\tau$ of permutations $\pi : [0, T] \to \text{Perm}(N)$ that may be extracted by evaluating $f_x$ throughout the whole path $\Xi$. This path can be represented by a sequence of permutations of minimal length $\pi^* = (p_0, \ldots, p_K)$, i.e., $p_{j-1} \neq p_j$, $\forall j \in \{1, \ldots, K\}$ and consecutive waypoints are adjacent transpositions, i.e., permutations that differ by exactly one swap of adjacent elements. Due to continuity, a transition from the $(j-1)$th permutation, $p_{j-1}$, to the $j$th permutation, $p_j$, implies the occurrence of an event $\tau_j$, which may be described as the intersection of the $x$-projections of the paths of two agents that were adjacent in the permutation $p_{j-1}$. The event $\tau_j$ may be represented as an elementary braid $\tau_j \in \sigma_i \in \mathcal{S}_n$, $i \in \{1, \ldots, n-1\}$, where $i$ corresponds to the index of the leftmost swapping agent in permutation $p_{j-1}$. Therefore the whole execution from $t = 0$ to $t = T$ may be abstracted into the braid that corresponds to the temporal sequence of events:

$$\tau = \tau_1 \tau_2 \ldots \tau_K \in B_n. \quad (4)$$

This braid word not only constitutes a topological characterization of the system path (see Fig. 5 for an example of characterizing a system path as a braid) but it also represents a topological class of system paths that are homotopy-equivalent with the system path in consideration. In the remainder of this paper, we will be referring to the sequence $\tau$ as the joint strategy or the entanglement of the system path. Essentially, we model the space of joint strategies $\mathcal{T}$ as the braid group, i.e., $\mathcal{T} := B_n$.

**Remark 1.** In our model, a braid constitutes a 2-dimensional abstraction of a 3-dimensional pattern of trajectories. Depending on the selection of the projection line, a different braid emerges. Although a change of projection line only changes the braid by conjugation (Thiffeault 2010), in practice, this implies that a set of non-communicating agents might encode the same joint strategy with a different symbolic representations (braids). However, this does not affect the convergence to a consensus on a mutually acceptable joint strategy among agents; despite their different representations, they still take actions that contribute towards the same outcome, as will be shown in our simulation results. Therefore, the selection of the projection plane for a planning agent is not important, as long as its action selection process is consistent with it.

### 4 Inference of Collective Behaviors

An individual agent has no sole control over a specific joint strategy. The joint strategy is an emergent behavior, resulting from the superposition of the individual strategies of all agents. In fact, since agents are not explicitly communicating or coordinating, they cannot have a priori knowledge or a precise sense of the actual joint strategy they are about to follow. However, understanding the dynamics of collective behavior may allow agents to adopt individual navigation strategies that allow others to infer their intentions more clearly, thus facilitating everyone’s decision making by reducing uncertainty fast. Judging from our everyday life experience, we may argue that this is the case with humans as well. When humans encounter others in a hallway, they do not exactly know the specific joint strategy they will be following. However, they realize that their decisions are coupled with the decisions of others and are able to reach a consensus regarding an avoidance protocol that is comfortable for everyone.

In this section, we present a probabilistic intention inference mechanism that connects an observed system path with a future system path topology, designed according to the insights of psychology studies on...
human action interpretation. This design is motivated by our goal of employing our framework on an autonomous social robot that will be navigating in a safe and socially competent fashion around human pedestrians.

4.1 Teleological Reasoning in Multi-Agent Navigation

Csibra and Gergely (1998, 2007) argued that the mechanisms of human action interpretation are teleological in nature, i.e., humans tend to interpret observed actions as goal-directed in a given context. Following their insights, we design an inference mechanism of the form $P(\tau | \Xi_t, M_t)$, corresponding to a belief over an emerging joint strategy $\tau \in \mathcal{T}$ given a partial system path $\Xi_t$ and the state of the context at time $t$. The joint strategy represents a collective goal, whereas the system path plays the role of the action. By context, we refer to publicly available information, such as a model of the static environment (e.g. a map, obstacles, points of interest etc) but also information extracted through processing, e.g. by employing secondary inference mechanisms regarding group formations, identification of reactive agents etc.

4.2 Inferring Joint Strategies from Context

From (4), the belief $P(\tau | \Xi_t, M_t)$ may be expanded as:

$$P(\tau | \Xi_t, M_t) = P(\tau_1, \ldots, \tau_K | \Xi_t, M_t), \quad (5)$$

which, by applying the chain rule, may be factored as:

$$P(\tau | \Xi_t, M_t) = \prod_{k=1}^{K} P(\tau_k | \Xi_t, M_t). \quad (6)$$

This belief quantifies the likelihood of a sequence of events $\tau_1, \ldots, \tau_K$ given observation of agents’ past behaviors and the context. Essentially, this corresponds to predicting the minimal sequence of permutations $\pi^*$ but also the quality of the physical transitions between consecutive permutation waypoints (passing from the right/left hand side).

A joint strategy describes the avoidance protocol that the agents followed to avoid each other throughout the scene evolution, while navigating from $Q^*$ to $Q^d$. These system path endpoints are not incorporated in the definition of the strategy as geometric entities but rather as the permutations $p^* = f_x(Q^*)$, $p^d = f_x(Q^d)$. This design decision reflects the observation that an agent navigating a multi-agent environment does not need to know the precise intended destinations of others to avoid collisions successfully; they just need to understand their passing preferences/intentions. However, the geometric arrangements of agents’ intended final configurations greatly influence the convergence to a joint strategy $\tau$. In particular, given the initial permutation $p^*$, only a subset $\mathcal{T} \in B_n$ may lead to $p^d$.

Given the importance of the final permutation in the prediction of a joint strategy, we may break the problem of predicting a joint strategy into two separate inference subproblems: (1) a prediction of the final permutation and (2) a prediction of a compatible system path entanglement – braiding word, conditioned on the predicted final permutation. Following this reasoning, eq. (6) may be rewritten as the following product:

$$P(\tau | \Xi_t, M_t) = P(p^d, \tau^d | \Xi_t, M_t)$$

$$= P(\tau^d | p^d, \Xi_t, M_t) P(p^d | \Xi_t, M_t), \quad (8)$$

where $\tau^d \in \mathcal{T}$ represents a braid that is compatible with the prediction of a final permutation $p^d$, given the permutation $p_t = f_x(Q_t)$ corresponding to the current system state $Q_t = \Xi(t)$. Fig. 6 depicts a graphic representation of the structure of our inference mechanism.

4.2.1 Inferring the Final Permutation of the System

A planning agent knows with certainty its own destination but has no access to other agents’ destinations. Although there is no need to make an inference regarding others’ actual destinations, it is important to infer a final permutation $p^d$ so as to make an informed inference regarding the emerging joint strategy, as discussed in the previous subsection. Under the assumption of rationality and given a model of the world, stored in the context $M_t$, the planning agent may infer the general directions of others through a belief:

$$P(p^d | \Xi_t, M_t) = P(p^d(1), \ldots, p^d(n) | \Xi_t, M_t), \quad (9)$$

where

$$p^d = \begin{pmatrix} 1 & \ldots & n \\ p^d(1) & \ldots & p^d(n) \end{pmatrix}. \quad (10)$$

For simplicity, let us assume that the planning agent’s ID is #1. Then, under the assumption that all agents are moving towards destinations from the known set $D$, the agent’s index under the permutation $p^d$, i.e., $p^d(1)$, is constrained. Leveraging this, the planning agent may conclude to a subset of feasible permutations $P \subset \operatorname{Perm}(N)$, by ruling out any incompatible permutations from $\operatorname{Perm}(N)$ as unlikely. Under the assumption that the cardinality $|D| \geq |N|$, multiple assignments of destinations to agents may be possible for each permutation in $P$. 

Prepared using sagej.cls
More formally, for each compatible permutation \( p^d_m \in Perm(N), m \in \{1, \ldots M\} \), where \( M = (n - 1)! \), we may derive a set of possible destination assignments \( \Delta_m \) that are compatible with (1) \( p^d_m \), and (2) the image of agent #1, through the permutation \( p^d_m \), i.e., \( p^d_m(1) \), corresponds to a final arrangement of all agents to destinations of \( D \), with agent #1 at its destination \( d_1 \). Essentially, an assignment \( \delta \in \Delta_m \) is an injective function \( \delta : N \to D \) that maps all agents to a subset of landmarks from \( D \). Upon marginalizing over all possible \( \delta \in \Delta_m \), the probability that the final permutation \( p^d = p^d_m \) may be derived as:

\[
P(p^d = p^d_m | \Xi_t, M_t) = \sum_{\delta \in \Delta_m} P(p^d = p^d_m | \Xi_t, M_t, \delta)
\]

\[
P(\delta | \Xi_t, M_t). \hspace{1cm} (11)
\]

By definition, \( p^d = p^d_m \), if we know that agents are going to \( \delta \) and therefore eq. (11) may be simplified as follows:

\[
P(p^d = p^d_m | \Xi_t, M_t) = \sum_{\delta \in \Delta_m} P(\delta | \Xi_t, M_t). \hspace{1cm} (12)
\]

The destination that an agent is aiming for is conditionally independent of the destinations of others, given \( \Xi_t \) and \( M_t \), therefore we may express \( P(\delta | \Xi_t, M_t) \) as:

\[
P(\delta | \Xi_t, M_t) = \prod_{j=2}^{n} P(q^d_j = \delta(j) | \Xi_t, M_t) \hspace{1cm} (13)
\]

where we incorporated the fact that agent #1 is certain about its destination. Finally, under the assumption of rationality, we follow an approach similar to Dragan and Srinivasa (2014) to model \( P(q^d_j = \delta(j) | \Xi_t, M_t) \) as:

\[
P(q^d_j = \delta(j) | \Xi_t, M_t) = \frac{1}{Z} \exp\left(-c(\xi_j) - c^*(q^d_j, \delta(j))\right), \hspace{1cm} (14)
\]

where \( \xi_j \) is the path agent \( j \) has followed so far, \( c \) measures the length of a path, \( c^* \) returns the shortest path between two points and \( Z \) represents a normalizer across the set of landmarks \( D \).

Algorithm 1 outlines the process of scoring all compatible final permutations from \( Perm(N) \). Function \text{Get_Perm} \ returns a permutation \( p_t \) of the set \( D \), corresponding to the arrangement of landmarks in an order of increasing \( x \)-coordinates with respect to the agent’s frame. Then, all permutations are accessed and checked for compatibility with the planning agent’s destination \( d \) (function \text{Check_Perm}). In case a permutation is compatible, the set of possible assignments of agents to destinations that are in compliance with the permutation, is extracted with function \text{Get_Assignments} and then scored (function \text{Score_Assignments}). Otherwise, the corresponding permutation is assigned a zero score. The scores are finally normalized and returned in the form of a probability distribution \( P \).
4.2.2 Inferring the Path Entanglement

The second distribution, over system path entanglements \( P(\tau^d|p^d, \Xi_t, M_t) \) is a harder distribution to approximate. Especially events that take place further than one event ahead may be impossible to be traced back to the decisions that agents have made until time \( t \). For this reason, in this work, we approximate \( P(\tau^d|p^d, \Xi_t, M_t) \) as:

\[
P(\tau^d|p^d, \Xi_t, M_t) = P(\tau^d_1, \ldots, \tau^d_K|p^d, \Xi_t, M_t) \approx \frac{1}{\Gamma} P(\tau^d_1|p^d, \Xi_t, M_t),
\]

that is, given the state of execution at time \( t \), as expressed in \( \Xi_t \) and \( M_t \), the probability of a braid word is set to be approximately proportional to the probability of the next generator, observed after time \( t \), being equal to the one prescribed by \( \tau_1 \), where \( \Gamma \) is an appropriate normalizer. All entanglements that share the same first generator \( \tau_1 \) are assigned the same probability. Finally, in case the current permutation of agents is equal to the predicted final permutation \( p^d \), it is assumed that the only possible path entanglement is the trivial one, i.e., \( \tau = e \), and the rest are assigned zero probabilities.

### Algorithm 1 Score Permutations \((d, \Xi, D, \text{perms})\)

**Input:** \( d \) – agent’s intended destination; \( \Xi \) – state history of all agents; \( D \) – list of landmark locations; \( \text{perms} \) – list of permutations.

**Output:** \( P \) – prob. distribution over permutations.

1: \( p_i \leftarrow \text{Get Permutation}(D) \)
2: for \( i = 1 : n! \)
3: \( \text{compatible} \leftarrow \text{Check Perm}(\text{perms}[i], p_i, d) \)
4: if \( \text{compatible} \) then
5: \( \Delta \leftarrow \text{Get Assignments}(p_i, \text{perms}[i]) \)
6: \( S[i] \leftarrow \text{Score Assignments}(\Delta, \Xi) \)
7: else
8: \( S[i] \leftarrow 0 \)
9: end if
10: end for
11: \( P \leftarrow S/\text{sum}(S) \)
12: return \( P \)

To model \( P(\tau^d_1|p^d, \Xi_t, M_t) \), we first encode a generator \( \tau^d_1 \in \{\sigma_1, \sigma^{-1}_1, \ldots, \sigma_n, \sigma^{-1}_n\} \) into a tuple \( g = (\text{swap}_g, \text{sign}_g) \), where \( \text{swap}_g \in \{1, \ldots, n-1\} \) contains the generator’s subscript (corresponding to the pair of agents that are exchanging sides) and \( \text{sign}_g \in \{-1, 1\} \) contains the generator’s superscript (how they are exchanging sides). This model allows us to decompose the generator prediction problem into (1) a prediction of the immediately swapping agents and (2) a prediction of the type of their swap:

\[
P(\tau^d_1 = g|p^d, \Xi_t, M_t) = P(\text{swap}_g, \text{sign}_g|p^d, \Xi_t, M_t)
\]

\[
= P(\text{sign}_g|\text{swap}_g, p^d, \Xi_t, M_t)
\]

\[
P(\text{swap}_g|p^d, \Xi_t, M_t).
\]

(17)

Regarding the prediction of the next swap, we employ the following model:

\[
P(\text{swap}_g|p^d, \Xi_t, M_t) = \frac{1}{H} \prod_{i=1}^{n-1} R_{\text{swap}}(i),
\]

(18)

where \( H \) is a normalizer across swaps and \( R_{\text{swap}} \) is defined as:

\[
R_{\text{swap}}(i) = \begin{cases} 
\frac{1}{1+\exp(-\Delta x_{i} - \epsilon)}, & i = \text{swap}_g \\
\frac{1}{1+\exp(\Delta x_{i} - \epsilon)}, & i \neq \text{swap}_g 
\end{cases}
\]

(19)

where \( \Delta x_{i} = x_{i+1} - x_{i} \) represents the \( x \)-distance between the agents of the pair \( i \) (agents \( p_i(i) \) and \( p_i(i+1) \) in the current permutation \( p_i \)) and \( \epsilon > 0 \) is a constant. Smaller distances indicate swaps corresponding to generators that are exponentially more likely.

Given a prediction of a \( \text{swap}_g \), the next step is to determine the type of swapping, i.e., \( \text{sign}_g \). In

---

Figure 7. Demonstration of the Momentum heuristic for predicting a generator superscript: The \( x \)-projections of the agents’ paths are about to cross, forming a \( \sigma_1 \) generator. The \( z \) component of their angular momentum is negative, indicating a tendency for a counterclockwise rotation, which indicates a negative braid exponent; likewise, in case the \( z \) component of the angular momentum were positive, the emerging exponent would be positive.
order to model $P(sign_y|\text{swap}_y, p^d, \Xi_t, M_t)$ we employ a heuristic based on the angular momentum of the system of the agents in consideration, which, assuming unit masses, may be defined as:

$$\vec{L}(\text{swap}_y) = \vec{r}_i \times \vec{v}_i + \vec{r}_{i+1} \times \vec{v}_{i+1}, \quad (20)$$

where $\vec{r}_i$, $\vec{r}_{i+1}$ are respectively the positions of the currently right and left agents, defined with respect to their current center of mass $\vec{r}_c = (\vec{r}_i + \vec{r}_{i+1})/2$, whereas $\vec{v}_i$, $\vec{v}_{i+1}$ are their respective velocities. For two masses moving on the same plane, the angular momentum is a vector, normal to the plane, along the direction of rotation. If the masses are about to rotate counterclockwise, with respect to an axis of reference, the $z$-component of the momentum, $L_z$, is positive, and negative in case the masses are about to rotate clockwise. Based on this observation, we model $P(sign_y|\text{swap}_y, p^d, \Xi_t, M_t)$ as:

$$P(sign_y|\text{swap}_y, p^d, \Xi_t, M_t) =$$

$$\begin{cases} 
\frac{1}{\Theta(1+\exp(-sign_y L_z(\text{swap}_y)))}, & \text{if } |\vec{v}_i| + |\vec{v}_{i+1}| > 0 \\
\frac{1}{\Theta(1+\exp(-sign_y \Delta y^i))}, & \text{otherwise}
\end{cases}, \quad (21)$$

where $\Theta$ is an appropriate normalizer. The more positive $L_z$ is, $P(sign_y = +1|\text{swap}_y, p^d, \Xi_t, M_t)$ gets exponentially closer to 1, whereas in the opposite case, $P(sign_y = -1|\text{swap}_y, p^d, \Xi_t, M_t)$ gets closer to 1. In case the velocities of both agents are currently zero, the corresponding scores only depend on their distance along the $y$-axis, $\Delta y^i = y_{i+1} - y_i$. Fig. 7 demonstrates schematically the concept of momentum (for nonzero velocities) and how it is used to predict $sign_y$.

Remark 2. It should be noted that the model of inference presented in this section constitutes an extended version of the one presented by Mavrogiannis and Knepper (2016), as it may handle a) uncertainty over destinations, b) redundancy of destinations (case with no unique mapping from a permutation to the set of destinations) and c) incorporates a novel heuristic for predicting the exponent of braid generators. However, it should also be noted that this distribution is a simplified approximation that cannot guarantee robust performance and generalization. We are using it to provide a proof of our concept. Recent work of ours (Mavrogiannis et al. 2017) presented a data-driven framework for directly learning to predict future trajectory topologies from simulated demonstrations of challenging multi-agent scenarios.

5 Decision Making

In multi-agent environments, where no explicit communication takes place among agents, uncertainty regarding everyone’s actions is typically high, which complicates decision making. Humans usually overcome such a complication by communicating implicitly, mostly through motion. Doing so is made possible through inference mechanisms that allow them to read the intentions of others and select socially compliant actions that reduce uncertainty. This enables them to reach a consensus over an avoidance protocol that serves everyone and ensures comfort, while making progress towards their destinations. The superposition of these considerations represents what – to our interpretation of the pedestrian bargain (Wolfinger 1995) – constitutes socially competent behavior in a pedestrian context.

To generate socially competent behaviors, we design a cost-based policy. Our cost function enables an artificial agent to take actions that not only contribute progress towards its destination but also towards a consensus over a joint strategy that appears to be mutually beneficial for everyone in the scene. Regarding the first specification, a distance-based efficiency cost is employed, whereas for the second one, the entropy of the distribution over joint strategies $P(\tau|\Xi, M)$ is used. The distribution allows the planning agent to estimate the long-term effects of an action in consideration and how it might influence the decision making of others. The reduction of the entropy of the distribution may allow the agent to select an action that reduces the uncertainty for everyone.

In the following subsections, we describe in detail our decision making framework.

5.1 Modeling Agents’ Cost Functions

We model the interests of an agent $i$ with a cost function $u_i : A_i \rightarrow \mathbb{R}$ that maps an action $a_i \in A_i$ to a real number. We design this cost to comprise two terms: (1) $E_i$, which represents the agent’s personal Efficiency and (2) $C_i$, which represents the state of Consensus over a joint strategy among agents, from the perspective of agent $i$, upon taking an action $a_i \in A_i$:

$$u_i(a_i) = \lambda E_i(a_i) + (1 - \lambda)C_i(a_i). \quad (22)$$

We define the personal efficiency term $E_i$, to be the length of the shortest path to the agent’s destination, whereas $C_i$ is modeled as the Information Entropy of the belief distribution over joint strategies $P(\tau|\Xi, M)$, from the perspective of agent $i$, i.e.,

$$C_i(a_i) = - \sum_{\tau \in \mathcal{T}} P(\tau|\Xi^+, M) \log_2 P(\tau|\Xi^+, M), \quad (23)$$

Prepared using sagej.cls
where $\Xi^+$ denotes the system path so far, $\Xi$, augmented with the action in consideration $a_i$. Finally, $\lambda$ is a weighting factor, expressing the compromise between efficiency and consensus. Formally, the decision making policy may be described as a minimization of eq. (22):

$$a_i^* = \arg\min_{a_i \in A_i} u_i(a_i). \quad (24)$$

Note that the cost function $u_i$ plays the role of a utility function, with the difference that lower values are better.

Overall, this policy enables an agent to make decisions that not only contribute progress towards its destination but also towards a mutually beneficial consensus over a scene outcome. The faster such a consensus is established, the lower the uncertainty will be for all agents throughout the remainder of the execution. The Efficiency term represents agents’ intention of reaching their destinations by spending low energy and is in line with the principle of rational action as highlighted in the definitions of the pedestrian bargain (Wolfinger 1995) and the teleological reasoning (Csibra and Gergely 1998). The Consensus term scores the current state of the global consensus among agents regarding the joint strategy to be followed and therefore, it directly incorporates a form of social understanding into the agent’s decision making policy. The lower the entropy, the lower the uncertainty regarding the emerging joint strategy. Thus, by consistently picking actions that contribute to entropy reduction, an agent communicates its intention of complying with a subset of scene outcomes that appear to be preferable by everyone according to the model $P(\tau | \Xi, M)$. As a result, the agents are expected to reach a consensus over $\tau$ easier and faster, avoiding ambiguous situations such as livelocks or deadlocks and reach their destinations with lower planning effort.

### 5.2 Planning Joint Strategies

In practice, making use of the distribution $P(\tau | \Xi, M)$ requires determining a set of candidate joint strategies $T$. The braid group $B_n$ is countably infinite; in practice, however, only a subset of joint strategies (braids) are meaningful under the context of a scene $M_t$ and given observations of agents’ past behaviors $\Xi_t$. In particular, as discussed in Sec. 4, given any compatible final permutation $p^d_m \in P$, only a subset $T^m \subset B_n$ may be achievable from the current permutation $p_t = f_x(Q_t)$. Consequently, there arises the problem of planning a set of joint strategies $T = \bigcup T^m, m = \{1, \ldots, M\}$, compatible with the set of different final permutations in consideration from $P$.

Planning a joint strategy that transitions the system from a permutation corresponding to the current state of the system, $p_t$, to a permutation corresponding to the final state of the system, $p^d_m$, may be decomposed into the following subproblems: (1) planning a path of permutations, $\pi^m = (p_t, \ldots, p^d_m)$ connecting $p_t$ with $p^d_m$ through a sequence of adjacent transpositions, and (2) assigning compatible elementary braids to the transitions between consecutive permutations. Each transition may be implemented in 2 different ways, i.e., by a compatible generator or its inverse. For example, a path $\pi$ of length $l_\pi$ (the number of transitions required to reach $p^d$ from $p^t$, through $\pi$) may be implemented by $2^{l\pi}$ different braids.

Assuming that we have concluded to a set $P \in Perm(N)$ of potential final permutations, for each one of them $p^d_m \in P$, we plan a set of paths of adjacent transpositions, $P^m$, based on which we generate a batch of joint strategies $T^m$. These are used to form the final set of joint strategies $T = \bigcup T^m, m = \{1, \ldots, M\}$.

In this paper, we convert the problem of planning a topological joint strategy into a search in a graph of permutations. In the following sections, we describe the construction of the graph and the planning procedure.

### 5.3 Permutation Graph Search

The set of all permutations on $N$, $Perm(N)$, along with the composition operation, form the symmetric
group $S_n$. $S_n$ is a group of order $n!$, that can be generated by the set of adjacent transpositions $\beta_j = [j \ j+1]$, for $1 \leq j < n - 1$.

We make use of the symmetric group to construct a graph $G = (V, E)$, where $V = \text{Perm}(N)$ and any pair of nodes $\nu_a, \nu_b \in V$ is only connected if $\exists \beta_{ab} \in S_n$ that permutes $\nu_a$ into $\nu_b$. The graph $G$ may be represented as a ($n-1$)-dimensional polytope, embedded in a $n$-dimensional space, which is commonly referred to as a permutohedron (Ziegler 1995). Fig. 9 depicts a permutohedron of order 4, along with example paths and indications of braid transitions.

Planning a path from a permutation $p^a$ to a permutation $p^b$, corresponding respectively to the vertices $\nu_a, \nu_b \in V$, is equivalent to finding a path of vertices-permutations that connect them. Fig. 8 illustrates the concept of planning a joint strategy. At planning time $t$, the agents have already followed trajectories $\Xi_t$. The planning agent (blue color) has predicted that they are aiming at reaching the destinations denoted by pointers of corresponding colors. Transitioning from the current system configuration to the predicted final system configuration corresponds to transitioning from the current permutation $p_t$ to a permutation $p^d_t$, both defined with respect to the dashed line, parallel to the x-axis of the agent’s body frame $\{B\}$.

5.4 Online Algorithm

In this section, we describe our algorithm design for online navigation planning that makes use of the components detailed in the previous sections. The algorithm compromises between making progress towards the agent’s destination and being respectful of everyone’s intentions, as inferred by their past behaviors. In case the planning agent does not observe other agents on its way to its destination, it switches to efficiency optimization.

The SCN algorithm (alg. 2) is our online algorithm for socially competent navigation. At every replanning cycle, the function $\text{Get Reactive Agents}$ first returns a list of agents, $R$, which the planning agent should be avoiding collisions with. Then, function $\text{Collision Check}$ generates a set of implementable, collision-free actions $A$. If $R \neq \emptyset$, the algorithm proceeds by planning a set of joint strategies $T$ (function $\text{Get Strategies}$) and then picking the minimizer of (22) (function $\text{Cost Optim}$). In case $R = \emptyset$, the the function $\text{Efficiency Optim}$ computes the most efficient action. The algorithm terminates when the boolean $\text{AtGoal}$ becomes true, indicating that the planning agent has reached its destination.

5.4.1 A Discussion of Complexity

The most computationally expensive part of the proposed algorithm is the computation of the set of joint strategies $T$. Obtaining $T$ involves determining the set of all compatible final permutations $P = \{p^d_1, \ldots, p^d_M\}$ that correspond to final system states with the planning agent at its destination. This set has cardinality $|P| = M = (n - 1)!$ and may be computed in time $O(n!)$. For each permutation $p^d_m$, we are computing:

1. A set of $K$ permutation paths $\Pi^m$ that connect the current system permutation $p_t$ with $p^d_m \in P$. For computing this path set, we employ an algorithm for finding K-shortest paths. Such algorithms typically make use of a shortest path algorithm, e.g. Dijkstra’s (Dijkstra 1959); therefore their complexity depends on the number of calls to the shortest path algorithm. Katoh’s algorithm (Katoh et al. 1982) appears to be the most efficient among them, with a runtime complexity of $O(K(|E| + |V|\log |V|))$, where $|V|$ and $|E|$ represent the number of nodes and edges in the graph respectively, which for a permutohedron of order $n$ are equal to $|V| =
2. A set of braids \( T^m \), consistent with each permutation path \( \pi^m \in \Pi^m \). For each permutation path in \( \Pi^m \), we derive \( 2^l \) different braids (where \( l \) is the number of edges in the path), by taking all possible permutations of consistent generator assignments on the permutation path edges. For a constant \( K \) and considering a maximum path length \( \frac{n(n-1)}{2} \), this computation runs in time \( O(2^{\frac{n(n-1)}{2}}) \) (worst case complexity).

**Algorithm 2** SCN\((D, Q, d, \Xi, At\_Goal, M)\)

**Input:**
- \( D \) — list of landmarks;
- \( Q \) — system state;
- \( d \) — planning agent’s intended destination;
- \( \Xi \) — state history of all agents;
- \( At\_Goal \) — boolean variable signifying arrival at agent’s destination;
- \( M \) — context.

**Output:**
- \( a \) — action selected for execution

1: while \( \neg At\_Goal \) do
2: \( R \leftarrow Get\_Reactive\_Agents(\Xi) \)
3: \( A \leftarrow Collision\_Check(\Xi, M, R) \)
4: if \( R \neq \emptyset \) then
5: \( T \leftarrow Get\_Strategies(d, Q, D, R) \)
6: \( a \leftarrow Cost\_Optim(A, T, d, \Xi, M) \)
7: else
8: \( a \leftarrow Efficiency\_Optim(A, d, M) \)
9: end if
10: end while
11: return \( a \)

The dominant term in the expression of the overall worst case complexity for computing the set of possible braids \( T^m \) for one permutation \( \pi^m \) is the exponential term \( O(n!) \). This implies that the complexity of the present algorithm does not scale well with large numbers of agents. Furthermore, considering that the aforementioned computation needs to run for all possible permutations in \( \mathcal{P} \), the complexity becomes \( O((|\mathcal{P}|)!) \). However, we argue that for our purposes, i.e., eventual deployment on a social robot navigating in real-world human environments, the following considerations may enable us to restrict ourselves to low \( n \) and \( |\mathcal{P}| \) and \( K \):

(a) A real robot has limited sensing capabilities, usually corresponding to a local radius of a few meters. Therefore, even in crowded environments, the surrounding agents will be considered as they enter the sensing radius and not universally.

(b) The rationality assumption for agents, which is supported by studies on human inference (Csibra and Gergely, 2007) and human navigation (Wolfinger, 1995). Rational agents aim at avoiding undesired divergences from the direction pointing towards their destinations (Moussaid et al., 2011). This implies that (i) humans end up following short permutation paths and (ii) a good final permutation prediction can be done by simply projecting forward agents’ current velocity on the boundary of the robot’s sensing radius (the robot does not need to know exactly where others are going). These observations motivate low \( K \) and low \( |\mathcal{P}| \) respectively.

(c) The rationality assumption also allows us to assume that once two agents pass each other, they stop reacting. For this reason, at replanning time, we restrict ourselves to considering only the number of agents that are ahead and are assumed to be observing the planning agent. Therefore, as the execution progresses and the robot approaches its destination, the number of reactive agents is expected to drop, allowing the robot to switch to efficient—and less computationally intense—execution.

(d) Our algorithm is based on frequent replanning. Plans have a short horizon but are made with a global reasoning (over joint strategies). The short planning horizon has been shown to be in compliance with human locomotion according to Carton et al. (2016a), who presented evidence that humans employ a shorter planning horizon as they navigate complex environments, to avoid collisions that could emerge from unexpected disturbances. The global planning horizon ensures that the motion of the robot will be consistent throughout the whole sequence of consecutive planning cycles.

For reference, in a game with 5 agents, a replanning cycle of SCN that generates 3 permutation paths per permutation runs at an average of \( \sim 185 ms \), with the worst case being \( \sim 402 ms \) in a non-optimized MatLab implementation on a MacBook Pro of 2015 with an Intel Core i7 processor of 2.5 GHz, running macOS Sierra. These times appear to be encouraging for real-time execution on a mobile robot platform, upon the transfer to a faster language and the appropriate code optimizations.

## 6 Evaluation

In this section, we present simulation results, demonstrating the benefits of our approach. Subsection 6.1 describes the experimental setup and provides implementation details, whereas subsection 6.2 presents
By the end of round 2, the agents have reached a system configuration corresponding to a clear consensus over a joint strategy.

By the end of round 2, the agents are in a system configuration that is about to lead to conflicting encounters.

Figure 10. A game with 3 agents in a $6 \times 6$ workspace. Fig. 10a and Fig. 10b depict partial executions of the same scenario (same start and end positions for all agents) with SCN and GREEDY respectively. The current system state is denoted with non-transparent system circles, whereas faded configurations correspond to configurations of past time steps.

results extracted by testing our algorithm under different settings.

6.1 Setup

We consider a setup where a set of $n$ agents navigate a discretized square workspace, partitioned into a set of $N_t^2$ tiles, where $N_t$ is the number of tiles per side (Fig. 10). Each agent $i \in \{1, \ldots, n\}$ starts from an initial tile $q_i$ and moves towards a final tile $d_i$. The game is played in rounds until all agents reach their destinations.

In order to assess our approach and demonstrate its benefits, we consider challenging game scenarios that reinforce intense encounters among agents. We do so by positioning agents on the sides of the workspace and having them navigate towards opposing sides. Each game scenario is sampled at random from the space of scenarios of size $N_t^n \times N_t^n$, corresponding to the number of distinct assignments of agents to initial and final configurations. At every round, the players simultaneously pick an action, which corresponds to a neighboring, unoccupied square. Forward, backward, left, right and diagonal, collision-free transitions are allowed. Since at planning time each agent has no access to the plans of others, in order to ensure collision avoidance, transitioning to a square that is adjacent to a square currently occupied by another agent is not allowed. Depending on the number of agents and the size of the workspace, this setup might result in deadlocks. In our evaluation, executions that result in deadlocks are discarded. However, note that the purpose of our evaluation is to study how agents behave when they have multiple actions available. We examine how the adoption of different strategies in action selection may affect the evolution of the game qualitatively and quantitatively.

Algorithm 3 GREEDY($Q, d, \Xi, AtGoal, M$)

Input: $Q$ – system state; $d$ – agent’s destination; $\Xi$ – state history of all agents; $At\_Goal$ – boolean variable signifying arrival at agent’s destination; $M$ – context.

Output: $a$ – action selected for execution

1: while $\neg At\_Goal$ do
2: $A \leftarrow \text{Collision\_Check}(Q, M)$
3: $a \leftarrow \text{Efficiency\_Optim}(A, Q, d, M)$
4: end while
5: return $a$

To demonstrate the importance of incorporating a topological understanding into agents’ decision making process, we compare the performance of our algorithm (SCN) against a greedy baseline (see algorithm 3) that plans actions, greedily seeking to maximize its efficiency (the progress to the agent’s destination) at every round. The GREEDY algorithm makes use of the same collision checking function as SCN. Their main difference lies in how they select an action when multiple of them are available. Considering a homogeneous setup ($n$ agents running SCN versus $n$ agents running GREEDY), we show that explicitly reasoning about the emerging joint strategy
Average Entropy profile per agent per experiment over 200 scenarios involving 3 agents. On average, SCN agents reached a consensus over a joint strategy faster.

Average Efficiency profile per agent per experiment over 200 scenarios involving 3 agents. On average, SCN agents made greater progress towards their destinations per round.

Figure 11. Comparative diagrams, generated upon running 200 experiments with 3 agents in a workspace of size $6 \times 6$. The red curves correspond to agents running SCN and the blue curves to agents running GREEDY. The compromise between Efficiency and Consensus was set to $\lambda = 0.2$ and the number of paths per permutation to 3. Student’s t-tests were performed on all rounds to determine the statistical significance of the profiles difference. The symbols *, ** and *** denote rounds on which the difference in the performance between SCN and GREEDY was found to be significant to a degree described by $p$-values < 0.05, 0.01 and 0.001 respectively, according to a paired Student’s t-test. Due to space constraints, the significance symbols were place vertically.

at planning time, benefits everyone in the scene, as it leads to a faster uncertainty decrease that simplifies everyone’s decision making. Note that despite the different braid convention that each agent is making, they still manage to converge to a mutually beneficial consensus on a joint strategy of avoidance much faster than GREEDY agents. Qualitatively, our algorithm leads to less ambiguous system configurations, which result in higher average progress per round and lower average time to destination.

6.2 Simulation Results

In this section, we present the behavior that our algorithm generates and demonstrate its benefits by comparing its performance with the performance of the GREEDY baseline.

6.2.1 Qualitative Behavior

Fig. 10 depicts partial executions, after 2 rounds, of a scenario from a game involving 3 agents. Fig. 10a depicts the play of agents running SCN, whereas 10b shows the corresponding play of the agents running GREEDY. It can be observed that the agents running our algorithm have led the game to a configuration that is more beneficial for everyone, as all of their encounters are essentially resolved by the end of the second round. This was achieved by planning informative actions that rapidly led to a significant entropy decrease and accelerated convergence to a consensus over a joint strategy. On the contrary, the agents running the baseline, having initialized their game by focusing on efficiency, are now reacting suboptimally to their constrained action spaces. A video presentation, demonstrating the concept of our approach as well as our system in action can be found at https://youtu.be/ge9fRI4eav4.

6.2.2 Performance

Fig. 11 depicts comparative performance diagrams, derived upon running 200 randomly sampled game scenarios involving 3 agents navigating a small workspace of size $6 \times 6$. The quality of agents’ decision making is illustrated in the profiles of average entropy and average progress to destination, depicted in Fig. 11a and Fig. 11b respectively. It can be observed that the systems of agents running SCN achieve faster entropy reduction and higher average progress towards destinations, compared to the systems running the baseline, with statistical significance noted in the diagrams. Fig. 13a depicts comparative plots of average time to destination (left) and average time to “get free”, which corresponds to the time an agent first has
full control over the scene evolution, i.e., the first time when no other agents are ahead.

Similar comments can be made for the case of 4 agents, navigating a workspace of the same size 6×6. Fig. 12 presents comparative performance diagrams for entropy and progress to destination, whereas Fig. 13b depicts a comparison of average time to destination and average time to get free.

For the simulated examples presented, each agent models joint strategies as braids, defined with respect to a projection line that is parallel to its starting side. For the case of the GREEDY agents, entropy was evaluated by employing the inference mechanism of the agents running SCN, i.e., we computed what their belief would be if they had access to the inference mechanism of SCN agents. To ensure proper comparison, the same set of braids was considered at each time step for the same agent in both setups. The weighting factor $\lambda$ was set to 0.2, as it was found experimentally to lead to a desired compromise between efficiency and consensus. For deriving multiple candidate paths in the permutation graph, we use the algorithm of Yen (1971) for finding $K$ shortest paths.

7 Discussion and Future Work

We considered the problem of decision making in a navigation scenario involving multiple rational and cooperative agents that do not explicitly communicate with each other. In such a scenario, the uncertainty over the exact strategies of other agents make it hard for an agent to predict their behaviors and thus to make safe and socially compliant decisions over its own actions. These settings may be found in a variety of real world application, such as robotic navigation in crowded human environments.

To address this problem, we presented an online planning framework, inspired by the insights of recent studies on the cooperative nature of pedestrian behavior (Wolfinger 1995) and the goal-directed inference of humans (Csibra and Gergely 2007). Our framework explicitly incorporates the concept of cooperation by modeling multi-agent collective behaviors as topological global joint strategies, using the formalism of braids (Birman 1975). Our topological model forms the basis of an inference mechanism that associates observed behaviors with future collective topologies. In the decision making stage, each agent decides on an action that corresponds to a compromise between its personal efficiency (progress towards
destination) and a form of joint efficiency (the status of a consensus on a joint strategy of avoidance). To clearly showcase the benefits of our decision making concept, we deliberately studied a simplified version of the real world problem by considering an abstract, discrete setup, involving artificial agents playing a cooperative game. Extensive trials over randomly-generated, challenging scenarios demonstrated the benefits of reasoning about joint strategies over a baseline that greedily prescribed actions of high efficiency. Our algorithm was shown to lead to a faster decrease of uncertainty regarding the scene evolution, which resulted in efficiency increase and lower execution times with high statistical significance.

Our purpose in this paper was to demonstrate the benefits of a topologically-based decision making mechanism for dynamic multi-agent environments. Future work involves modifications of the proposed approach towards enabling online, sensor-based, real-world operation in unstructured environments. We contributed a first step towards that direction by developing a data-driven approximation of our inference mechanism with a deep neural model, learned from demonstrations of simulated, challenging, multi-agent scenarios in continuous settings (Mavrogiannis et al. 2017); a comparative evaluation against the Social Force model (Helbing and Molnár 1995) demonstrated promising results in simulation. Another step towards approaching more realistic execution settings was done with our Social Momentum motion planner (Mavrogiannis et al. 2018), which also follows the principles of topological braid theory to generate legible behaviors in multi-agent environments; this approach was also shown to outperform Social Force and ORCA (van den Berg et al. 2009) in terms of intent-expressiveness, social compliance and topological complexity of executions.

Ongoing work involves learning a data-driven model of our inference mechanism from human trajectory datasets (e.g. (Brščić et al. 2013)) that would enable agents to form a humanlike belief over the future behavior of systems of multiple agents. We are also in the process of planning an extensive user study that will measure the effects of the behaviors generated by our planning architecture on human subjects in a controlled lab experiment.

Notes
1. This specific convention is not particularly important, as long as one is consistent. There are works that use the inverse convention when defining the positive and negative generator exponents. Our selection facilitates the exposition of further concepts in the remainder of the paper.
2. A transposition can be described as a permutation involving exactly one swap of a pair of elements. An adjacent transposition is a transposition involving an exchange of two adjacent elements. An adjacent transposition implementing an exchange of the elements with order \( j \) and \( j + 1 \), with \( 1 \leq j < n - 1 \) in a list of \( n \) elements, is commonly denoted as \( \beta_j = [j \ j + 1] \).

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grants IIS-1526035 and IIS-1563705. We are grateful for this support.

References


