On Legible and Predictable Robot Navigation in Multi-Agent Environments

by

Jean-Luc Bastarache

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2022

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Legibility has recently become an important property to consider in the design of social navigation planners. Legible motion is intent-expressive, which when employed during social robot navigation, allows others to quickly infer the intended avoidance strategy. Predictability, although less commonly studied for social navigation, is, in a sense, the dual notion of legibility, and should also be accounted for in order to promote efficient motions. Predictable motion matches an observer's expectation which, during navigation, allows others to confidently carryout the interaction. In this work, we present a navigation framework capable of reasoning on its legibility and predictability with respect to dynamic interactions, e.g., a passing side. Our approach generalizes the previously formalized notions of legibility and predictability by allowing dynamic goal regions in order to navigate in dynamic environments. This generalization also allows us to quantitatively evaluate the legibility and the predictability of trajectories with respect to navigation interactions. Our approach is shown to promote legible behavior in ambiguous scenarios and predictable behavior in unambiguous scenarios. We also provide an adaptation to the multi-agent case, allowing the robot to reason on its legibility and predictability with respect to multiple interactions simultaneously. This adaptation promotes behaviors that are not illegible to other agents in the environment. In simulation, this is shown to resolve scenarios of high-complexity in an efficient manner. Furthermore, our approach yields an increase in safety while remaining competitive in terms of goal-efficiency when compared to other robot navigation planners in randomly generated multi-agent environments. The code of this work will be made publicly available[1].

[1] https://github.com/jlbas/LPSNav

Acknowledgements

I would like to express my deepest gratitude to my supervisors Christopher Nielsen and Stephen L. Smith. They have provided invaluable insight and support throughout my studies and I am very fortunate to have had the opportunity of working with them. I will continue to cherish our weekly discussions as they've been very enjoyable and thought provoking. Seeing their level of dedication and strong work ethic has been inspirational and something I aspire to attain.

I would also like to thank my loving parents Anne and Roger for always showing support, whatever challenge I decide to take on. They taught me to be appreciative of what I have, something I remind myself daily which helped maintain a positive mindset over the years. I am also thankful for my younger sister Sophie, who is never scared to challenge me and who I strive to be a better role model for.

Lastly, I thank my partner Renée for joining me on this journey. Our adventures together have kept me sane along the way.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Without having to explicitly communicate their intentions, humans are able to seemingly effortlessly navigate amongst one another in a collision-free manner. Even in open social spaces devoid of navigation rules or underlying structure, they can safely make continued progress towards their goals. This can largely be attributed to their ability to infer others' interaction strategies, as depicted in Figure 1.1. This inference allows each human to adapt their trajectory in order to avoid a collision. Ultimately, this cooperation reduces the combined effort spent by each agent while deviating from their initially planned trajectory to avoid a collision, and in turn, increases the navigation efficiency.

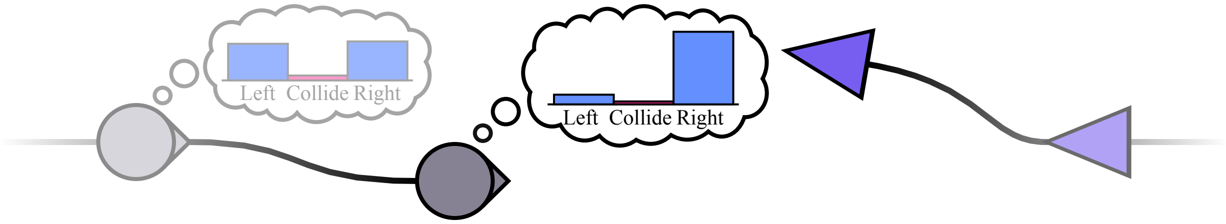


Figure 1.1: *A human (gray) inferring a robot's (purple) avoidance strategy.*

When constructing their navigation strategies, human's are likely to prioritize nearby humans, where there is a more immediate risk for collision. Nonetheless, during each interaction, they will attempt to indicate its strategy to the other as early as possible in hopes of reducing the uncertainty about its intended passing side. The degree to which these indications are made clear vary depending on the complexity of the environment. For example, one might infer a passing side from something as subtle as an other's shift in eye gaze or something as obvious as a prominent adjustments in direction. From an early age, we learn to identify and employ these motion queues to aid our navigation in social spaces.

Social navigation is the task of a social agent, either a human or a robot, navigating in a shared environment with other social agents to reach its goal. Although the goals are unknown to each other, social agents are aware that others in the environment are also solving their own navigation problems. In social navigation, the agents adapt their motions by taking into account those of the others in order to respect social conventions (e.g. personal space). This differs from traditional navigation in that an agent is not represented as a dynamic obstacle, but rather a social entity.

Recently, there has been an increasing interest for the deployment of autonomous mobile robots into social environments. However, the complexities of social navigation continue to prevent state-of-the-art frameworks from being seamlessly deployed into the real world. Many existing approaches take inspiration from humans' navigation strategies in hopes of generating human-like motion. Since robots typically don't exhibit the same motion queues as a human, this begs the question as to how the robot should best make use of its motion when navigating a dynamic multi-agent environment? We focus on the implicit communication of intention through motion, rather than an explicit modality such as signal lights, since changing the speed and heading is fundamental to all mobile robots.

This thesis analyzes and discusses how a mobile robot should make use of the shape and time parameterization of its trajectory in order to indicate its avoidance strategy during navigation interactions with other agents. A more formal problem formulation is given in Section 3.1. We generalize existing properties of motion which arise from the inference of trajectories and goals that were shown to facilitate human-robot interaction (HRI) tasks to the problem of social navigation. The presented navigation framework allows the robot to reason on the degree to which it should seek to indicate its intention to surrounding agents.

1.1 Literature Review

Although autonomous robots have traditionally operated in controlled and human secluded environments, recent advancements have given rise to their deployment into social spaces. There is an increasing interest for their use in warehouses, in the healthcare sector, for search and rescue, for deliveries, for surveillance, in the workplace and other social services [69]. To motivate the social navigation task, the following literature review begins with an overview of human locomotion studies. These studies seek to explain and model the strategies taken by humans when navigating a social space. Afterwards, existing social robot navigation frameworks, some of which have taken inspiration from the mentioned human locomotion studies, will be discussed. This will lead to the current state of the field, where we will briefly introduce important properties of motion in which modern frameworks have found to be important to consider for social navigation.

1.1.1 Social Navigation

Pedestrian motion is inherently complex and continues to be a focal point of interest amongst autonomous robot researchers. During locomotion, humans make use of numerous motion cues such as gaze [73] and head movement [45] to avoid collisions. It remains uncertain as to how they construct their anticipatory strategies while navigating dynamic environments. The concepts of *personal space*, originally proposed in the theory of proxemics [40], describes a physical region in which people seek to maintain around themselves. The size of the space is non constant and was found to depend on demographic factors such as cultural background [40], ethnic group [9], age [9] and gender [2]. In [39], it is shown that the personal space takes the shape of an ellipse and its size depends on environmental factors such as the speed and certainty in others' motion. They also found that anticipation and preplanning are involved in the navigation task, as participants were shown to exhibit early gate adaptation and maintain constant clearances around obstacles. Furthermore, in North America, a tendency to walk on the right was found [98, 11]. People also tend to walk in a way that minimizes their energy expenditure [66]. One such strategy is to maintain a constant speed while navigating [89].

Many existing studies have focused on determining the collision avoidance strategies employed by humans during locomotion [23, 8, 46]. These studies seek to find at which point and to what extent a human on a collision course with another human adapts their motion. In [23], participants were asked to avoid a collision with an oncoming human doll, fixated to a guy wire and travelling at a constant velocity regulated by a stepper motor. For different approach velocities, they found that the participants adapted their motion at the same time. However, the rate of change of avoidance maneuver's lateral motion was found to be correlated to the approach velocity of the object. In [8], they examined the collision avoidance strategy for a subject on a collision course with an interferer travelling orthogonally on a collision course. The interferer was specifically instructed not to react to the subject. The avoidance strategy was similarly studied between a human and non-reactive interferer in [46]. There, they evaluated the effect that the angle between the subjects and the speed of the interferer had on the human's avoidance strategy. Although these studies seek to determine human collision avoidance strategies during locomotion, they do not take into account collaboration in the interaction.

More recently, it was found that humans collaborate during a collision avoidance task [75, 74]. In these studies, when instructed to avoid a collision with one another, the participants shared the navigation load by reorienting their path and adapting their speed. In [75], they introduce a collision avoidance metric, which tracks the minimal predicted distance (MPD) during an interaction between walkers. The pedestrians were found to adapt their motion only when it is required, that is, when the MPD drops below a threshold of 1 meter. In [74], they showed that the navigation load is not equally shared between pedestrians, but rather that it is role-dependent. In fact, the participant giving way was shown to contribute more to the collision avoidance task than the one crossing first. The contribution of each agent is computed by integrating the effect of its motion adaptations to the MPD over the course of the interaction. The first to cross was found to only adjust their speed, whereas the human giving way adjusted both their speed and path.

1.1.2 Social Robot Navigation

This section briefly reviews the progression of social robot navigation from early works, which modelled humans as non-reactive obstacles, to state-of-the-art human inspired approaches which explicitly model cooperation in the interaction.

Modeling Humans as Non-Reactive Obstacles

Early works in autonomous robot navigation considered humans to be non-reactive obstacles, and did not attempt to model pedestrian behavior [34]. These approaches, termed reaction-based, relied on frequent re-planning in order to capture the other agents' dynamics. However, as they do not consider the human's cooperation, these approaches often suffered from oscillatory behaviors [32]. Some approaches have employed existing pedestrian behavior models, such as the social force model [41](SFM). This potential field-based approach models human behavior as a sum of attractive and repulsive forces, which represent the human's motivation to reach its goal while avoiding obstacles and other humans in the environment. In [31], the robot models social interactions using the SFM to navigate crowded environments. The SFM is considered to perform well when simulating crowds, which has useful applications when analyzing crowd evacuation dynamics. However, this model was shown to perform poorly when predicting individual motion, especially for evasive maneuvers [52].

Other reaction-based approaches have focused on multi-robot collision avoidance [96, 95, 87]. These approaches, categorized as reciprocal collision avoidance, build on the idea of velocity obstacles [32] to guarantee collision free motion, assuming all agents are homogeneous, that is they follow the same policy. Their extension in [87] improves on the previous approaches by explicitly taking into account that other robots may also change their trajectories based on their surroundings. As a result, the multi-robot navigation is guaranteed to be oscillation free in addition to being collision free. However, the homogeneity assumption is not well suited for the deployment into human-shared environments.

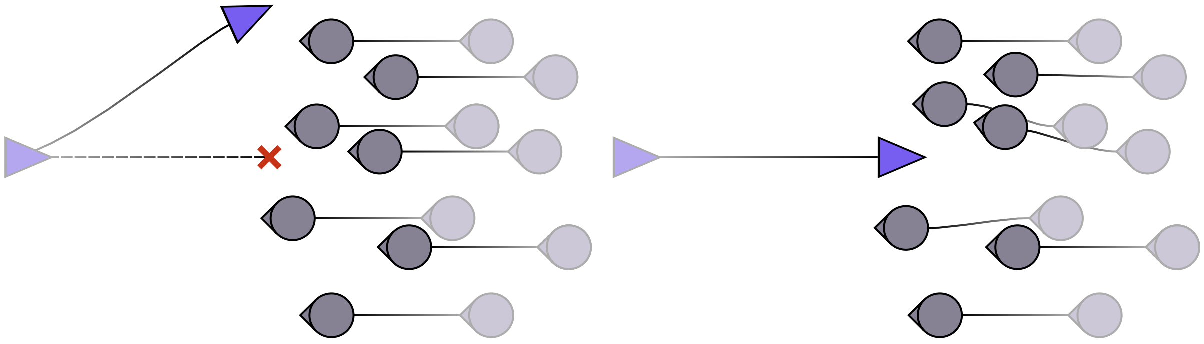
Human Inspired Social Navigation

Recently, research in autonomous robot navigation has taken inspiration from pedestrians' navigation abilities. These approaches seek to generate motion that respect social conventions as followed by humans. In [49], they present a social navigation framework based on proxemics theory [40] and passing side convention [11]. A human-like behavior model is developed in [90] which takes into account other pedestrians' intentions. Their model is based on the SFM, where the virtual forces affecting the pedestrian's motion are based on a dynamically set subgoal and surrounding obstacles. To evaluate the performance of their motion planner, they compute the difference between its generated motion and that which was observed by a human participant. The trajectories generated by the navigation framework presented in [38] are said to be *human-friendly*, in that they can intuitively be predicted by humans. Their approach makes use of the same heuristics as employed by humans during collision avoidance.

Learning based approaches have focused on extracting navigation heuristics while observing human trajectories from pedestrian datasets [47, 78]. In [99], they present an inverse reinforcement learning approach for robot navigation in an office environment. Their approach employs the principle of maximum entropy to obtain a probabilistic model of human behavior. This is learned on a large dataset of example human trajectories. Given the human probabilistic model, their approach seeks to minimally disrupt the human's motion, given by the prediction model. During experiments, their algorithm is shown to be able to generalize to entirely new environments. In [59], human relative motion is learned from a dataset and used to compute a dynamic cost map which is used by a robot to plan socially acceptable motion. They evaluate their approach against human behavior to see how well it can generalize to unseen human motion.

Modeling Cooperation During Navigation Interactions

A key constituent in human navigation is their ability to infer others' interaction intentions, e.g., to which side are they trying to pass. In crowded environments, this inference allows them to cooperate in the interaction, enabling efficient navigation. Early robot navigation frameworks often overlooked or did not attempt to model these aspects, resulting in undesirable behaviors such as oscillations [1] or the freezing robot problem [93]. The freezing robot problem arises in complex dynamic environments when the motion planner deems all paths to be unsafe, causing the robot to freeze in place. This occurs frequently if the planner does not account for cooperation from other agents in the environment (Figure 1.2 (a)). In fact, it is often the case that nearby agents would be willing to make way for the robot (Figure 1.2 (b)). By accounting for cooperation, many paths are no longer considered unsafe, and the robot can continue to make progress in the environment.



(a) Planned path without cooperation

(b) Planned path with cooperation

Figure 1.2: *Freezing robot problem where many paths are deemed unsafe since the robot does not account for the other agents' cooperation (a), despite the fact that many are willing to adapt their trajectories (b).*

Due to the myopic nature of non-cooperative approaches, recent research in social robot navigation has focused on modeling coupled interactions. Such approaches have sought to model social robot navigation as a cooperative collision avoidance task. To overcome the freezing robot problem, in [92], they develop interacting Gaussian processes which explicitly capture cooperative collision avoidance. They validate their approach by performing many runs accross a dense human crowd. By comparing against existing non-cooperative approaches, they found that a cooperation model is critical for safe robot navigation in complex social environments.

In [64], the authors use topological braids to encode agents' joint behaviors. They show that their algorithm more rapidly decreases the uncertainty of the emerging avoidance strategy in the workspace. To demonstrate the usefulness of their concept, they consider a discrete cooperative game representing a simplified version of the real-world problem. A non-cooperative game-theoretic approach is proposed in [94], where human-like motion is generated by modeling the decision making process using Nash equilibria. Recently, deep reinforcement learning (DRL) approaches have demonstrated promising results [28, 19]. These approaches train policies that implicitly encode the agents' models and interaction intentions. However, complex learning-based approaches make it difficult to extract the social strategies being employed during navigation.

Legible and Predictable Motion for Social Robot Navigation

Since mobile robots are not equipped with the same motion cues as humans (e.g. gaze and head movement), humans have much more difficulty in inferring their intentions. Studies have shown that humans are more conservative (i.e. give a larger clearance) when avoiding moving inanimate objects [88] or objects with fixed limbs [61] than they are when avoiding humans. Legibility has become an important property of motion to consider in human-robot interactions (HRI) [85]. This property is used to describe intent-expressive motion, that is, motion in which the robot's goal or intent can quickly and confidently be inferred by a human observer [24].

Early works claiming to generate legible motion often did so indirectly by targeting related properties, such as [10]. In [27], the authors directly use legibility as an optimization criterion to generate legible motion. However, since they only consider stationary goal points in the workspace, their metrics cannot be directly applied to locomotion applications. Although largely studied for HRI, legibility has recently become an important property to consider in the design of social navigation planners. During a user study, humans were found to prefer larger signaling distances, i.e., the distance at which the robot initiates the avoidance maneuver [76]. Furthermore, in a human-robot crossing scenario, legible navigation was shown to increase the perceived safety [57]. In [51], a cost function is formulated that takes into account social and context dependent costs. The authors show that their planner is able to generate legible motion with respect to the robot's underlying goal in the workspace.

Many navigation frameworks seeking to produce legible motion do so by considering the robot's legibility with respect to its global goal pose [38, 51, 55]. However, in dynamic and crowded environments, humans are in many cases unaware of others' global goals and adjust their trajectories based on the inferred avoidance strategy (e.g. the passing side).

In [17], legible motion was shown to reduce the planning effort in a locomotion setting. A social navigation planner which considers its legibility alongside the observer's viewpoint is proposed in [91]. In a simulated restaurant setting, the robot seeks to indicate its goal using the legibility objective from [24] and [72] while remaining in the observer's field of view. Their method is effective at increasing the legibility for observers with a limited view of the restaurant. However, tailoring the path to be legible to certain participants made it less clear to others. Future research is required to determine how legibility should be optimized for multiple observers at different view points.

Motion cues for a doorway yielding maneuver between a robot and human are specifically studied in [44]. They use a broader definition for the legibility of robot behavior from [57]. They compare five yielding cues: stop, decelerate, retreat, tilt and nudge. In an online user study, they found that the retreating maneuver was the most effective at being correctly interpreted with the highest confidence. Nonverbal legibility cues are studied in [43], where they compare the use of projected arrows and flashing lights on the ground to communicate goal information or path information or both. Their findings suggest that arrows should communicate path information and flashing lights should communicate goal information. Although they found projected arrows to be communicative in both scenarios, flashing lights were only communicative for goal information.

The combination of explicit and implicit robot-human communication is analysed for social navigation in [18]. A haptic feedback device is worn by the human, with which the robot can modulate the vibration pattern to indicate robot priority or human priority. They found that users were more confused when the robot only used implicit communication, especially in robot priority tasks. Using a combination of explicit and implicit communication allowed the user to understand the robot's intention 92% of the time. They showed that this proactive communication method reduced the users' effort while navigating, as indicated by a shorter path length. Due to certain assumptions in their work, the planner is unable to deal with certain scenarios, such as when the human suddenly stops. The authors also suggest that their planner could benefit from a sampling-based method since their current optimization is computationally expensive.

A simplified legibility objective is proposed in [67] which enables the use of optimal control algorithms. They use a simpler observer model than what was originally used for the derivations of the legibility metric [24]. The simplified model avoids having to evaluate the partition function, which is an integral over an infinite dimensional space. Previously, a quadratic cost function was assumed for the observer's model of the robot in order to simplify the computation. Since this assumption is no longer needed, they show that their simplification allows the optimization of nonlinear robot dynamics at an equivalent computational complexity.

An interactive model predictive control robot navigation framework for dense crowds is proposed in [21]. They propose a pedestrian motion model with which the robot predicts their intentions and interactions in the crowd to find an optimal trajectory. In simulation and in real-world scenarios, their approach generates more gentle variations in angular velocity, which the authors claim improves the legibility.

Legibility has also been considered in the interaction between a single agent and a multi-robot system. In [16], they consider the trajectory, dispersion and stiffness of a group of mobile robots to communicate their intentions to a user. Through a user study conducted in a virtual environment, they found the trajectory of the group to affect the correctness of communication, whereas the dispersion and the stiffness affected the time it took to correctly infer the robots' intentions. Their study is extended in [15], where they consider the same three motion-variables on the legibility of multiple robot groups. In contrast to their first study, the trajectory and dispersion affected the correctness of communication, whereas only the stiffness affected the response time. The legibility and glanceability of a multi-robot system is considered in [48]. They introduce glanceability as a measure of an observer's ability to infer the robots' intentions pre-attentively. In an online user study, the collective behavior-based motion was the most legible, whereas the trajectory-based motion was the most glanceable.

In a multi-agent environment, *Social Momentum* makes use of topological braid theory to generate legible motion [65]. They use the angular momentum between two agents as an approximation for the navigation interaction dynamics. Their approach enable topological reasoning about the interaction while having a lower computational cost than their work which directly computed the topological braids [64]. Their planner's optimization scheme trades off between the legibility of the interaction and the robot's progress to its goal. To evaluate their framework, they measure the topological complexity of the trajectories as a proxy for legibility. Our work is more closely related to this approach, since it explicitly reasons about the emerging collision avoidance strategy, rather than the global goal within the workspace. However, rather than directly optimizing the progress to goal, we suggest that the robot should instead optimize over its predictability with respect to the navigation interaction. Predictability, as it was defined in [24], measures the degree to which motion matches an observer's expectation. However, in their work, it is defined as a property which depends on a complete trajectory, and as such, is not well suited for motion planning applications. In this work, we generalize predictability to partial trajectories, allowing it to be considered for navigation interactions.

When comparing different works that use these properties of motion, one might find slight inconsistencies across the various definitions for legibility and predictability [71, 86, 4, 50, 56]. For example, in [55], legibility is used to describe robot locomotion that is both intent expressive and matching expectation. Legibility has also previously been assumed to be a consequence of predictable motion [10]. In this paper, we refer to legibility and predictability as formalized in [24], where they propose mathematical models for their inferences and find them to be fundamentally different and sometimes contradicting properties of motion.

1.2 Contributions

The main contribution of this work is to generalize legibility and predictability as formalized in [24] to social navigation. Existing navigation frameworks claiming to generate legible motion have done so by evaluating their approach through the use of user studies. Until now, there has not been a quantitative method to evaluate the legibility or predictability of a robot with respect to navigation interactions. Thus, our contributions are fourfold. First, we present an approach to explicitly model navigation interactions as dynamic goal regions. Second, we generalize legibility and predictability for static goal points as formalized in [24] to dynamic goal regions. Third, we propose an approximation of the predictability score for partial trajectories discussed in [33], making it suitable for real-time planners. Lastly, we propose a navigation planner that is capable of optimizing over the legibility and the predictability of its motion with respect to navigation interactions.

1.3 Notation

In this work, scalars are denoted in lowercase, x, vectors in bold lower case, $\boldsymbol{x}$, matrices in upper case, X, sets in calligraphic, $\mathcal{X}$, and collections in bold calligraphic, $\boldsymbol{\mathcal{X}}$. The symbol $:=$ denotes equal by definition and the set of non-negative real numbers is denoted by $\mathbb{R}_{\geq 0}$. If $a, b \in \mathbb{R}$, then $[a, b] := \{x \in \mathbb{R} : a \leq x \leq b\}$. If $x \in \mathbb{R}^n$, then $\|x\|$ denotes the Euclidean norm. The workspace of a robot is denoted by $\mathcal{W} \subseteq \mathbb{R}^n$, where $n \in \mathbb{N}$ depends on the type of robot and the application. A workspace trajectory, or trajectory for short, of the robot is a parameterized curve $\xi \colon [t_s, t_f] \subset \mathbb{R} \to \mathcal{W}$, where $t_s < t_f$, t_s denotes the start of an interaction while t_f is the end of the interaction (to be defined below). The set of all workspace trajectories is denoted by $\mathcal{T}$ and, for notational simplicity, we denote by $\xi_{\boldsymbol{a} \to \boldsymbol{b}}$ the trajectory along ξ from the configuration $\boldsymbol{p}_R(t_a)$ to $\boldsymbol{p}_R(t_b)$, where $t_s \leq t_a < t_b \leq t_f$, and $\boldsymbol{a} = \boldsymbol{p}_R(t_a)$ and $\boldsymbol{b} = \boldsymbol{p}_R(t_b)$. For consistency, when referring to a position along a trajectory, we choose the notation $\boldsymbol{p}_R(t_a)$ as opposed to $\xi(t_a)$.

Chapter 2

Preliminaries

In this chapter, we review the existing notions of legibility and predictability for static goal points in a robot's workspace. First, to get an intuitive understanding of these properties, an illustrative example of a simplified scenario is provided. Second, the mathematical derivations used to model legibility and predictability are reviewed.

2.1 Legible Motion

Let $\mathcal{G} \subset \mathcal{W}$ be a finite set of possible goals and let $\mathbf{g}^\star \in \mathcal{G}$ denote the robot's true goal.

Definition 2.1.1 (Legible Motion [24])**.** The motion generated by the partial trajectory $\xi_{\boldsymbol{s} \to \boldsymbol{t}}$ from the starting position to the current position is said to be legible when an observer can quickly and confidently infer the robot's true goal $\boldsymbol{g}^\star \in \mathcal{G}$.

To get an intuitive understanding of legibility, let us consider a simplified scenario where $\mathcal{G}$ is uniquely comprised of two distinct static goals, $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$. The robot's true goal is an element $\boldsymbol{g}^\star \in \mathcal{G}$ which the robot is trying to reach. An example legible trajectory for the case of the robot's true goal being $\boldsymbol{g}_2$ is shown in Figure 2.1. We notice that the trajectory exhibits an aggressive deviation towards $\boldsymbol{g}_2$ very early. Since it was assumed that the robot's only possible goals are $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$, an observer that knows $\mathcal{G}$ would quickly infer the true goal $\boldsymbol{g}_2$, thus making the motion legible.

More formally, legible motion is modeled by the goal inference function

$$f_L \colon \mathcal{T} \to \mathcal{G}, \tag{2.1}$$

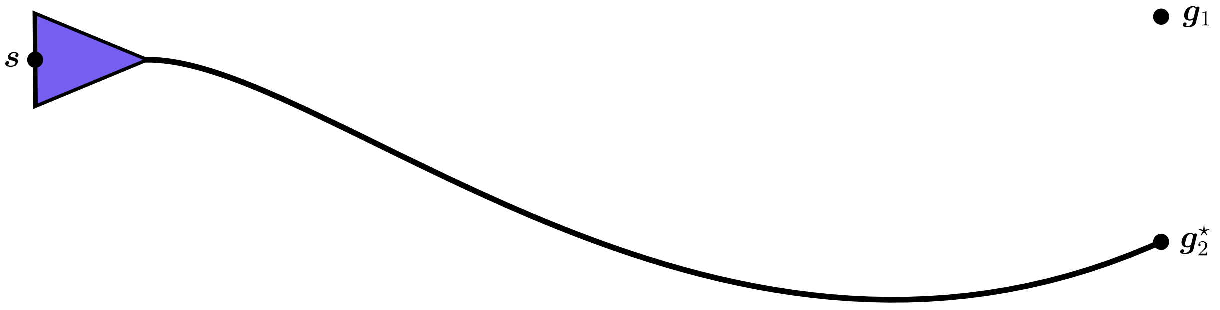


Figure 2.1: *Example legible trajectory with respect to $\boldsymbol{g}_2^\star$, assuming $\boldsymbol{g}_1$ and $\boldsymbol{g}_2^\star$ are the robot's only goals in the workspace.*

mapping trajectories to goals, and $\xi_{\boldsymbol{s}\to\boldsymbol{t}}$ is legible at time $t \leq t_f$ if

$$f_L(\xi_{\boldsymbol{s}\to\boldsymbol{t}}) = \boldsymbol{g}^\star. \tag{2.2}$$

The earliest t for which this is true is a measure of how legible the trajectory $\xi_{\boldsymbol{s}\to\boldsymbol{f}}$ is. As an example, let us consider the five trajectories reaching the robot's true goal $\boldsymbol{g}_2$ in Figure 2.2. Each endpoint of the partial trajectories $\xi_{\boldsymbol{s}\to\boldsymbol{t}_i}$, $i \in \{1, 2, \ldots, 5\}$, $t_i < t_f$, depict the positions along the trajectory to which an observer correctly inferred the robot's true

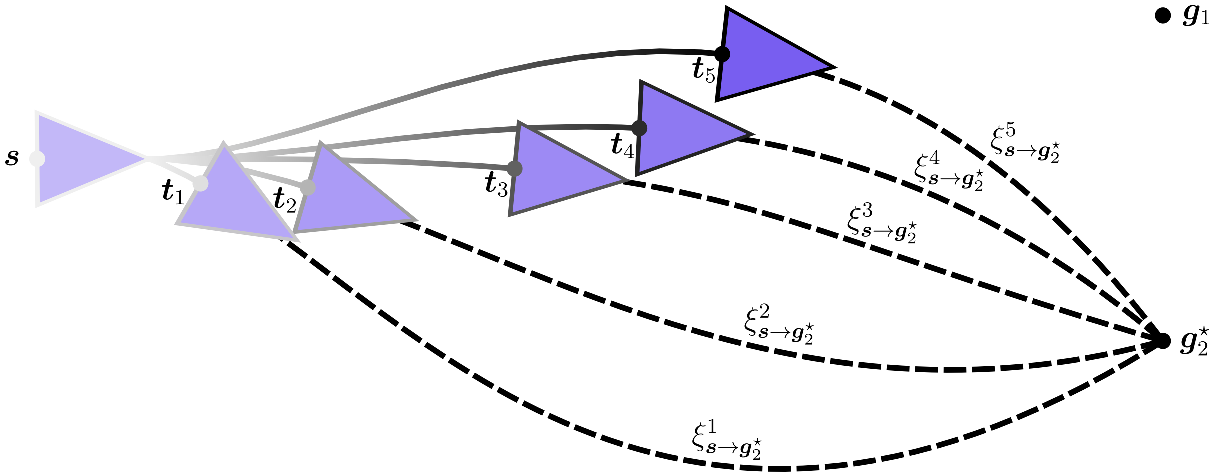

$\boldsymbol{g}_1$
$\boldsymbol{t}_5$
$\boldsymbol{s}$
$\boldsymbol{t}_4$
$\boldsymbol{t}_1$
$\boldsymbol{t}_2$
$\boldsymbol{t}_3$
$\xi^5_{\boldsymbol{s}\to\boldsymbol{g}_2^\star}$
$\xi^4_{\boldsymbol{s}\to\boldsymbol{g}_2^\star}$
$\xi^3_{\boldsymbol{s}\to\boldsymbol{g}_2^\star}$
$\xi^2_{\boldsymbol{s}\to\boldsymbol{g}_2^\star}$
$\xi^1_{\boldsymbol{s}\to\boldsymbol{g}_2^\star}$
$\boldsymbol{g}_2^\star$

Figure 2.2: *Trajectories of decreasing legibility from bottom to top, assuming $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$ are the robot's only goals in the workspace and $\boldsymbol{g}_2$ is the robot's true goal. Here, the time taken to correctly infer the goal increases with the indices, indicating that the trajectories are becoming less legible.*

goal, i.e. $f_L(\xi_{\boldsymbol{s}\to\boldsymbol{t}_i}) = \boldsymbol{g}_2$. Assuming the robot is travelling at constant speed, the times corresponding to these positions are increasing with the index i. Therefore, the legibility of the trajectories is decreasing with the index i. Considering the five trajectories from $\boldsymbol{s}$ to $\boldsymbol{g}_2$, $\xi^1_{\boldsymbol{s}\to\boldsymbol{g}_2}$ is the most legible and $\xi^5_{\boldsymbol{s}\to\boldsymbol{g}_2}$ is the least legible.

2.2 Predictable Motion

Conversely, predictability assumes the true goal is known to the observer. It relates to how closely the motion matches what the observer expects.

Definition 2.2.1 (Predictable Motion [24])**.** Given the true goal $\boldsymbol{g}^\star \in \mathcal{G}$, the motion resulting from the entire trajectory $\xi_{\boldsymbol{s}\to\boldsymbol{g}^\star}$ from $\boldsymbol{s}$ to $\boldsymbol{g}^\star$ is said to be predictable if it matches the observer's inference.

Let us consider a simplified scenario where the robot has a single true goal $\boldsymbol{g}^\star \in \mathcal{G}$. Prior to have started moving, it is also assumed that the robot's true goal is known to the observer. There are many possible trajectories the robot could take to reach $\boldsymbol{g}^\star$. However, as previously mentioned, predictability depends on what the observer expects. This expectation will therefore be specific to the individual and vary based on their familiarity and prior experiences with robotic tasks. An example inferred trajectory for an observer expecting the robot to be a rational agent seeking to navigate efficiently is shown in Figure 2.3

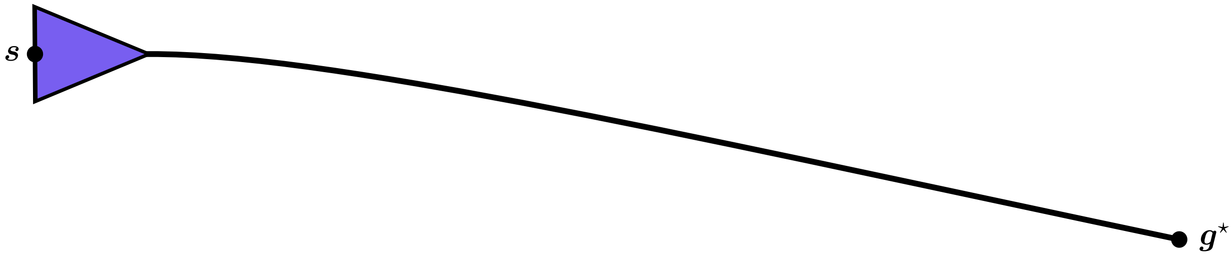


Figure 2.3: *Example inferred trajectory matching an observer's expectation.*

More formally, predictable motion is modeled by the trajectory inference function

$$f_P\colon \mathcal{G} \to \mathcal{T}, \tag{2.3}$$

mapping goals to trajectories, and $\xi_{\boldsymbol{s}\to\boldsymbol{g}^\star}$ is predictable if

$$f_P(\boldsymbol{g}^\star) = \xi_{\boldsymbol{s}\to\boldsymbol{g}^\star}. \tag{2.4}$$

The closeness of $\xi_{\boldsymbol{s}\to\boldsymbol{g}^\star}$ to the observer's inferred trajectory $f_P(\boldsymbol{g}^\star)$, given by a distance metric $d\colon \mathcal{T} \times \mathcal{T} \to \mathbb{R}_{\geq 0}$, is a measure of how predictable the trajectory $\xi_{\boldsymbol{s}\to\boldsymbol{t}}$ is. As an example, let us consider the five possible trajectories from $\boldsymbol{s}$ to $\boldsymbol{g}^\star$ in Figure 2.4. If we use the Fréchet distance [5] with the Euclidean distance as its distance function to compare trajectories, we would find that the dashed black line most closely matches the observer's inferred trajectory.

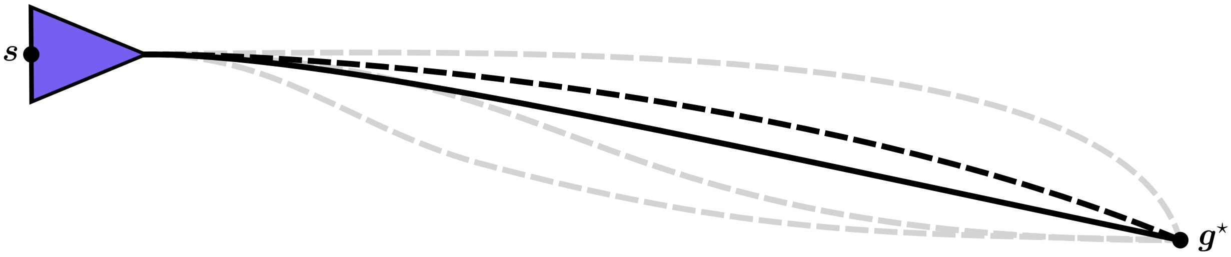


Figure 2.4: *Trajectory (dashed black) most closely matching the observer's expectation (solid black) of the possible robot trajectories from* $\boldsymbol{s}$ *to* $\boldsymbol{g}^\star$ *(dashed black/grey).*

2.3 Modelling the Robot's Motion

In order for a robot to consider how legible or predictable its motion is to an observer, it requires a model of the observer's expectation. That is, given the goal inference (2.1) and trajectory inference (2.3) functions, how does the observer map trajectories to goals and goals to trajectories? In this work, the observer is modeled according to [24] and using the principle of rational action [35]. From this, it is assumed that the observer expects the robot to be a rational agent seeking to reach its goal efficiently. This is represented by a cost functional

$$c\colon \mathcal{T} \to \mathbb{R}_{\geq 0}, \tag{2.5}$$

which the observer expects the robot to minimize.

2.3.1 Modelling Legible Motion

The goal inference function (2.1) models how the observer infers the robot's goal $\boldsymbol{g} \in \mathcal{G}$ given its past trajectory $\xi_{\boldsymbol{s}\to\boldsymbol{t}}$. This is modeled by computing the probability for each goal

$\boldsymbol{g} \in \mathcal{G}$ and choosing the most likely [24]:

$$f_L(\xi_{\boldsymbol{s}\to\boldsymbol{t}}) = \operatorname*{argmax}_{\boldsymbol{g}\in\mathcal{G}} P(\boldsymbol{g} \mid \xi_{\boldsymbol{s}\to\boldsymbol{t}}). \tag{2.6}$$

The above posterior probability is computed using Bayes' theorem:

$$P(\boldsymbol{g} \mid \xi_{\boldsymbol{s}\to\boldsymbol{t}}) = z_{\boldsymbol{s}\to\boldsymbol{t}}^{-1} P(\xi_{\boldsymbol{s}\to\boldsymbol{t}} \mid \boldsymbol{g}) P(\boldsymbol{g}), \tag{2.7}$$

where $P(\boldsymbol{g}) \in [0, 1]$ represents a prior on the goals with $\sum_{\boldsymbol{g}\in\mathcal{G}} P(\boldsymbol{g}) = 1$ and where

$$z_{\boldsymbol{s}\to\boldsymbol{t}} \coloneqq \sum_{\boldsymbol{g}\in\mathcal{G}} P\left(\xi_{\boldsymbol{s}\to\boldsymbol{t}} \mid \boldsymbol{g}\right) P\left(\boldsymbol{g}\right) \tag{2.8}$$

is a normalizer over $\mathcal{G}$. It is assumed that the prior on the goals, $P(\boldsymbol{g})$, is known or learned. However, in the general case, with no *a priori* knowledge, equal probability of $\frac{1}{|\mathcal{G}|}$ could be assigned to each goal point.

Having related the posterior as such, the likelihood, $P(\xi_{\boldsymbol{s}\to\boldsymbol{t}} \mid \boldsymbol{g})$, is modeled using the following Boltzmann policy:

$$P(\xi_{\boldsymbol{s}\to\boldsymbol{t}} \mid \boldsymbol{g}) = \frac{\exp\big(-\beta c_{\boldsymbol{g}}[\xi_{\boldsymbol{s}\to\boldsymbol{t}}]\big)}{\int_{\bar{\xi}_{\boldsymbol{s}\to\boldsymbol{g}\in\mathcal{T}}} \exp\big(-\beta c[\bar{\xi}_{\boldsymbol{s}\to\boldsymbol{g}}]\big) d\bar{\xi}_{\boldsymbol{s}\to\boldsymbol{g}}}, \tag{2.9}$$

where

$$c_{\boldsymbol{g}}[\xi_{\boldsymbol{s}\to\boldsymbol{t}}] = c[\xi_{\boldsymbol{s}\to\boldsymbol{t}}] + c[\xi^\star_{\boldsymbol{t}\to\boldsymbol{g}}], \tag{2.10}$$

expresses the cost to reach the goal $\boldsymbol{g}$ through $\xi_{\boldsymbol{s}\to\boldsymbol{t}}$ and the optimal trajectory $\xi^\star_{\boldsymbol{t}\to\boldsymbol{g}}$ and where $\xi^\star_{\boldsymbol{t}\to\boldsymbol{g}} = \operatorname{argmin}_{\xi_{\boldsymbol{t}\to\boldsymbol{g}}\in\mathcal{T}} c[\xi_{\boldsymbol{t}\to\boldsymbol{g}}]$. The Boltzmann policy models the observer as choosing trajectories proportional to their exponential cost [7]. The likelihood in (2.9) defines a probability distribution over the entire set of feasible trajectories $\xi_{\boldsymbol{s}\to\boldsymbol{t}} \in \mathcal{T}$. Here, β acts as a rationality parameter [7] and controls how likely, in the observer's model, the robot is to deviate from its rational trajectory (i.e. the minimizer of c). From (2.9), we see that as $\beta \to 0$, each trajectory becomes equally likely. On the other hand, when $\beta \to \infty$, the optimal trajectory converges to a probability of one and all other trajectories converge to zero probability. In other words, the observer models the robot as an irrational agent for small values of β and a rational agent for large values of β.

Since the integral in (2.9) is challenging to compute, the authors in [24] make an approximation using Laplace's method as derived in [25]. They begin by taking the second-order Taylor series expansion of $c[\xi_{\boldsymbol{s}\to\boldsymbol{g}}]$ about its optimal trajectory $\xi^\star_{\boldsymbol{s}\to\boldsymbol{g}}$:

$$c[\xi_{\boldsymbol{s}\to\boldsymbol{g}}] \approx c[\xi^\star_{\boldsymbol{s}\to\boldsymbol{g}}] + \delta\xi^\intercal_{\boldsymbol{s}\to\boldsymbol{g}} \nabla_{\xi^\star_{\boldsymbol{s}\to\boldsymbol{g}}} + \frac{1}{2}\delta\xi^\intercal_{\boldsymbol{s}\to\boldsymbol{g}} H_{\xi^\star_{\boldsymbol{s}\to\boldsymbol{g}}} \delta\xi_{\boldsymbol{s}\to\boldsymbol{g}} \tag{2.11}$$

where $\nabla_{\xi^\star_{s\to g}}$ and $H_{\xi^\star_{s\to g}}$ are respectively the gradient and Hessian of c evaluated at $\xi^\star_{\boldsymbol{s}\to\boldsymbol{g}}$ and $\delta\xi_{\boldsymbol{s}\to\boldsymbol{g}} = (\xi_{\boldsymbol{s}\to\boldsymbol{g}} - \xi^\star_{\boldsymbol{s}\to\boldsymbol{g}})$. Since $\xi^\star_{\boldsymbol{s}\to\boldsymbol{g}}$ minimizes c, the first order term vanishes. Plugging this into the integral in (2.9), they get the following:

$$\int_{\xi_{s\to g\in\mathcal{T}}} \exp\big(-\beta c[\xi_{\boldsymbol{s}\to\boldsymbol{g}}]\big) d\xi_{\boldsymbol{s}\to\boldsymbol{g}} \approx$$
$$\exp\big(-\beta c[\xi^\star_{\boldsymbol{s}\to\boldsymbol{g}}]\big) \int_{\xi_{\boldsymbol{s}\to\boldsymbol{g}\in\mathcal{T}}} \exp\big(-\tfrac{1}{2}\beta\delta\xi^\intercal_{\boldsymbol{s}\to\boldsymbol{g}} H_{\xi^\star_{\boldsymbol{s}\to\boldsymbol{g}}} \delta\xi_{\boldsymbol{s}\to\boldsymbol{g}}\big) d\xi_{\boldsymbol{s}\to\boldsymbol{g}}. \quad (2.12)$$

Lastly, they compute the Gaussian integral, which when substituted back into (2.9) gives:

$$P(\xi_{\boldsymbol{s}\to\boldsymbol{t}} \mid \boldsymbol{g}) \approx \exp\Big(\beta\big(c[\xi^\star_{\boldsymbol{s}\to\boldsymbol{g}}] - c_{\boldsymbol{g}}[\xi_{\boldsymbol{s}\to\boldsymbol{t}}]\big)\Big)\sqrt{\frac{\beta^n |H_{\xi^\star_{\boldsymbol{s}\to\boldsymbol{g}}}|}{2^n\pi^n}}. \quad (2.13)$$

This allows the approximation of (2.7) as

$$P(\boldsymbol{g} \mid \xi_{\boldsymbol{s}\to\boldsymbol{t}}) \approx z^{-1}_{\boldsymbol{s}\to\boldsymbol{t}} \exp\Big(\beta\big(c[\xi^\star_{\boldsymbol{s}\to\boldsymbol{g}}] - c_{\boldsymbol{g}}[\xi_{\boldsymbol{s}\to\boldsymbol{t}}]\big)\Big)\sqrt{\frac{\beta^n |H_{\xi^\star_{\boldsymbol{s}\to\boldsymbol{g}}}|}{2^n\pi^n}} P(\boldsymbol{g}). \quad (2.14)$$

Since the normalizer over the goals, $z^{-1}_{\boldsymbol{s}\to\boldsymbol{t}}$, is a constant term, plugging (2.14) into the goal inference (2.6) gives:

$$f_L(\xi_{\boldsymbol{s}\to\boldsymbol{t}}) = \operatorname*{argmax}_{\boldsymbol{g}\in\mathcal{G}} \exp\Big(\beta\big(c[\xi^\star_{\boldsymbol{s}\to\boldsymbol{g}}] - c_{\boldsymbol{g}}[\xi_{\boldsymbol{s}\to\boldsymbol{t}}]\big)\Big)\sqrt{\frac{\beta^n |H_{\xi^\star_{\boldsymbol{s}\to\boldsymbol{g}}}|}{2^n\pi^n}} P(\boldsymbol{g}). \quad (2.15)$$

By assuming a quadratic cost functional c, the Hessian also becomes a constant term, where the goal inference can be further simplified to

$$f_L(\xi_{\boldsymbol{s}\to\boldsymbol{t}}) = \operatorname*{argmax}_{\boldsymbol{g}\in\mathcal{G}} \exp\Big(\beta\big(c[\xi^\star_{\boldsymbol{s}\to\boldsymbol{g}}] - c_{\boldsymbol{g}}[\xi_{\boldsymbol{s}\to\boldsymbol{t}}]\big)\Big) P(\boldsymbol{g}). \quad (2.16)$$

Intuitively, modelling the observer's goal inference with (2.16) will select the goal to which the cost-to-goal of the robot's trajectory up to time t, $c_{\boldsymbol{g}}[\xi_{\boldsymbol{s}\to\boldsymbol{t}}]$, most closely matches the starting cost-to-goal of the optimal trajectory, $c[\xi^\star_{\boldsymbol{s}\to\boldsymbol{g}}]$. In other words, the most likely goal is that to which the robot's trajectory is most closely aligned with the optimal trajectory to that goal. Here, the closeness of the trajectories is measured by the chosen cost functional c. Figure 2.5 illustrates the goal inference (2.16) for our simplified scenario with two candidate goals in the workspace using trajectory length as the cost functional. We can see that the path length to $\boldsymbol{g}_2$ through $\boldsymbol{t}$ is much shorter than that to $\boldsymbol{g}_1$ through $\boldsymbol{t}$ relative to their optimal trajectories from the start, $\xi^\star_{\boldsymbol{s}\to\boldsymbol{g}_2}$ and $\xi^\star_{\boldsymbol{s}\to\boldsymbol{g}_1}$ respectively. Therefore, $|c[\xi^\star_{\boldsymbol{s}\to\boldsymbol{g}_2}] - c_{\boldsymbol{g}_2}[\xi_{\boldsymbol{s}\to\boldsymbol{t}}]| < |c[\xi^\star_{\boldsymbol{s}\to\boldsymbol{g}_1}] - c_{\boldsymbol{g}_1}[\xi_{\boldsymbol{s}\to\boldsymbol{t}}]|$, and from (2.16), $\boldsymbol{g}_2$ is correctly chosen as the most likely goal.

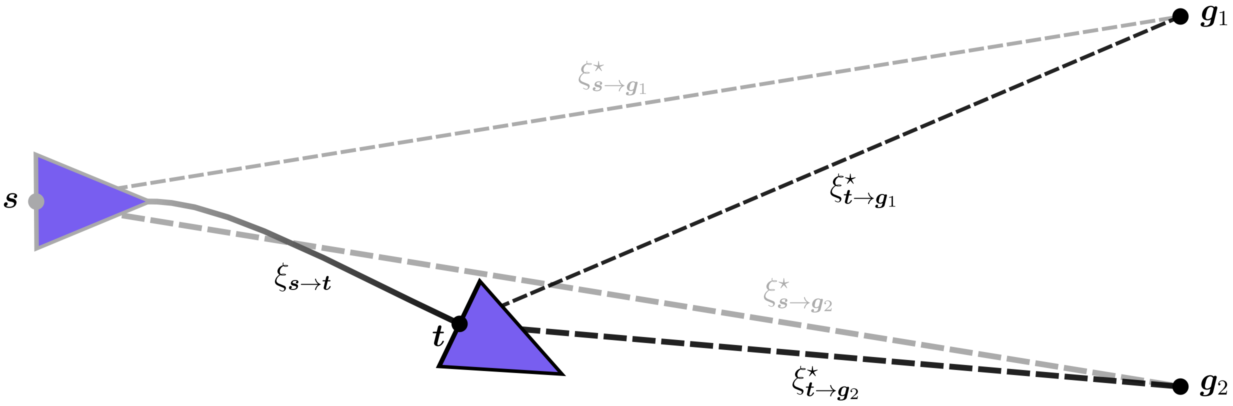

$\boldsymbol{g}_1$
$\xi^\star_{\boldsymbol{s}\to\boldsymbol{g}_1}$
$\xi^\star_{\boldsymbol{t}\to\boldsymbol{g}_1}$
$\boldsymbol{s}$
$\xi_{\boldsymbol{s}\to\boldsymbol{t}}$
$\xi^\star_{\boldsymbol{s}\to\boldsymbol{g}_2}$
$\boldsymbol{t}$
$\xi^\star_{\boldsymbol{t}\to\boldsymbol{g}_2}$
$\boldsymbol{g}_2$

Figure 2.5: *Optimal trajectories from the start* $\boldsymbol{s}$ *and from the position at time* $\boldsymbol{t}$ *to the goals* $\boldsymbol{g}_1$ *and* $\boldsymbol{g}_2$*. Since the path length to* $\boldsymbol{g}_2$ *through* $\boldsymbol{t}$ *is shorter than that to* $\boldsymbol{g}_1$ *through* $\boldsymbol{t}$ *relative to* $\xi^\star_{\boldsymbol{s}\to\boldsymbol{g}_2}$ *and* $\xi^\star_{\boldsymbol{s}\to\boldsymbol{g}_1}$ *respectively,* $\boldsymbol{g}_2$ *is correctly inferred as the robot's goal using* (2.16)*.*

2.3.2 Modelling Predictable Motion

The trajectory inference function (2.3) models how the observer infers the robot's trajectory, $\xi_{\boldsymbol{s}\to\boldsymbol{g}^\star}$, having knowledge of the robot's intended goal $\boldsymbol{g}^\star$. In fact, this directly models the observers expectation of the robot's motion. Recall that the observer is modelled as expecting the robot to seek to minimize the cost functional c (2.5). The trajectory inference function (2.3) is therefore modeled as the minimizer of c:

$$f_P(\boldsymbol{g}^\star) = \operatorname*{argmin}_{\xi_{\boldsymbol{s}\to\boldsymbol{g}^\star}\in\mathcal{T}} c[\xi_{\boldsymbol{s}\to\boldsymbol{g}^\star}]. \tag{2.17}$$

Given the set of robot trajectories $\xi_{\boldsymbol{s}\to\boldsymbol{g}^\star} \in \mathcal{T}$ depicted in Figure 2.6, and setting the cost functional c to be the trajectory length, the shortest trajectory (solid black) is the minimizer of (2.17) and therefore what the robot would model the observer as having inferred.

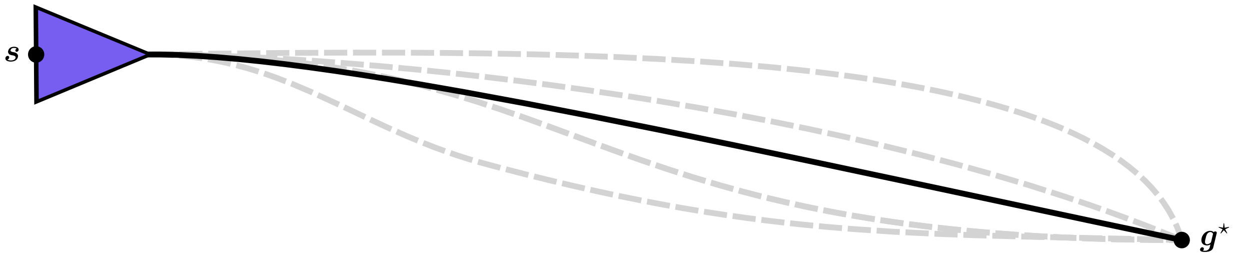


Figure 2.6: *Example set of robot trajectories $\mathcal{T}$ from $\boldsymbol{s}$ to $\boldsymbol{g}^\star$. Given trajectory length as the cost functional c, the robot would model the observer as having inferred the shortest trajectory (solid black).*

Chapter 3

Legibility and Predictability for Social Robot Navigation

In this section, we present an approach that extends the notions of legibility and predictability from [24] to dynamic goal regions. This allows us to apply these properties to navigation interactions in dynamic environments.

3.1 Problem Formulation

We consider a robot R moving in a planar workspace $\mathcal{W} \subseteq \mathbb{R}^2$ towards a stationary goal $\boldsymbol{g}_R \in \mathcal{W}$ and sharing the workspace with another dynamic agent A. An adaptation to the multi-agent case is provided in Section 3.8. To only make use of what is readily observable in a social environment, it is assumed that the robot's goal is unknown to A and the robot has no means of explicitly communicating its intended goal. Through its onboard sensors, we assume that the robot has access to the position $\boldsymbol{p}_i(t) \in \mathcal{W}$ and velocity $\boldsymbol{v}_i(t) \in \mathbb{R}^2$, $i \in \{R, A\}$, for $\|\boldsymbol{p}_R(t) - \boldsymbol{p}_A(t)\| \leq d_{\text{sense}}$, where d_{sense} is the sensor range. The heading of the robot and the other agent at time t are, respectively, the angle that $\boldsymbol{v}_R(t)$ and $\boldsymbol{v}_A(t)$ make with respect to a fixed axes in the inertial frame.

The control space of the robot is a finite set of motion primitives $\mathcal{P}$ and each motion primitive $\rho_i \in \mathcal{P}$ has the same time duration $\delta t > 0$. A navigation scheme for the robot is a sequence of primitive selections. The robot R is considered to be in a collision with the other agent A at time t if $\|\boldsymbol{p}_R(t) - \boldsymbol{p}_A(t)\| \leq r^C$ where $r^C > 0$ is a positive constant depending on the footprint of the robot and agent.

We seek to design a navigation strategy so the robot reaches its goal while simultaneously using the shape and speed of its trajectory to disambiguate its intentions to the other agent. To accomplish this, we take inspiration from the notions of legibility and predictability, in the sense of [24] and as reviewed in Section 2.1 and Section 2.2, which have been validated in the HRI field [26]. The way in which legibility and predictability are to be optimized poses an important design decision. In this work, we design a navigation framework based on the following high-level principles:

1. A robot should disambiguate its intended navigation strategy by being legible.
2. Once legible, the robot should proceed predictably.
3. Motion should adhere to social norms (i.e., left or right passing conventions).

3.2 Defining a Navigation Interaction Using Dynamic Goal Regions

It is important to differentiate the types of goals involved in social navigation. As a robot navigates an environment, there is likely a global goal position to which it is trying to reach. Although certain areas in the environment are more frequented than others, such as doorways and charging stations, it is unreasonable to assume that the other agent has complete knowledge of all possible goals nor that there are finitely many. For this reason, rather than trying to infer the robot's global goal from a possibly infinite set of candidate goals in the surroundings, the other agent will try and infer an intermediate goal, such as the robot's intended passing side. The differentiation between the global and intermediate goals is illustrated in Figure 3.1.

Although the other agent seeks to infer the intermediate goal, that is not to say that global goals will not affect this inference. For example, in Figure 3.1, at the beginning of the interaction, both agents are initially aligned to pass each other on the right. Having noticed the robot swerve to the left, the other agent will likely assume it is trying to reach the doorway, further decreasing the likelihood of the robot swerving back to pass on the right. However, in the absence of the doorway, the other agent might instead assume the robot is avoiding a nearby danger, such as a spill, after which it is likely to swerve back and return to its originally planned trajectory.

To compute the goal (2.1) and trajectory (2.3) inferences during social navigation, the robot first needs a model of a navigation interaction. As the robot navigates towards its

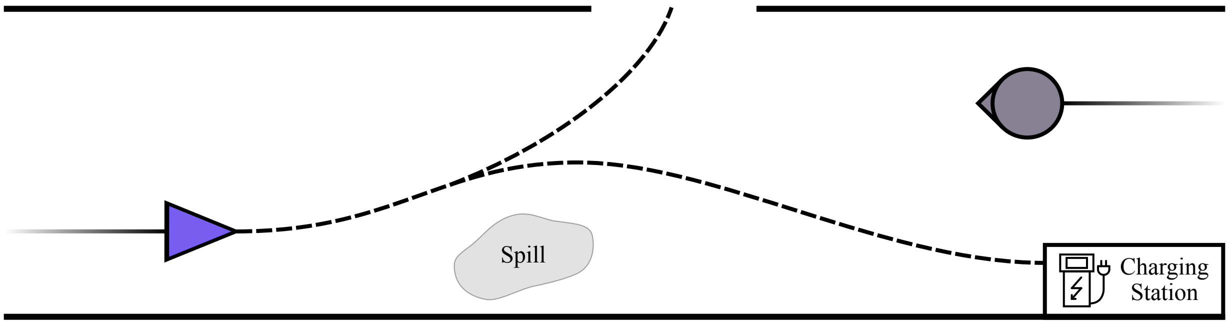


Figure 3.1: *The effect of global goals on the intermediate goal inference.*

global goal $\boldsymbol{g}_R$, it will attain intermediate navigation goals, which we refer to as interaction goals. We consider three possible interaction goals: passing on the left, passing on the right and colliding.[1] These goals can be represented by regions that translate according to the other agent's motion. The interaction goals are therefore dynamic and non-singleton *sets*. These properties distinguish the interaction goals from the static singletons taken as goals in previous works [77, 14, 30].

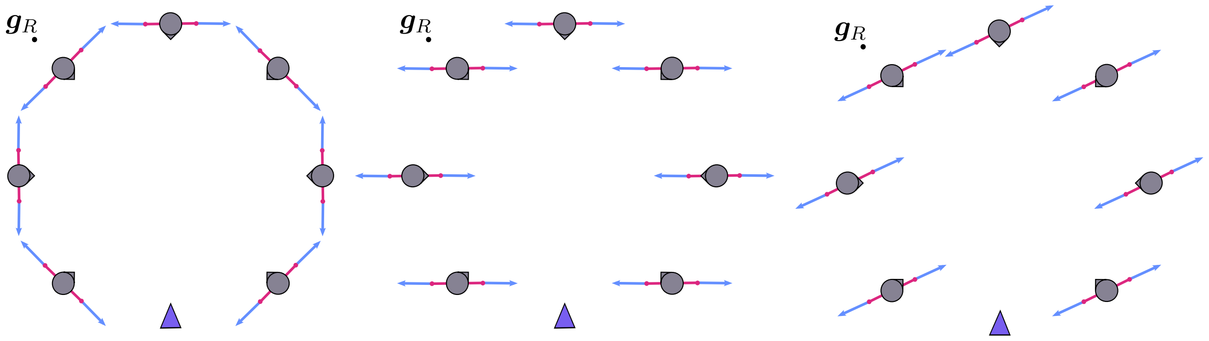
Let $\mathcal{I}(t)$, hereinafter referred to as the interaction line at time t, represent the line passing through the other agent's position $\boldsymbol{p}_A(t)$ oriented to be orthogonal to the vector pointing from the robot's position $\boldsymbol{p}_R(t)$ to its goal $\boldsymbol{g}_R(t)$ (see Figure 3.3). Orienting the interaction line as such allows the robot to reason about a passing side irrespective of the other agent's heading (i.e. front/back in a t-junction scenario or left/right in a head-on scenario).

A comparison between candidate methods of orienting the interaction line is illustrated in Figure 3.2. For example, let us consider the configurations in (a) where the line is oriented with the other agent's heading. Specifically, the agents travelling orthogonally to the robot's motion (left/right-most agents) depict a problematic configuration. In this configuration, the interaction line is parallel to the robot's motion. This wouldn't allow the robot to evaluate the goal and trajectory inferences for passing in front or in back of the other agent. Instead, it would always plan according to the left and right regions. This modeling would also make the interaction very sensitive to changes in the other agent's heading.

Orienting the line to be orthogonal to the robot's heading, as shown in (b), avoids the problematic configuration discussed above. However, orienting the line with the robot would make the interaction very sensitive to changes in the robot's heading. Imagine

[1] The terminology for interactions is from the robot's point-of-view.

a simple avoidance manoeuvre where the robot swerves right to indicate its intention. During the manoeuvre, the interaction line would rotate towards the robot and possibly pass through it. Orienting the line to be orthogonal to the vector pointing from the robot to its goal, as shown in (c) overcomes these issues. The robot can reason on its motion with respect to a passing side, be it left/right or front/back depending on the configuration, while the interaction line remains robust to changes in either agent's heading.



(a) Agent oriented **(b)** Robot oriented **(c)** Robot-goal oriented

Figure 3.2: *Comparison of different methods to orient the interaction line.*

Formally, at time t and with $\boldsymbol{p}_R(t) \neq \boldsymbol{g}_R$, define the unit vector

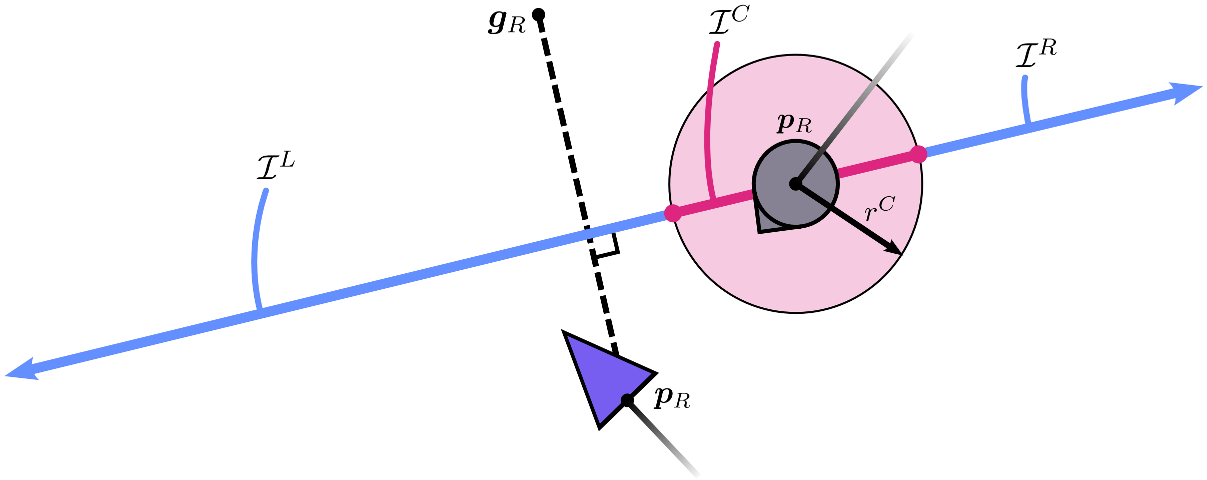


Figure 3.3: *Interaction line segmented into its collision segment $\mathcal{I}^C$ (magenta) and two rays, $\mathcal{I}^L$ and $\mathcal{I}^R$ (blue) representing passing on the left and right respectively.*

$$\boldsymbol{e}(t) \coloneqq \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \left(\frac{\boldsymbol{g}_R - \boldsymbol{p}_R(t)}{\|\boldsymbol{g}_R - \boldsymbol{p}_R(t)\|} \right) \tag{3.1}$$

and the real one-dimensional subspace

$$\mathcal{V}(t) \coloneqq \operatorname{span}\{\boldsymbol{e}(t)\},$$

then the interaction line is the time-varying one-dimensional affine subspace

$$\mathcal{I}(t) \coloneqq \{\boldsymbol{v} + \boldsymbol{p}_A(t) : \boldsymbol{v} \in \mathcal{V}(t)\}. \tag{3.2}$$

The time-to-interaction (TTI) at time t is defined to be the infimum (possibly infinite) time it would take the robot R to reach the interaction line assuming the robot and other agent continue at their current velocities and $\mathcal{I}(t)$ translates with A.

Definition 3.2.1 (Navigation interaction)**.** Robot R is said to be interacting with agent A if $\|\boldsymbol{p}_R(t) - \boldsymbol{p}_A(t)\| \leq d_{\text{sense}}$, the interaction line separates R from its goal $\boldsymbol{g}_R$ and the TTI is less than or equal to a pre-defined maximum interaction time $t_{\mathcal{I}}^{\max} > 0$.

In order to define the interaction goals, the interaction line is segmented into passing and collision regions as shown in Figure 3.3. The collision line segment, $\mathcal{I}^C$, is defined as the intersection between the closed disc centered at $\boldsymbol{p}_A$ of radius r^C and the interaction line

$$\mathcal{I}^C(t) \coloneqq \mathcal{I}(t) \cap \left\{\boldsymbol{p} \in \mathcal{W} : \|\boldsymbol{p} - \boldsymbol{p}_A(t)\| \leq r^C\right\}. \tag{3.3}$$

Each passing interaction, $\mathcal{I}^R$ and $\mathcal{I}^L$, is described by an open ray starting at the collision segment's endpoints, extending in the direction opposite to the other agent. The left passing side is defined as

$$\mathcal{I}^L(t) \coloneqq \left\{\boldsymbol{p}_A(t) + \alpha\boldsymbol{e}(t) : \alpha > r^C\right\} \tag{3.4}$$

and the right passing side is defined as

$$\mathcal{I}^R(t) \coloneqq \left\{\boldsymbol{p}_A(t) - \alpha\boldsymbol{e}(t) : \alpha > r^C\right\}. \tag{3.5}$$

These three interaction goals, denoted by the collection $\boldsymbol{\mathcal{G}} = \left\{\mathcal{I}^L, \mathcal{I}^C, \mathcal{I}^R\right\}$, represent dynamic goal regions fixed to the other dynamic agent. The passing interaction goals are specifically denoted by $\boldsymbol{\mathcal{G}}^P \coloneqq \left\{\mathcal{I}^L, \mathcal{I}^R\right\}$.

Remark 1 (Role of the Global Goal). The robot's global goal point, $\boldsymbol{g}_R \in \mathcal{W}$, is not an interaction goal, i.e. $\boldsymbol{g}_R \notin \boldsymbol{\mathcal{G}}$. This matches reality, where the observer does not try to make an inference on the robot's motion using an inferred global goal, but rather uses the interaction goal.

3.3 Robot and Observer Motion Models

The goal and trajectory inferences presented in Section 2 are made based on the observer's model of the robot's behavior. For example, if the observer expects the robot to be a rational agent acting efficiently, they might model the robot so as to always choose the trajectory of shortest length. Another observer might instead model the robot's behavior to be more aggressive, thus promoting swerving as opposed to slowing down to minimize the time-to-goal. We assume the observer expects the robot to be a rational agent seeking to move efficiently in the environment. As such, the observer's model of the robot's motion minimizes the cost functional

$$c[\xi] = (t_f - t_s)^2, \tag{3.6}$$

where $c\colon \mathcal{T} \to \mathbb{R}_{\geq 0}$ maps robot trajectories $\xi \in \mathcal{T}$ to the square of its duration.

The interaction goals defined in (3.3), (3.4) and (3.5) are dynamic and translate according to the other agent's motion. Evaluating the costs of trajectories $\xi \in \mathcal{T}$ will therefore require a trajectory prediction model. This is still an active area of research [81], where there has recently been a significant amount of progress amongst the computer vision community [3, 37, 82, 68, 36, 58]. These prediction models are, however, not trained on trajectories in which a robot is involved. This new and relatively less researched area has shown that humans exhibit different navigation strategies when interacting with a robot than with another human [97, 22]. Although this field is continuing to expand, very few approaches consider the effect of including robots in the environment [80].

Given a prediction model chosen by the designer, the robot predicts the interaction goal from $\mathcal{I}(t)$ at the current time to the end of the interaction at time t_f with

$$\widehat{\mathcal{I}}_{t\to t_f} = \text{prediction}(\xi^A_{\boldsymbol{s}\to\boldsymbol{t}}, \mathcal{I}(t)), \tag{3.7}$$

where $\widehat{\mathcal{I}}_{t\to t_f} \in \boldsymbol{\mathcal{G}}$ is the predicted interaction in the closed interval $[t, t_f]$ and $\xi^A_{\boldsymbol{s}\to\boldsymbol{t}}$ is an observed segment of the other agent's trajectory.

3.4 Legibility and Predictability of Navigation Interactions

Given a robot trajectory $\xi_{\boldsymbol{s}\to\boldsymbol{t}}$ up to time t, we model the observer's legibility inference function (2.1) as returning the most likely interaction goal $\mathcal{I}(t)$ from the finite collection of possible goals $\boldsymbol{\mathcal{G}}(t)$:

$$f_L(\xi_{\boldsymbol{s}\to\boldsymbol{t}}) = \underset{\mathcal{I}(t)\in\boldsymbol{\mathcal{G}}(t)}{\operatorname{argmax}} P(\mathcal{I}(t) \mid \xi_{\boldsymbol{s}\to\boldsymbol{t}}). \tag{3.8}$$

We compute the above posterior probability following the derivations in [24], as was reviewed in Section 2.3. This involves using Bayes' theorem to obtain the likelihood of $\mathcal{I}$, which we model using a Boltzmann policy whose partition function (an infinite integral over $\mathcal{T}$ appearing in the denominator) is approximated using Laplace's method as derived in [25]. Assuming a quadratic cost functional c, the posterior of interaction $\mathcal{I} \in \boldsymbol{\mathcal{G}}$ can be approximated by

$$P(\mathcal{I} \mid \xi_{\boldsymbol{s}\to\boldsymbol{t}}) \approx \frac{\exp\Big(\beta\big(c[\xi^\star_{\boldsymbol{s}\to\widehat{\mathcal{I}}}] - c_{\widehat{\mathcal{I}}}[\xi_{\boldsymbol{s}\to\boldsymbol{t}}]\big)\Big)}{\sum_{\bar{\mathcal{I}}\in\boldsymbol{\mathcal{G}}} P\left(\xi_{\boldsymbol{s}\to\boldsymbol{t}} \mid \bar{\mathcal{I}}\right) P\left(\bar{\mathcal{I}}\right)} P(\mathcal{I}), \tag{3.9}$$

where the denominator is a normalizer over goals $\mathcal{I} \in \boldsymbol{\mathcal{G}}$, $\xi^\star_{\boldsymbol{t}\to\widehat{\mathcal{I}}} = \operatorname{argmin}_{\xi_{\boldsymbol{t}\to\widehat{\mathcal{I}}}\in\mathcal{T}} c[\xi_{\boldsymbol{t}\to\widehat{\mathcal{I}}}]$ is the optimal cost to reach the predicted interaction goal, $c_{\widehat{\mathcal{I}}}$ is the cost to reach $\widehat{\mathcal{I}}$ through $\xi_{\boldsymbol{s}\to\boldsymbol{t}}$ and the optimal trajectory $\xi^\star_{\boldsymbol{t}\to\widehat{\mathcal{I}}}$, $P(\mathcal{I}) \in [0, 1]$ represents the prior on interaction $\mathcal{I}(t)$ with $\sum_{\mathcal{I}\in\boldsymbol{\mathcal{G}}} P(\mathcal{I}) = 1$ and $\beta_{\geq 0}$ acts as a rationality parameter [7]. To adhere to social norms, a larger prior could be assigned to the customary passing side.

Let us now assume the robot's interaction goal region at time t, $\mathcal{I}^\star(t)$ is known to the observer. We model the observer's predictability inference function (2.3) as the most likely trajectory $\xi_{\boldsymbol{t}\to\mathcal{I}^\star}$ from the set of possible trajectories $\mathcal{T}$ in the following sense:

$$f_P(\mathcal{I}^\star(t)) = \underset{\xi_{\boldsymbol{t}\to\mathcal{I}^\star(t)}\in\mathcal{T}}{\operatorname{argmax}} P(\xi_{\boldsymbol{t}\to\mathcal{I}^\star(t)} \mid \mathcal{I}^\star(t)). \tag{3.10}$$

At the start, i.e. $\xi(t_s) = s$, the observer's predictability inference (3.10) is the same as was proposed in [24]. Modeling f_P with (3.10) allows us to consider the inference of partial trajectories. As a result, the notion of predictability becomes well-suited for motion planning in dynamic environments. It should be noted that this is more closely related to the notion of t-predictability from [33]. However, rather than inferring a sequence of actions, we consider a trajectory.

To compute the trajectory inference (3.10), we model $P(\xi_{\boldsymbol{t}\to\mathcal{I}^\star})$ as a Boltzmann policy and again approximate its partition function using Laplace's method to obtain:

$$P(\xi_{\boldsymbol{t}\to\mathcal{I}^\star} \mid \mathcal{I}^\star) \approx \exp\Big(\beta\big(c[\xi^\star_{\boldsymbol{t}\to\widehat{\mathcal{I}}^\star}] - c[\xi_{\boldsymbol{t}\to\mathcal{I}^\star}]\big)\Big). \tag{3.11}$$

Remark 2 (Other Agent's Motion Model). Solving the goal inference (3.9) and the trajectory inference (3.11) require the computations of the optimal trajectories to the dynamic interaction regions, namely $\xi^\star_{\boldsymbol{s}\to\widehat{\mathcal{I}}}$ and $\xi^\star_{\boldsymbol{t}\to\widehat{\mathcal{I}}}$, subject to c. In this work, the other agent's motion is modeled using the constant velocity model (CVM). This is reasonable as it has recently been shown to perform comparatively well, and in some cases outperform, state-of-the-art learning-based pedestrian prediction methods [83].

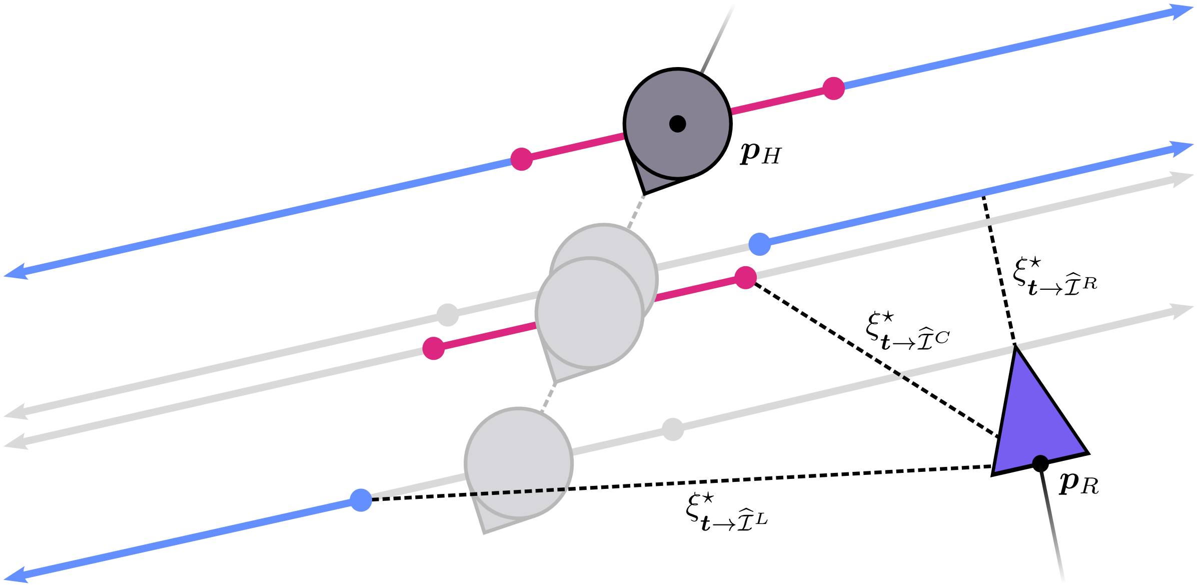
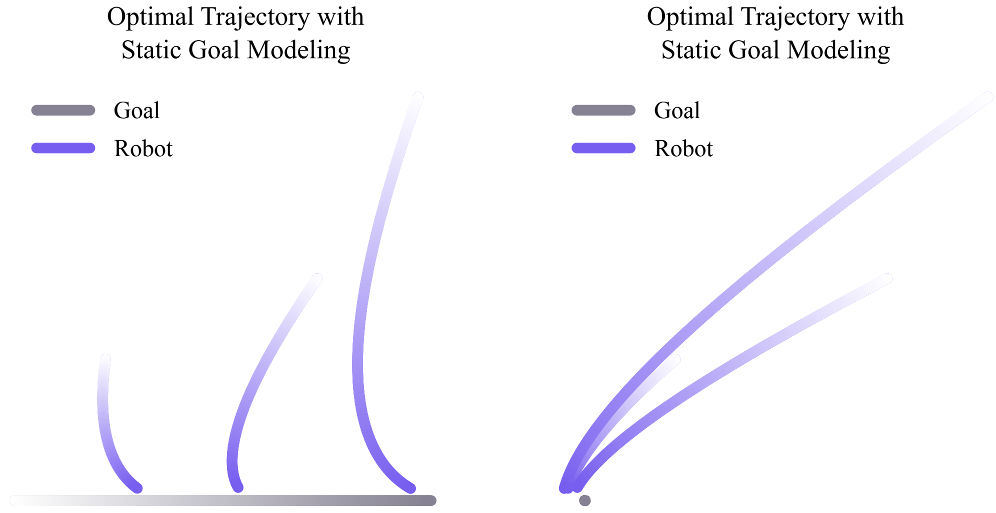

$\boldsymbol{p}_H$
$\xi^\star_{\boldsymbol{t}\to\widehat{\mathcal{I}}^R}$
$\xi^\star_{\boldsymbol{t}\to\widehat{\mathcal{I}}^C}$
$\boldsymbol{p}_R$
$\xi^\star_{\boldsymbol{t}\to\widehat{\mathcal{I}}^L}$

Figure 3.4: *Optimal trajectories for single integrator dynamics, $\xi^\star_{\boldsymbol{t}\to\mathcal{I}^R}$, $\xi^\star_{\boldsymbol{t}\to\mathcal{I}^C}$ and $\xi^\star_{\boldsymbol{t}\to\mathcal{I}^L}$, from the robot's current position $\boldsymbol{t}$ to the predicted passing on the right, collision and passing on the left interaction regions respectively.This configuration corresponds to the first case in* (3.21).

3.5 Deriving the Optimal Costs for the CVM

The cost of the optimal trajectory to the predicted interaction, $c[\xi^\star_{\boldsymbol{t}\to\widehat{\mathcal{I}}}]$, appears in both the trajectory (3.9) and goal (3.11) conditionals. It should be noted that alternate costs to (3.6) can be chosen. Depending on the chosen prediction model in (3.7), this cost risks being a computationally heavy trajectory optimization problem. To improve efficiency, existing prediction models often make simplifying assumptions, such as assuming the other's behavioral states are static within the observation and prediction horizon [60]. To overcome high dimensionality when predicting multiple future trajectories, they can also be approximated through sampling [54].

Computing the conditionals from Section 3.4 in real-time for dynamic goals is more challenging than when considering static goals. An initial attempt at simplifying this computation is to model the other agent as a static obstacle. This is akin to early reactive-based navigation approaches, which, by assuming a sufficiently fast update frequency, seek to capture the other agent's dynamics. To illustrate this, let us consider a single goal point translating at constant velocity, represented by the gray line in Figure 3.5. The optimal trajectories, shown in purple, to reach the goal from different start locations were obtained using CasADi [6]. The robot and the goal's speeds are constrained to the same value. These trajectories illustrate the robot's attempt at attaining the goal's instantaneous position, rather than an eventual position. From each of the three starting locations, the robot ends up trailing the goal from behind, failing to intercept it. This illustrates how modelling the goal as a static obstacle can result in a poor approximation of the optimal trajectory to reach a dynamic goal.

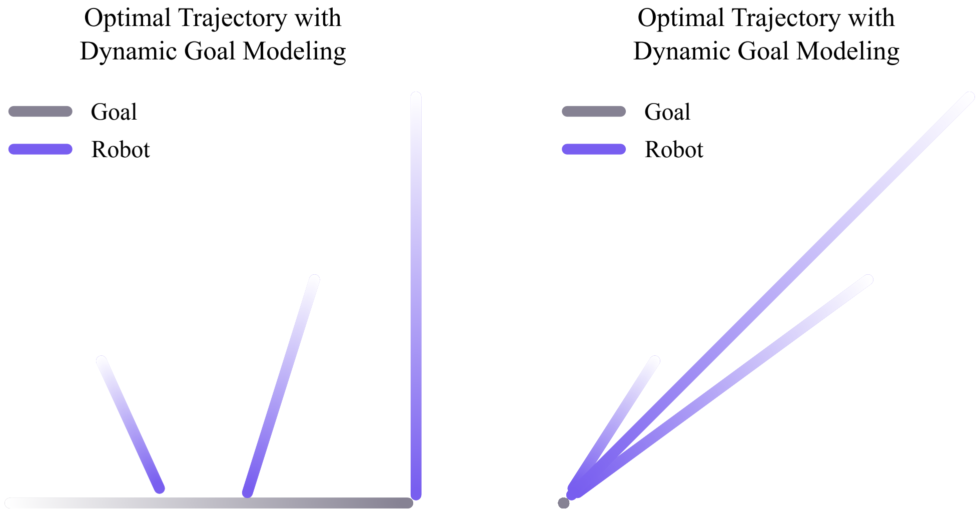


(a) Absolute coordinates

(b) Goal-relative coordinates

Figure 3.5: *Optimal trajectories with static goal modeling to reach the moving goal in absolute (a) and goal-relative (b) coordinates from different start locations.*

To improve the approximation of the optimal trajectory to reach the goal, let us model the goal's motion using its current position and velocity, rather than just its current position. The optimal trajectories obtained with dynamic modelling are shown in Figure 3.6. Since the goal is translating at a constant velocity, the optimal solution is the shortest straight path resulting in an interception. This is also shown in the goal's coordinate frame in Figure 3.6 (b). For each of the robot's starting positions, its motion is directly aligned to intercept the goal. Here, the bearing angle is maintained constant, which was not the case with static modelling 3.5 (b).



(a) Absolute coordinates

(b) Goal-relative coordinates

Figure 3.6: *Optimal trajectories with dynamic goal modeling to reach the moving goal in absolute (a) and goal-relative (b) coordinates from different start locations.*

In the above example, the robot's speed was fixed to that of the goal. However, in practice, the robot should be able to adjust its speed, allowing it to slow down to avoid a collision and accelerate when there are no nearby obstacles. The cost function in (2.5) should be carefully chosen. An initial attempt to capture the robot's efficiency would be to use its trajectory length as the cost. However, since the goal is dynamic, it's possible the robot will stop and wait for the goal to come, thus minimizing the cost. This would result in undesirable behavior since the robot would make very little progress in an environment with nearby goals. Alternatively, a time-based cost function would be better suited for dynamic goals. As such, the robot would seek to minimize the time taken to reach the goals. This cost will ensure efficient motion while encouraging continued progress in the environment.

We approximate the costs of the optimal trajectories appearing in (3.9) and (3.11) as the minimum time to reach each of the three interaction regions assuming the other agent follows the CVM. These are calculated by building on the idea of constant bearing control from [12] for a single integrator robot and by translating the interaction line at time t along with the other agent's predicted trajectory. To find the optimal costs to each interaction region, one need only compute the minimal times to reach the endpoints of the collision line segment, $\mathcal{I}^C$, and the interaction line (see Figure 3.4). We denote the positions of the passing on the right and left interaction rays' endpoints by

$$\boldsymbol{p}_{\mathcal{I}^R}(t) \coloneqq \boldsymbol{p}_A(t) - r^C \boldsymbol{e}(t) \tag{3.12}$$

and

$$\boldsymbol{p}_{\mathcal{I}^L}(t) \coloneqq \boldsymbol{p}_A(t) + r^C \boldsymbol{e}(t) \tag{3.13}$$

respectively.

To compute the minimal time taken by the robot to reach the point $\boldsymbol{p}_{\mathcal{I}}$, $\mathcal{I} \in \{\mathcal{I}^R, \mathcal{I}^L\}$, we use the constant bearing angle. The goal is to determine the optimal trajectory from the robot's position to the dynamic goal point, assuming each agent travels at constant velocity. For this tracking strategy, the robot must match its perpendicular velocity component, $\boldsymbol{v}_{R_\perp}$, to that of the interaction point $\boldsymbol{v}_{\mathcal{I}_\parallel}$. Given the robot's maximum speed, $v_R^{\max}$, an eventual interception is only guaranteed if $v_R^{\max} \leq |\boldsymbol{v}_{\mathcal{I}}|$. The constant bearing strategy is illustrated in Figure 3.7, where the bearing angle β, determined by the angle between the robot's heading and the vector directed from the robot to the interaction point $\mathcal{I}$, can be seen to remain constant at each configuration along the trajectory. Since the point's position and velocity are known, it remains to find the direction to which the robot should travel to match the interaction point's perpendicular velocity component $v_{\mathcal{I}_\perp}$ while navigating at its maximum speed.

We begin by setting the perpendicular component of the robot's velocity to that of the interaction point's:

$$\boldsymbol{v}_{R_\perp} = \boldsymbol{v}_{\mathcal{I}_\perp}. \tag{3.14}$$

Since the distance between $\boldsymbol{p}_R$ and $\boldsymbol{p}_{\mathcal{I}}$ is known and their perpendicular velocity components are equal, we compute the interception time using the parallel components of the velocities:

$$t_{\boldsymbol{p}_R,\boldsymbol{p}_{\mathcal{I}}} = \frac{\|\boldsymbol{p}_R - \boldsymbol{p}_{\mathcal{I}}\|}{v_{R_\parallel} + v_{\mathcal{I}_\parallel}}. \tag{3.15}$$

It should be noted that $v_{\mathcal{I}_\parallel}$ is a signed speed, where a positive value indicates a velocity that is directed towards the robot (as is the case in Figure 3.7), whereas a negative value indicates a velocity that is pointed outwards. Since we're only concerned with the optimal time to reach $\boldsymbol{p}_{\mathcal{I}}$, we ignore the negative speed $v_{R_\parallel}$ corresponding to the other velocity $\boldsymbol{v}_R$ matching $v_{\mathcal{I}_\perp}$ given by $\beta > \frac{\pi}{2}$. Since the interaction point's velocity $v_{\mathcal{I}}$ is known, the sign

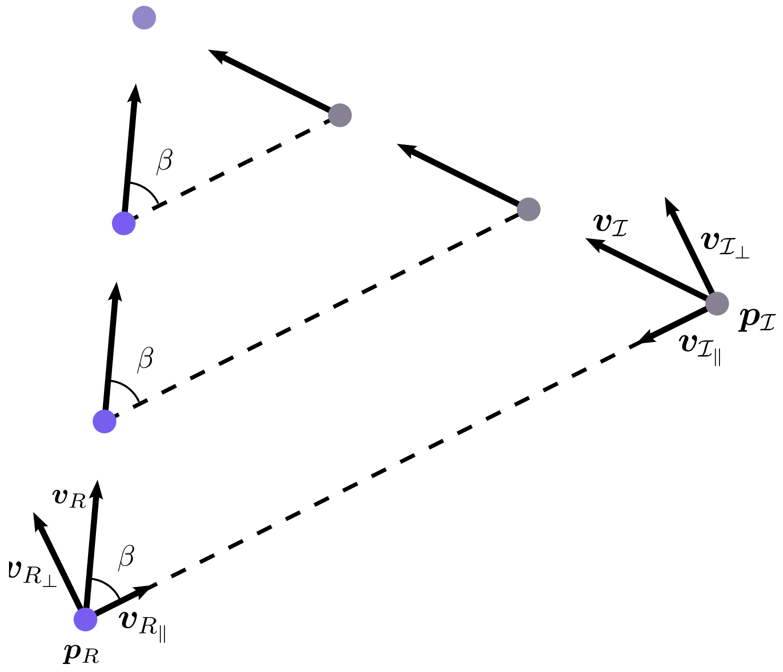

β
β
β
$\boldsymbol{v}_{\mathcal{I}}$
$\boldsymbol{v}_{\mathcal{I}_\perp}$
$\boldsymbol{p}_{\mathcal{I}}$
$\boldsymbol{v}_{\mathcal{I}_\parallel}$
$\boldsymbol{v}_R$
$\boldsymbol{v}_{R_\perp}$
$\boldsymbol{v}_{R_\parallel}$
$\boldsymbol{p}_R$

Figure 3.7: *Constant bearing strategy to intercept the interaction point $\mathcal{I}$ from $\boldsymbol{p}_R$.*

of its parallel component $v_{\mathcal{I}_\parallel}$ is given when projecting onto the vector pointing from $\boldsymbol{p}_{\mathcal{I}}$ to $\boldsymbol{p}_R$. The parallel speed of the robot is given by:

$$v_{R_\parallel} = \boldsymbol{v}_R \cos\beta, \tag{3.16}$$

where, by the assumption in (3.14),

$$\beta = \arcsin\left(\frac{v_{\mathcal{I}_\perp}}{v_R}\right). \tag{3.17}$$

Substituting (3.17) into (3.16) we get:

$$\begin{aligned} v_{R_\parallel} &= v_R \cos\left(\arcsin\left(\frac{v_{\mathcal{I}_\perp}}{v_R}\right)\right) \\ &= v_R\sqrt{1-\left(\frac{v_{\mathcal{I}_\perp}}{v_R}\right)^2}. \end{aligned} \tag{3.18}$$

Finally, if we set the robot's speed to its maximum $v_R^{\max}$, we can substitute (3.18) into (3.15) to obtain the optimal time taken by the robot to reach the interaction point:

$$t^\star_{\boldsymbol{p}_\mathcal{I}}(t,\mathcal{I}) = \frac{\|\boldsymbol{p}_R(t) - \boldsymbol{p}_\mathcal{I}(t)\|}{v_R^{\max}\sqrt{1-\left(\frac{v_{\mathcal{I}_\perp}(t)}{v_R^{\max}}\right)^2} + v_{\mathcal{I}_\parallel}(t)}, \quad \mathcal{I}\in \boldsymbol{\mathcal{G}}^P(t). \tag{3.19}$$

Computing the minimal time to reach the interaction line is simpler, and is given at each time t by:

$$t^\star_\mathcal{I}(t) = \frac{\|(\boldsymbol{p}_R(t)-\boldsymbol{p}_A(t))^\intercal\, \hat{\boldsymbol{q}}(t)\|}{v_R^{\max}+\boldsymbol{v}_A^\intercal(t)\hat{\boldsymbol{q}}(t)}, \tag{3.20}$$

where $\hat{\boldsymbol{q}} = (\boldsymbol{g}_R - \boldsymbol{p}_R)/\|\boldsymbol{g}_R - \boldsymbol{p}_R\|$ is a unit vector normal to the interaction line.

The three minimal times computed in (3.19) and (3.20) are assigned to the three interaction regions as follows:

$$(t^\star_{\mathcal{I}^R}, t^\star_{\mathcal{I}^C}, t^\star_{\mathcal{I}^L}) = \begin{cases} (t^\star_\mathcal{I}, t^\star_{\boldsymbol{p}_{\mathcal{I}^R}}, t^\star_{\boldsymbol{p}_{\mathcal{I}^L}}) & \text{if } t^\star_{\boldsymbol{p}_{\mathcal{I}^R}} < t^\star_\mathcal{I} < t^\star_{\boldsymbol{p}_{\mathcal{I}^L}} \\ (t^\star_{\boldsymbol{p}_{\mathcal{I}^R}}, t^\star_{\boldsymbol{p}_{\mathcal{I}^L}}, t^\star_\mathcal{I}) & \text{if } t^\star_{\boldsymbol{p}_{\mathcal{I}^L}} < t^\star_\mathcal{I} < t^\star_{\boldsymbol{p}_{\mathcal{I}^R}} \\ (t^\star_{\boldsymbol{p}_{\mathcal{I}^R}}, t^\star_\mathcal{I}, t^\star_{\boldsymbol{p}_{\mathcal{I}^L}}) & \text{otherwise.} \end{cases} \tag{3.21}$$

For example, in Figure 3.4, the minimal time to reach the interaction line, $t^\star_\mathcal{I}$, is associated with the trajectory perpendicular to the interaction line, $\xi^\star_{\boldsymbol{t}\to\mathcal{I}^R}$, whereas the optimal costs to each ray's endpoint, $t_{\mathcal{I}^R}$ and $t_{\mathcal{I}^L}$, are associated with $\xi^\star_{\boldsymbol{t}\to\mathcal{I}^C}$ and $\xi^\star_{\boldsymbol{t}\to\mathcal{I}^L}$ respectively.

3.6 Goal and Trajectory Conditionals of Primitives

To trade-off between legibility and predictability, the robot queries the scores from its set of motion primitives $\mathcal{P}$. Each motion primitive $\rho_i \in \mathcal{P}$ has the same duration δt. At each planning cycle, the trajectory (3.9) and interaction (3.11) conditionals are computed for each primitive with respect to each goal. To adapt more quickly in dynamic environments, we set the trajectory's starting position in (3.9) to a receding position of finite time-horizon t_p, as the interaction progresses. Specifically, at each new planning horizon, we redefine t_s as $\max(t_s, t - t_p)$. Intuitively, this means the robot will reason about the legibility of its more recent trajectory segments, rather than over its entire trajectory from the beginning of the interaction. The trajectory conditional for primitive ρ_i and interaction goal $\mathcal{I} \in \boldsymbol{\mathcal{G}}(t)$ becomes

$$P(\mathcal{I} \mid \xi_{\boldsymbol{s}\to\boldsymbol{\rho}_i}) \approx \frac{\exp\Big(-\beta\big(c[\xi^\star_{\boldsymbol{s}\to\widehat{\mathcal{I}}}] - c_{\widehat{\mathcal{I}}}[\xi_{\boldsymbol{s}\to\boldsymbol{\rho}_i}]\big)\Big)}{\sum_{\bar{\mathcal{I}}\in\boldsymbol{\mathcal{G}}} P\left(\xi_{\boldsymbol{s}\to\boldsymbol{\rho}_i} \mid \bar{\mathcal{I}}\right) P\left(\bar{\mathcal{I}}\right)} P(\mathcal{I}), \tag{3.22}$$

where $\xi_{\boldsymbol{s}\to\boldsymbol{\rho}_i} = \xi_{\boldsymbol{s}\to\boldsymbol{t}} + \xi_{\boldsymbol{t}\to\boldsymbol{\rho}_i}$ and $\boldsymbol{\rho}_i$ is the robot's position after completing primitive ρ_i.

In previous works [24], the robot's goal remains unchanged. In contrast, to adapt to the other agent's behavior, we make no such assumption and allow the robot to dynamically switch its intended passing side. During an interaction, the robot continuously updates $\mathcal{I}^\star$ to be the goal region which is most likely, i.e., we set $\mathcal{I}^\star = \text{argmax}_{\mathcal{I}\in\boldsymbol{\mathcal{G}}} P(\mathcal{I} \mid \xi_{\boldsymbol{s}\to\boldsymbol{t}})$. The goal conditional for trajectory $\xi_{\boldsymbol{t}\to\boldsymbol{\rho}_i}$ becomes:

$$P(\xi_{\boldsymbol{t}\to\boldsymbol{\rho}_i} \mid \mathcal{I}^\star) \approx \exp\Big(\beta\big(c[\xi^\star_{\boldsymbol{t}\to\mathcal{I}^\star}] - c_{\mathcal{I}^\star}[\xi_{\boldsymbol{t}\to\boldsymbol{\rho}_i}]\big)\Big). \tag{3.23}$$

Unlike the trajectory conditional (3.22), for the goal conditional (3.23), it is important that the underlying interaction be set. To illustrate this, let us consider the two motion primitives, ρ_1 and ρ_2, highlighted in Figure 3.8. Let us assume that ρ_1 and ρ_2 are perfectly aligned with the optimal trajectory from the robot's current position to the right and left interaction regions respectively. Without having set an underlying interaction, each optimal trajectory, $\xi^\star_{\rho_1\to\widehat{\mathcal{I}}^R}$ and $\xi^\star_{\rho_2\to\widehat{\mathcal{I}}^L}$, will have probability one when conditioned on the interaction region to which they attain. Since both solutions are optimal and equally likely, without implementing a tie-breaker condition, the robot will oscillate between between each interaction region. To mitigate this behavior, the robot instead conditions the trajectories on the goal region to which it is most legible, as explained above.

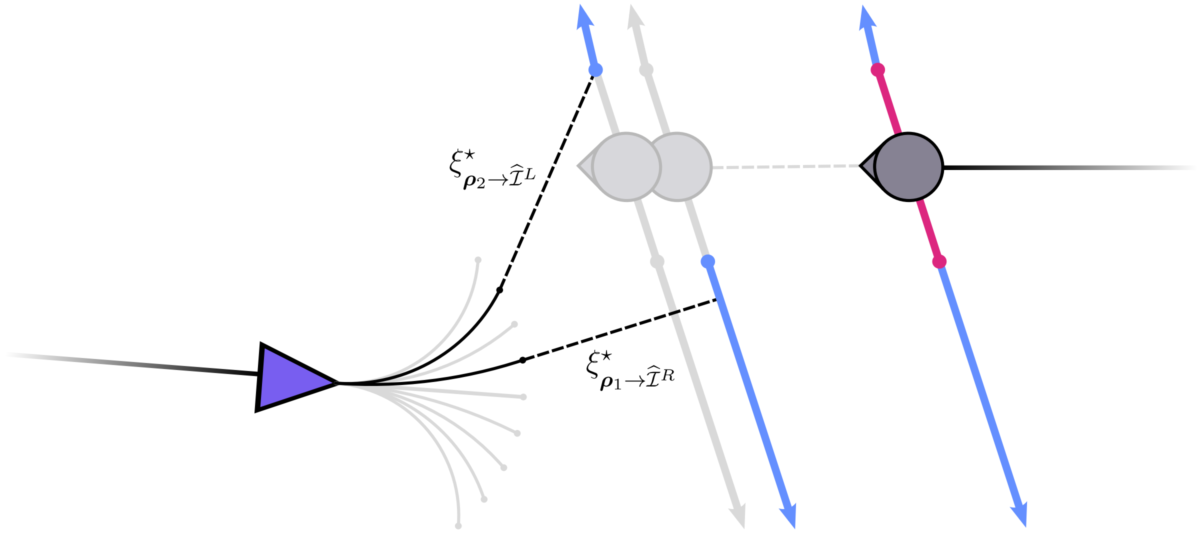

$\xi^\star_{\boldsymbol{\rho}_2\to\widehat{\mathcal{I}}^L}$
$\xi^\star_{\boldsymbol{\rho}_1\to\widehat{\mathcal{I}}^R}$

Figure 3.8: *Visualization of two motion primitives, ρ_1 and ρ_2, when each are aligned with the optimal trajectory to different interaction regions.*

Caution should be taken when evaluating motion primitives that intersect the interaction region. When the primitive crosses the interaction line, only the portion up to the intersection should be considered in the cost. For example, in Figure 3.9, the dashed portion of the primitive has already reached the right interaction region. The solid portion of the primitive should be included in the cost, whereas the dashed portion has zero cost. Therefore, depending on whether or not primitive ρ crosses the interaction line, the interaction cost, $c_{\mathcal{I}}[\xi_{\boldsymbol{t}\to\rho}]$, appearing in (3.22) and (3.23) is expanded as follows:

$$c_{\mathcal{I}}[\xi_{\boldsymbol{t}\to\rho}] = \begin{cases} c[\xi_{\boldsymbol{t}\to\boldsymbol{\rho}}] + c[\xi^{\star}_{\boldsymbol{\rho}\to\mathcal{I}}] & \text{if } \rho \cap \mathcal{I} = \emptyset \\ c[\xi^{\rho}_{\boldsymbol{t}\to\mathcal{I}}] & \text{otherwise,} \end{cases} \tag{3.24}$$

where $\xi^{\rho}_{\boldsymbol{t}\to\mathcal{I}}$ is the partial primitive along ρ up to the intersection with the interaction region and $c[\xi^{\rho}_{\boldsymbol{t}\to\mathcal{I}}] \leq c[\xi_{\boldsymbol{t}\to\boldsymbol{\rho}}]$.

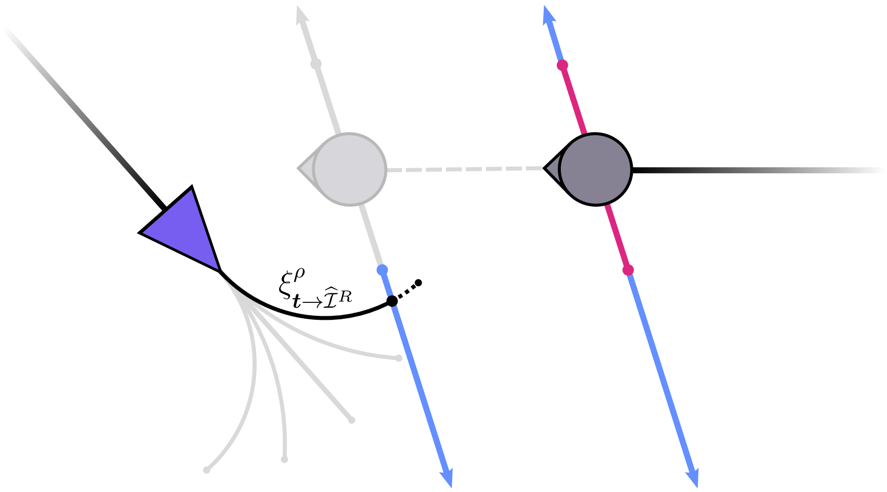
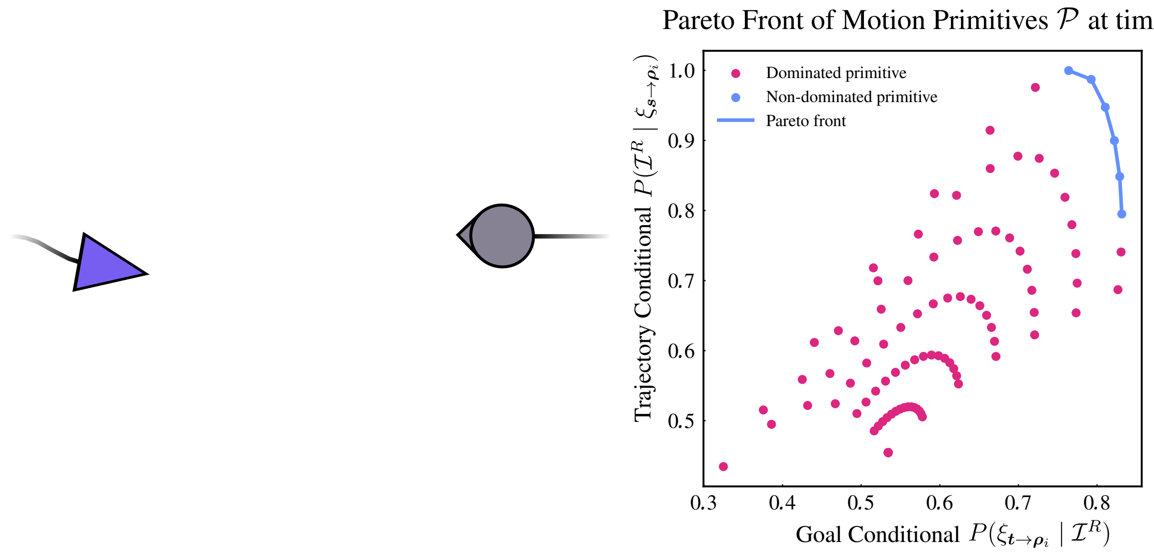


Figure 3.9: *A motion primitive intersecting the interaction line.*

3.7 Optimizing Legibility and Predictability

In planning motion, the robot must balance between the objectives of legibility and predictability. Strictly optimizing for one of the objectives may lead to behaviors that are too unpredictable [27] or will not sufficiently express intent. It is therefore important to consider both objectives in the optimization scheme. Ideally, the robot should indicate its intention when its motion is unclear to the observer, otherwise, the robot should proceed predictably. However, without any means of communication, detecting how the other agent perceives the robot's motion is not an easy task. Since the trajectory (3.22) and goal (3.23) conditional are of different scales, we can generate the Pareto front, to which an efficient primitive can be selected. The Pareto front is the set of all Pareto efficient solutions. A Pareto efficient solution is a primitive $\rho_i \in \mathcal{P}$ to which there exists no primitive $\rho_j \in \mathcal{P}$ with improved trajectory and goal conditional. In other words, an objective can only be improved at the expense of worsening another. An example of the Pareto front of the robot's primitives $\mathcal{P}$ during a swap scenario is shown in Figure 3.10. An optimal primitive can be selected from the Pareto front by trading-off between the goal and trajectory conditional.



(a) Robot (purple) and other agent (gray) trajectories

(b) Pareto optimal primitives

Figure 3.10: *Pareto front (b) from the set of robot motion primitives $\mathcal{P}$ for the configuration in (a).*

Choosing an efficient primitive from the Pareto front works well in the single agent case. However, as will be explained in the next section, the Pareto front quickly degenerates to the entire set of primitives $\mathcal{P}$ in the multi-agent case. A first attempt to balance between the objectives of legibility and predictability is to scalarize the objectives and take a convex combination of these inferences as follows:

$$\rho^\star = \underset{\rho_i \in \mathcal{P}, \mathcal{I} \in \boldsymbol{\mathcal{G}}^P}{\operatorname{argmax}} (1 - \lambda(t)) P(\mathcal{I} \mid \xi_{\boldsymbol{s} \to \boldsymbol{\rho}_i}) + \lambda(t) P(\xi_{\boldsymbol{t} \to \boldsymbol{\rho}_i} \mid \mathcal{I}^\star), \tag{3.25}$$

where $\lambda \colon [t_s, t_f] \to [0, 1]$. Here, λ monotonically increases with time and does not depend on the interaction itself. In cases where the robot is not interacting with another agent, as per Section 3.2, the robot simply optimizes for its global goal $\boldsymbol{g}_R \in \mathcal{W}$.

The effect of different values of λ on the robot's trajectories is illustrated in Figure 3.11. Small values of λ weigh the trajectory conditional more heavily. For these smaller values, the trajectory more clearly expresses the robots intended passing side. Larger values of λ weigh the goal conditional more heavily. In this case, more efficient trajectories are favored, as shown by the straighter paths.

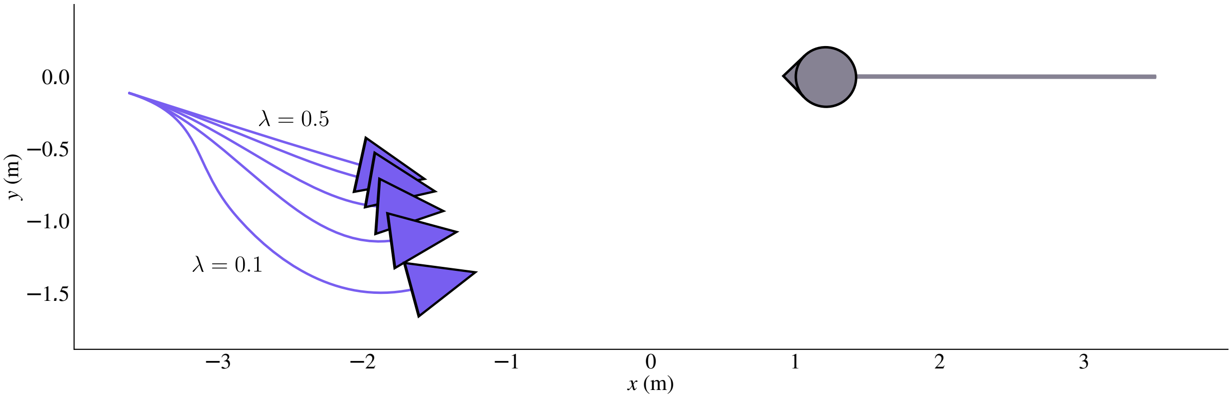


Figure 3.11: *Trajectories during a navigation interaction for* $\lambda \in \{0.1, 0.2, \ldots, 0.5\}$.

Rather than blindly increasing lambda in (3.25), we choose to track the ambiguity as the interaction progresses and set lambda accordingly. Certain scenarios are more ambiguous than others; the authors in [27] state that scenarios where the legibility of the predictable trajectory is lower are more ambiguous. In order to promote legible motion when the robot's intentions are ambiguous and predictable motion otherwise, we set

$$\lambda(\mathcal{I}, \xi_{\boldsymbol{s} \to \boldsymbol{t}}) = \max\left(0, \min\left(1, \frac{\alpha(\mathcal{I}, \xi_{\boldsymbol{s} \to \boldsymbol{t}}) - a_L}{a_P - a_L}\right)\right) \tag{3.26}$$

where $\alpha(\mathcal{I}, \xi_{\boldsymbol{s} \to \boldsymbol{t}}) = |P(\mathcal{I}^L \mid \xi_{\boldsymbol{s} \to \boldsymbol{t}}) - P(\mathcal{I}^R \mid \xi_{\boldsymbol{s} \to \boldsymbol{t}})|$ and a_L and a_P are scalar parameters that determine the values of α at which the robot should strictly optimize legibility or predictability. Predictability is often required to convey intent [27]. In fact, strictly optimizing the legibility functional can lead to arbitrarily unpredictable motions [84]. We recommend setting $a_L < 0 < a_P$ to safeguard the robot from becoming too unpredictable. Therefore, when a passing side is much more likely than the other, λ approaches 1 (weighing primitives that match expectation more heavily), whereas if they are approximately equal, λ approaches 0 (weighing primitives that convey a passing side more heavily).

The motion planner for the single-agent case is outlined in Algorithm 1. At each iteration, the robot first passes its motion primitives $\mathcal{P}$ through a collision checker. Since our problem is formulated with convex agents, this allows the use of a linear-time collision detection algorithm. The interaction line is then updated and used to predict the time-to-interaction using (3.20). If the agent is interacting, as defined in 3.2.1, the planner proceeds to find the optimal primitive for the interaction. After making a prediction over the horizon t_p of the other agent's position, the goal and trajectory conditionals are computed for each primitive $\rho \in \mathcal{P}^\star$ using (3.22) and (3.23) respectively. The weighting parameter λ is then computed using (3.26) to obtain the optimal primitive for the interaction in (3.25). If, on the other hand, the robot was not interacting with the other agent, it would simply optimize for its global goal $\boldsymbol{g}_R \in \mathcal{W}$, i.e. $\rho^\star = \text{argmin}_{\rho_i \in \mathcal{P}} \|\boldsymbol{g}_R - \boldsymbol{\rho}_i\|$.

3.8 Multi-Agent Adaptation

Although generating the Pareto front works well in the single agent case, it is not as effective in the multi-agent case. With two objectives (trajectory and goal conditional) for each additional agent in the environment, the Pareto front quickly degenerates to the entire set of primitives. Intuitively this makes sense since the Pareto front can only increase or remain the same with additional agents. With many agents, a primitive is less likely to be Pareto dominated, that is, there exists another primitive with improved objectives for all the agents.

Here, we propose an adaptation that extends (3.25) to multiple dynamic agents; the adaptation reduces to (3.25) when optimizing over a single agent. We assign to each agent A in the set of interacting agents $\mathcal{A}$, its own corresponding interaction regions $\mathcal{I}_A(t) \in \boldsymbol{\mathcal{G}}_A(t)$ and lambda parameter $\lambda_A(t)$. To penalize robot motions that are illegible to others, we optimize over $\mathcal{A}$ by maximizing the minimum score:

$$\rho^\star = \underset{\rho_i \in \mathcal{P}}{\text{argmax}} \left\{ \min_{A \in \mathcal{A}} \left\{ \max_{\mathcal{I}_A \in \boldsymbol{\mathcal{G}}_A} (1 - \lambda_A) P(\mathcal{I}_A \mid \xi_{\boldsymbol{s}_A \to \boldsymbol{\rho}_i}) + \lambda_A P(\xi_{\boldsymbol{t} \to \mathcal{I}_A} \mid \mathcal{I}_A) \right\} \right\}. \tag{3.27}$$

Algorithm 1: Motion Planner

```
Input: robot $R$, set of motion primitives $\mathcal{P}$, other agent $A$, maximum interaction
       time $t^{\mathcal{I}}_{\max}$, robot goal $\boldsymbol{g}_R$, prediction horizon $t_p$
Output: optimal primitive $\rho^\star$
$\mathcal{P}^\star \leftarrow$getCollisionFree($\mathcal{P}, R, A$)
$\mathcal{I} \leftarrow$updateInteractionLine($R, A, \boldsymbol{g}_R$)
$\widehat{t} \leftarrow$predictInteraction($R, A, \mathcal{I}$)                    // computes (3.20)
if isInteracting($\widehat{t}, t^{\mathcal{I}}_{\max}, R, P, \boldsymbol{g}_R$)                // as per Definition 3.2.1
 then
    $\widehat{A} \leftarrow$predictPos($A, t_p$)                                // depends on (3.10)
    $s_L \leftarrow$getLegibilityScore($\mathcal{P}^\star, \widehat{A}$)                  // computes (3.22)
    $s_P \leftarrow$getPredictabilityScore($\mathcal{P}^\star, \widehat{A}$)              // computes (3.23)
    $\lambda \leftarrow$getLambda($R, A, \mathcal{I}, s_L, s_P$)                      // computes (3.26)
    $\rho^\star \leftarrow$getOptPrim($\mathcal{P}^\star, s_L, s_P, \lambda$)                   // computes (3.25)
else
    $s_G \leftarrow$getGoalScore($\boldsymbol{g}_R, \mathcal{P}^\star$)
    $\rho^\star \leftarrow$getOptPrim($\mathcal{P}^\star, s_G$)
end
return $\rho^\star$
```

To determine the set of interacting agents, the robot uses their current velocities to predict the time to complete the interactions. Any agent for which the predicted time-to-interaction falls below a predefined value $t^{\mathcal{I}}$ will be considered interacting. The interaction is complete once the robot crosses the interaction line.

We should note that other strategies to deal with multiple agents could also be used. A simple open loop strategy could make an initial estimate of the interaction duration, and use this to linearly increment λ. The added complexity in the environment motivates other interesting strategies that could incorporate feedback of each agent's motion in the interaction. Doing so would allow the robot to reason about the other agent's inference. For example, given that its legibility score is not above a specified threshold, it would attempt to further increase its legibility before optimizing predictability. Another interesting strategy would be to detect at which point in the interaction the other agent acknowledges the robot's intention, that is, the point in which the robot's true interaction intention $\mathcal{I}^{\star}$ becomes known. Again, if we assume the other agent cannot directly communicate this, one method to detect the other agent's acknowledgment of the robot's intention would be to watch for changes in its velocity, indicating cooperation in the interaction. A bang-bang control strategy could then use the intention's detection time to switch from optimizing legibility to predictability by changing $\lambda = 0$ to $\lambda = 1$.

Further improvements include an environment-dependent optimization strategy. This would allow the robot to adapt its behavior based on the crowd density and the surrounding obstacles in the environment. For example, more importance could be placed on nearby agents. In doing so, illegible motions would be penalized more heavily for nearby agents than for far away agents. The interaction, as defined in Section 3.2, could also take into account the structure of the environment. For example, the collision radius, r_C, could be made adaptable to the density of the crowd. In low-density spaces, the collision radius is increased to give more clearance to nearby agent, which has been shown to correlate to an increased feeling of comfort [70]. In high-density crowds, the collision radius is decreased to allow the robot to make continued progress in the environment. In the results that follow, we make r^C from (3.3) adapt to the density of the crowd.

Chapter 4

Results

Existing works that claim to generate legible navigation often validate their motion through qualitative results and user studies. In order to compare against other approaches, they resort to some of the more classical planning metrics. These include path efficiency, collision rate and success rate. The average acceleration along a trajectory and complexity index have also been used in an attempt to capture legibility. The lack of standardized legibility and predictability metrics make it difficult to compare the degree to which legible and predictable motion is generated across frameworks. In this section, we suggest how the goal and trajectory inferences derived in 3.4 can be used to quantify the legibility and predictability of navigation interactions. We then evaluate the extent to which our approach (LPSNav) promotes legible behavior in ambiguous scenarios and predictable behavior in unambiguous scenarios. We also evaluate the multi-agent performance with a varying number of agents in randomly generated scenarios.

4.1 Post Hoc Trajectory Evaluation

Recall from Section 2.1, legibility is a time-dependent property of motion. The earlier the global goal is inferred, the more legible the trajectory. To compute (3.9) and (3.11) in practice, we discretize the trajectories by uniformly sampling the interval $[t_s, t_f]$ with time-step Δt. A discrete-time trajectory is therefore given by the vector of configurations $\xi_{\boldsymbol{s}\to\boldsymbol{t}} = [\boldsymbol{p}_{R_0}, \boldsymbol{p}_{R_1}, \dots, \boldsymbol{p}_{R_N}]^\intercal$, where $N := \frac{\lfloor t_f - t_s \rfloor}{\Delta t}$ and $\boldsymbol{p}_{R_k} = \boldsymbol{p}_R(t_s + k\Delta t)$, $k \in \{0, 1, \dots, N\}$. Although the trajectory and goal conditionals are used to model the observer's inferences, they don't directly express how legible or predictable a trajectory is. For this reason, it would be desirable to have a score for each of these properties, allowing us to evaluate either partial or entire trajectories. These scores are obtained by tracking (3.9) and (3.11) along a robot's trajectory. We adopt a discrete version of the legibility score proposed in [24], where goals are now considered to be dynamic interaction regions. The legibility of a trajectory with respect to the interaction $\mathcal{I} \in \boldsymbol{\mathcal{G}}(t)$ is given by:

$$\text{legibility}_{\mathcal{I}}(\xi_{\boldsymbol{s}\to\boldsymbol{t}}) = \frac{\sum_{k=0}^{N} P(\mathcal{I} \mid \xi_{0:k})(N-k)}{\sum_{k=1}^{N}(N-k)}, \tag{4.1}$$

where $\xi_{0:k}$ denotes the discretized trajectory from t_s to $t_s + k\Delta t$. The above score can be used to evaluate either partial or complete trajectories and uses a discount in time to weigh earlier parts of the trajectory more heavily. Since predictability does not evolve with time, we use (3.11) to evaluate the predictability of a trajectory segment with respect to an interaction region $\mathcal{I}(t) \in \boldsymbol{\mathcal{G}}(t)$:

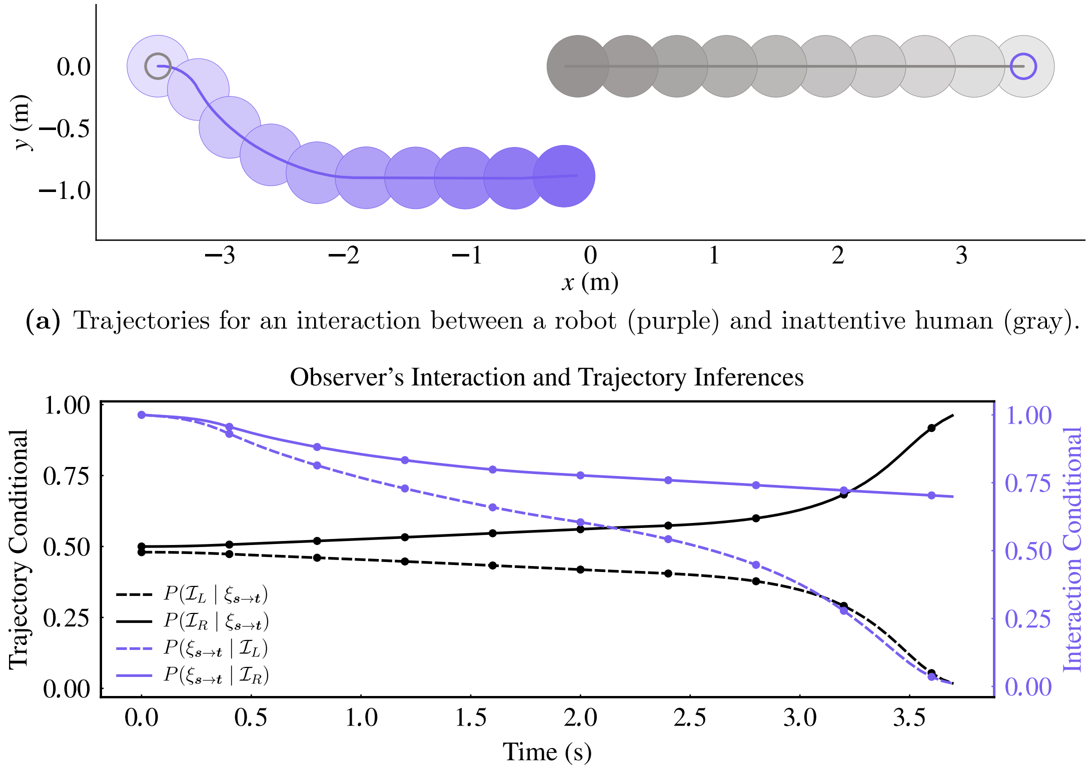
$$\text{predictability}_{\mathcal{I}}(\xi_{\boldsymbol{s}\to\boldsymbol{t}}) = \exp\Big(\beta\big(c[\xi^{\star}_{\boldsymbol{s}\to\widehat{\mathcal{I}}}] - c_{\widehat{\mathcal{I}}}[\xi_{\boldsymbol{s}\to\boldsymbol{t}}]\big)\Big). \tag{4.2}$$

An illustrative example is given in Figure 4.1. Computing (4.1) for each passing interaction in Figure 4.1a gives $\text{legibility}_{\mathcal{I}_R}(\xi_{\boldsymbol{s}\to\boldsymbol{f}}) = 54\%$ and $\text{legibility}_{\mathcal{I}_L}(\xi_{\boldsymbol{s}\to\boldsymbol{f}}) = 44\%$. Similarly, computing (4.2) gives $\text{predictability}_{\mathcal{I}_R}(\xi_{\boldsymbol{s}\to\boldsymbol{f}}) = 70\%$ and $\text{predictability}_{\mathcal{I}_L}(\xi_{\boldsymbol{s}\to\boldsymbol{f}}) = 1\%$.

4.2 Implementation Details

We compare our framework to the following approaches:

- *Optimal Reciprocal Collision Avoidance (ORCA)* [95]: a collision-free navigation framework (assuming homogeneous agents) that minimizes the effort spent by minimally adjusting each agent's velocity,
- *Social Force Model (SFM)* [42]: a model that captures social interactions as a sum of forces resulting from the environment,
- *Social Momentum (SM)* [65]: a planning framework aimed at generating motion that clearly communicates an agent's intended collision avoidance strategy,
- *SA-CADRL* [20]: state-of-the-art socially aware DRL collision avoidance navigation framework,



(a) Trajectories for an interaction between a robot (purple) and inattentive human (gray).

(b) Goal and trajectory inferences for passing the human on the right and on the left.

Figure 4.1: *Observer's inferences tracked along an interaction where each marker in (b) corresponds to one of the linearly sampled positions in (a).*

- *GA3C-CADRL* [29]: adaptation of [20] to deal with an arbitrary number of agents.

Agents are simulated as either inattentive, where they take the straight path to goal at their maximum speed, or as attentive, modeled using the ORCA framework [95], allowing cooperation in the interaction. The mobile robot is modeled as a second-order unicycle [53]:

$$\begin{aligned} \dot{x} &= v\cos\theta, \quad \dot{v} = u_a \\ \dot{y} &= v\sin\theta, \quad \dot{\omega} = u_\alpha \\ \dot{\theta} &= \omega, \end{aligned} \tag{4.3}$$

with $\boldsymbol{u} = [u_a, u_\alpha]^\intercal$ the translational and angular acceleration inputs respectively. We constrain $|u_a| \in [0, u_a^{\max}]$ and $|u_\alpha| \in [0, u_\alpha^{\max}]$ where $u_a^{\max}$ and $u_\alpha^{\max}$ are finite bounds, which we set to 3m/s^2 and 5rad/s^2 respectively. The simple unicycle model is shown in

Figure 4.2. The set of motion primitives $\mathcal{P}$ is discretized by n speeds and m headings as shown in Figure 4.3.

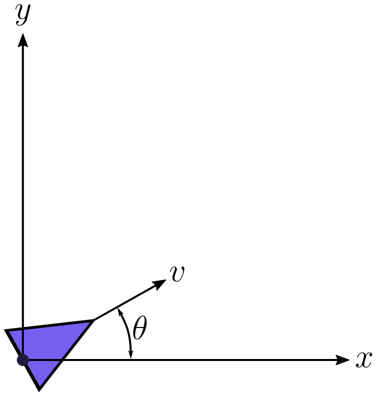


Figure 4.2: *Simple unicycle model.*

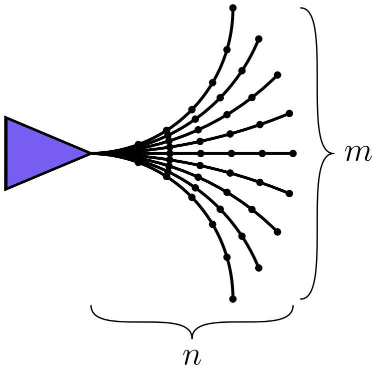


Figure 4.3: *Robot's set of motion primitives $\mathcal{P}$, discretized by n speeds and m headings.*

The LPSNav agents are configured with $r^C \in [0.35, 0.65]$m, $t_s = 2$s, $a_L = -0.02$ and $a_P = 0.5$. Passing on the right is given a higher prior to adhere to social norms. The robot's motion primitives have a fixed duration of 1s and are discretized by 5 speeds in the range $[0\text{m/s}, 1\text{m/s}]$ and 31 headings in the range $[-\frac{\pi}{4}, \frac{\pi}{4}]$.

4.3 Qualitative Results

4.3.1 Basic Scenarios

Six basic scenarios with an inattentive agent (gray) are overlaid in Figure 4.4. To emulate initially ambiguous and unambiguous scenarios, the first four scenarios are initialized on a collision course whereas the last two are not. The overtake scenario is considered ambiguous, however, differs from the others in that the robot is not initialized in the other agent's field of vision. We omit the ORCA and SFM trajectories to avoid cluttering the plots, but report their performance in the next section.

Across all ambiguous scenarios, our policy and SM indicate their intention early. Our approach respects the passing side convention in the swap scenario and maintains a more conservative behavior in the t-junction by passing from behind. The DRL approaches exhibit a more aggressive swerve later in the interaction to avoid a collision, which specifically in the t-junction results in roundabout trajectories. This could be due to their encoding of other agents' cooperativeness.

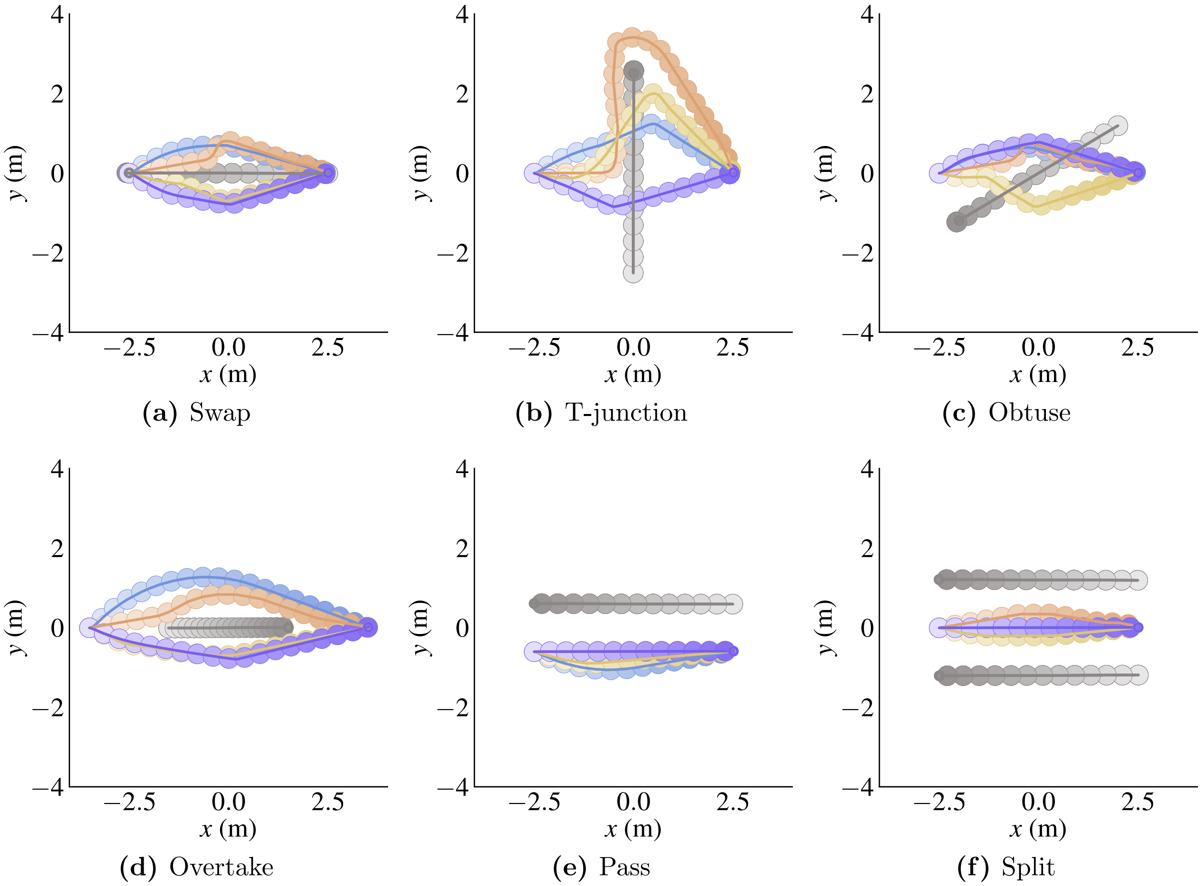
In the t-junction scenario, we note that our policy becomes predictable much quicker than in the swap scenario. The reason for this is the rate at which the λ parameter is increased in (3.26). Here, the ratio between the conditionals of passing in behind to in front increases much more rapidly than the ratio of passing on the right to passing on the left in the swap scenario.

Although the overtake scenario is initialized on a collision course, the robot is not in the other's field of vision. Our policy and SA-CADRL exhibit similar behaviors and overtake the other agent in a predictable manner. SM is shown to swerve unnecessarily wide, whereas GA3C-CADRL has a more aggressive swerve later to overtake.

In the unambiguous scenarios, our policy chooses the straight path to goal, suggesting predictable behavior. The other policies compromise their goal-efficiency by needlessly seeking to increase their legibility, which in the split scenario reduces the legibility to the third agent. Since a passing side is initially sufficiently likely in both scenarios, our policy strictly optimizes predictability.

4.3.2 Antipodal Circle Swap

Our next set of experiments evaluate how well the proposed method performs in complex scenarios, such as the congestion that arises in homogeneous antipodal circle swaps. We



(a) Swap **(b)** T-junction **(c)** Obtuse

(d) Overtake **(e)** Pass **(f)** Split

Figure 4.4: *Overlay of the trajectories resulting from basic scenarios for the LPSNav (purple), SM (blue), SA-CADRL (yellow) and GA3C-CADRL (orange) policies with an inattentive agent (gray).The circles, who's sizes match the agents' radii, darken as the simulation progresses.*

report the trajectories for 4, 5, 8 and 13 agent configurations in Figure 4.5. Since our policy places a higher prior for passing on the right than on the left, it produces a counter clockwise avoidance strategy in each configuration. Our policy and GA3C-CADRL are able to resolve the congestion in an efficient manner across the varying number of agents. SM is less efficient in the 13 agent configuration than in the fewer agent configurations. SA-CADRL struggles to deal with the 8 and 13 agent configurations. It is limited in that its trained network can accept at most 3 nearby agents, which explains the degradation in performance as the number of agents increases.

4.4 Quantitative Results

To quantify the trade-off between legibility and goal-efficiency, we would ideally need access to the underlying inference being run by the observer on the robot's avoidance strategy. Since we cannot determine this in simulation, we use a combination of the minimal predicted distance (MPD) [75] and the extra distance traveled beyond the straight path as a proxy for this trade-off. The MPD is a continuous function of time, where at instant t, MPD(t) represents the minimum distance attained between the agents if they were to continue at their current velocities

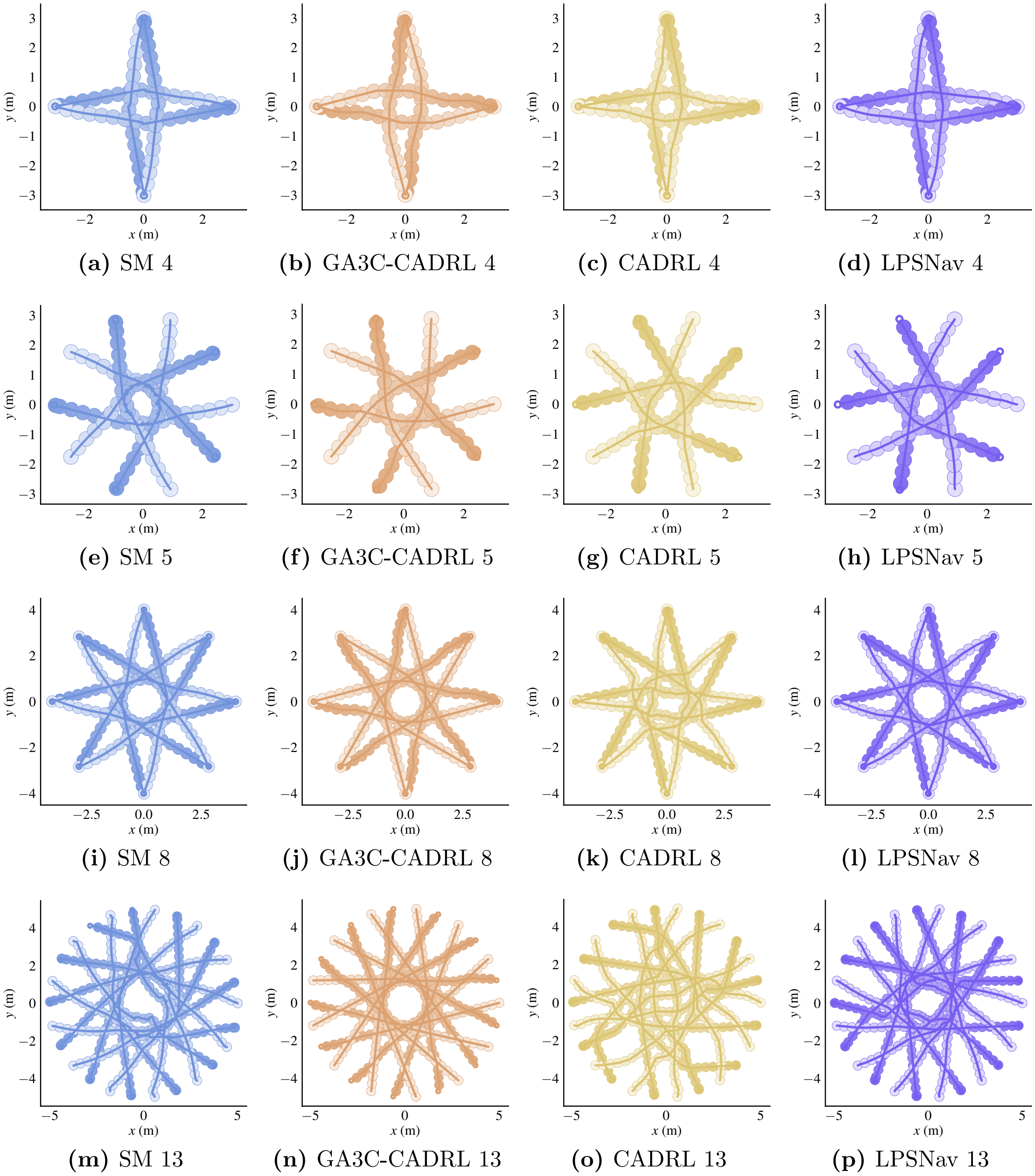
$$\mathrm{MPD}(t) = \operatorname*{argmin}_{u} \|\widehat{\boldsymbol{p}}_R(t) - \widehat{\boldsymbol{p}}_A(t)\|, \tag{4.4}$$

where $\widehat{\boldsymbol{p}}_i(t) = \boldsymbol{p}_i(t) + (u-t)\boldsymbol{v}_i(t), i \in \{R, A\}$ and u is a time parameter. In [75], they found that humans adapt their motion only if it is required, that is, when the MPD falls below a threshold of 1m.

4.4.1 Basic Scenarios

We track the MPD (Figure 4.6) and measure the average extra distance and minimum distance to the other agent (Table 4.1) across 100 random configurations with an attentive agent centered around the basic scenarios from Figure 4.4.

In the ambiguous swap scenario, LPSNav and SM have the largest initial deviation to express their intent, as indicated by the rapid increase in the MPD. SA-CADRL, GA3C-CADRL and SFM exhibit more subtle initial deviations and are required to perform a last minute avoidance maneuver, as indicated by the late peak in MPD. Although ORCA also has a subtle initial deviation, it does not swerve late and passes very close to the other



(a) SM 4 **(b)** GA3C-CADRL 4 **(c)** CADRL 4 **(d)** LPSNav 4

(e) SM 5 **(f)** GA3C-CADRL 5 **(g)** CADRL 5 **(h)** LPSNav 5

(i) SM 8 **(j)** GA3C-CADRL 8 **(k)** CADRL 8 **(l)** LPSNav 8

(m) SM 13 **(n)** GA3C-CADRL 13 **(o)** CADRL 13 **(p)** LPSNav 13

Figure 4.5: *Trajectories from circle scenarios with 4, 5, 8 and 13 homogeneous agents.*

Table 4.1: The average extra distance and minimum distance to the other agent over 100 random swap and pass scenarios.

Policy	Extra Distance (m)		Minimum Distance (m)	
	Swap Scenario	Pass Scenario	Swap Scenario	Pass Scenario
ORCA	0.04	0.03	0.06	0.66
SFM	0.32	0.05	0.45	0.80
SM	0.22	0.13	0.35	1.08
CADRL	0.23	0.04	0.29	0.67
GA3C	0.23	0.07	0.22	0.85
LPSNav	0.25	0.03	0.46	0.72

agent (0.06m on average in the swap scenario as shown in Table 4.1). These findings are similarly observed in the obtuse scenario. In the t-junction scenario, although LPSNav and SM have the largest initial deviation, the peak in the MPD is lower than in the swap scenario. When considering the overtake scenario, SM and SA-CADRL have noticeably different behaviors. Even though the robot is not initialized in the other's field of vision, SM and SA-CADRL exhibit large initial deviations. SM specifically, continues to deviate and passes by the other agent with the largest clearance.

In the unambiguous pass scenario, the MPD is initially at an acceptable value according to [75]. LPSNav, ORCA and GA3C-CADRL are the most predictable as suggested by a smaller extra distance in Table 4.1 and relatively constant MPD value. However, GA3C-CADRL counterintuitively decreases its MPD, suggesting a decrease in legibility with respsect to its passing side. SA-CADRL and SM unnecessarily increase the MPD, something that was also observed in the overtake scenario. In the split scenario, SA-CADRL and GA3C-CADRL increase the MPD to one agent while decreasing it to the other, thus confirming the findings from the qualitative results.

We also compute the legibility and the predictability of the randomly generated basic scenarios using (4.1) and (4.2) respectively, and take an average over the 100 configurations. We performed the Mann-Whitney U test with a 95% confidence interval between LPSNav and each baseline. The p-values for the legibility and the predictability are reported in Table 4.2 and Table 4.3 respectively. LPSNav and SM are the most legible in the ambiguous swap and obtuse scenarios. In the ambiguous t-junction scenario, LPSNav and CADRL are the most legible. Furthermore, LPSNav, along with ORCA are the most predictable across the unambiguous scenarios.

Table 4.2: P-values resulting from the Mann-Whitney U test (95% CI) of the legibility between LPSNav and each baseline. Statistically significant values are shown in bold.

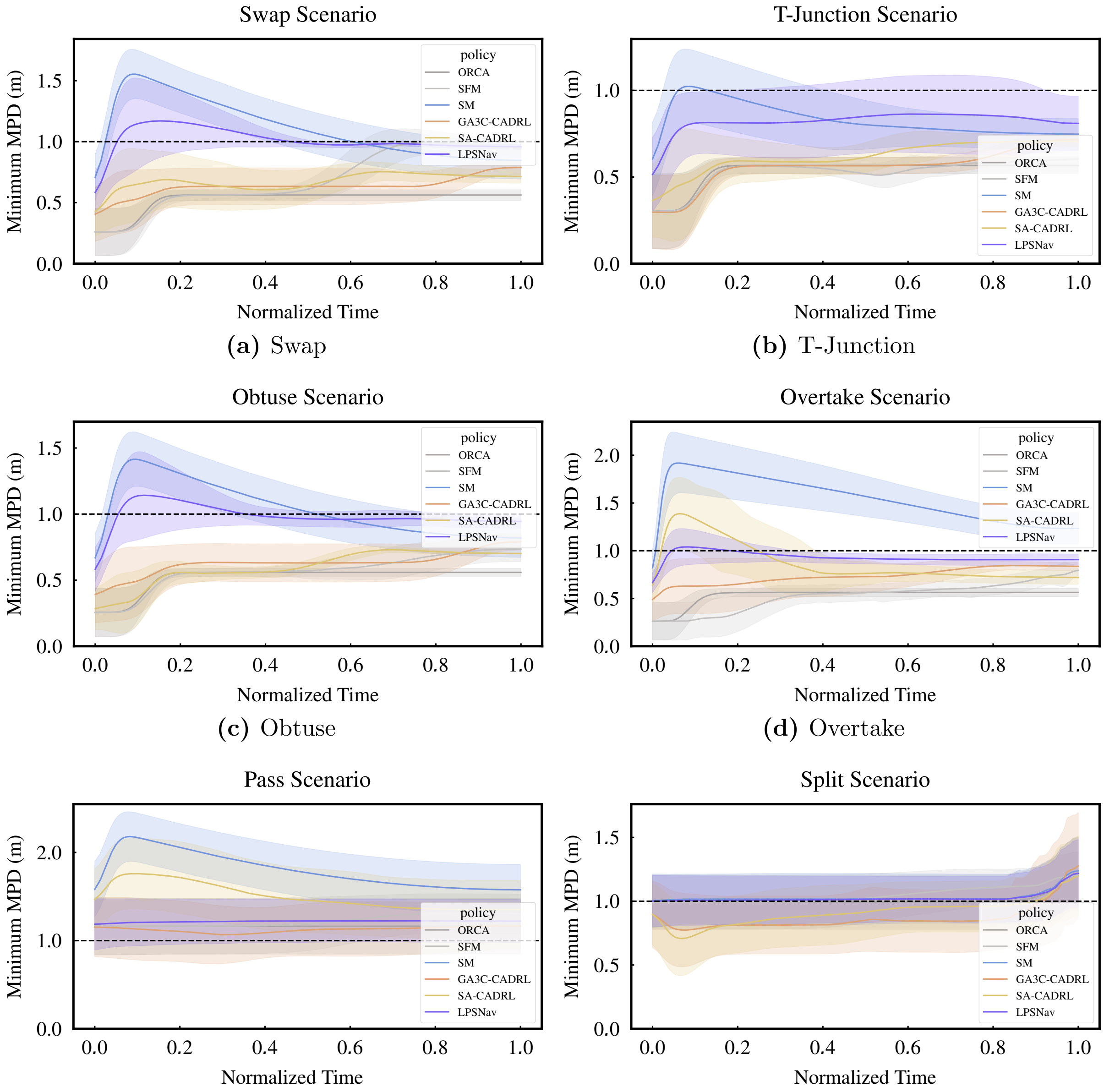
Policy	Swap	T-Junction	Pass	Split	Obtuse	Overtake
ORCA	**0.00**	**0.00**	0.60	0.99	**0.00**	**0.00**
SFM	**0.00**	**0.01**	0.97	0.96	**0.00**	**0.03**
SM	**0.02**	**0.01**	**0.00**	0.96	0.64	**0.00**
CADRL	**0.00**	0.55	**0.00**	0.78	**0.00**	**0.00**
GA3C	**0.00**	**0.00**	0.24	0.46	**0.00**	**0.00**

Table 4.3: P-values resulting from the Mann-Whitney U test (95% CI) of the predictability between LPSNav and each baseline. Statistically significant values are shown in bold.

Policy	Swap	T-Junction	Pass	Split	Obtuse	Overtake
ORCA	**0.00**	**0.00**	0.30	0.63	**0.00**	0.77
SFM	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
SM	0.23	0.14	**0.00**	0.84	0.32	**0.00**
CADRL	**0.00**	**0.02**	**0.00**	**0.00**	**0.00**	**0.00**
GA3C	0.51	**0.00**	0.24	**0.00**	0.31	0.07

4.4.2 Randomly Generated Multi-Agent Scenarios

To evaluate the multi-agent performance, we generate 100 random configurations by setting the starts and goals within an 8m × 8m area for 3, 5, 7 and 9 agents and by setting their maximum speed by randomly sampling $v_i^{\max} \sim \mathcal{N}(1.42, 0.26)$m/s [13]. We report the average extra distance travelled beyond the euclidean distance-to-goal, the failure rate, the minimum time-to-collision and the minimum distance to another agent in Figure 4.7. Our approach has a competitive goal-efficiency and scales well with the number of agents. Our approach furthermore remains collision free across all configurations. Although ORCA is the most goal-efficient, it had the smallest minimum distance to the other agents, suggesting a more aggressive behavior. As a proxy for legibility and safety, we also report the minimum time-to-collision (TTC). By indicating its intent early, our policy maintains a high minimum TTC with a varying number of agents. This, combined with an elevated minimum distance to the other agents suggests safe behavior.

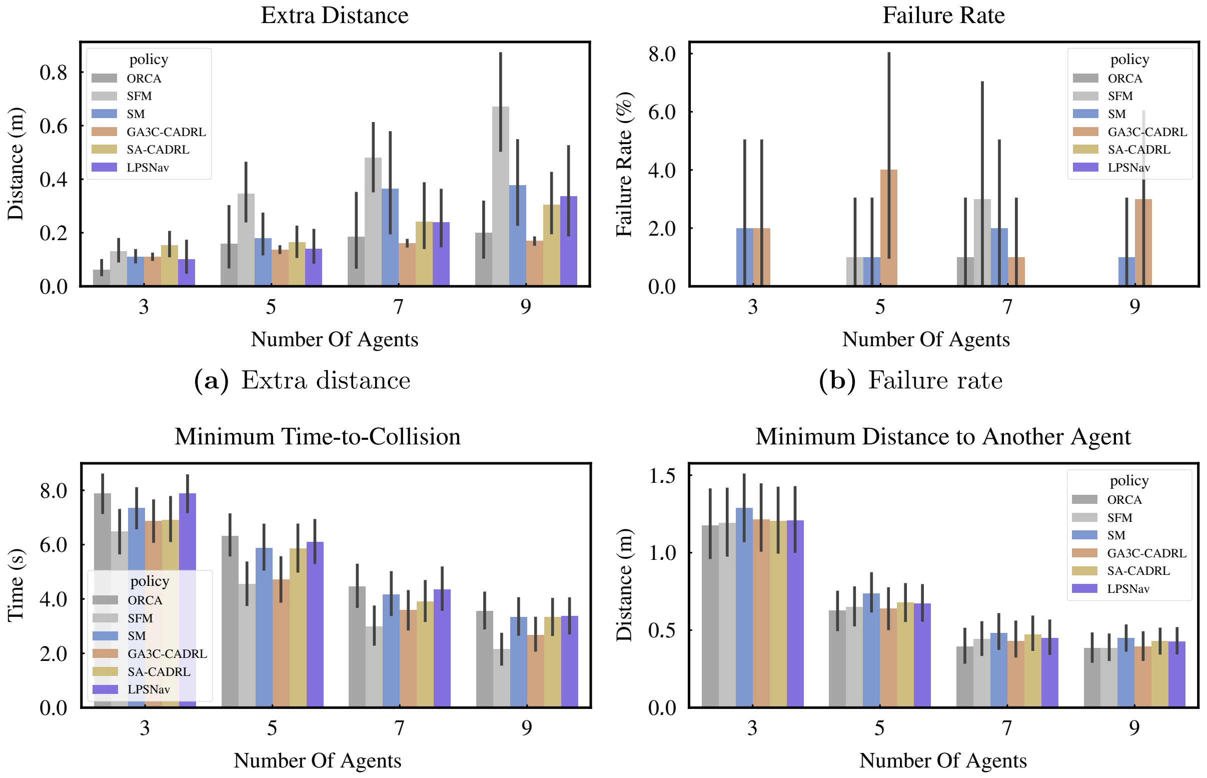


(a) Swap **(b)** T-Junction

(c) Obtuse **(d)** Overtake

(e) Pass **(f)** Split

Figure 4.6: *The minimal predicted distance (MPD) aggregated over 100 randomized basic scenarios.*



(a) Extra distance

(b) Failure rate

(c) Minimum time-to-collision

(d) Minimum distance

Figure 4.7: *Performance metrics averaged over 100 random scenarios for 3, 5, 7 and 9 agents.*

Chapter 5

Conclusions and Future Research

In this chapter, we summarize the contributions of this work. Afterwards, we discuss its main limitations and possible directions for future work.

5.1 Summary

In this thesis, we presented an approach for representing social navigation interactions as dynamic goals with which a motion planner can use to reason on its legibility and predictability with respect to a passing side. Existing navigation frameworks claiming to generate legible motion have done so by evaluating their approach through the use of user studies. Furthermore, existing works which employ the notions of legibility and predictability as derived in [24] have assumed fixed goal points. In this work, we generalized these scores to dynamic goal regions. The interaction regions, as defined in Section 3.2, are represented by a segmented line which translates according to the other agent's motion. Unlike in the static case, generalizing the trajectory (3.9) and goal (3.11) conditionals to dynamic regions requires a motion prediction model.

Given a prediction model chosen by the designer, in Section 3.5, we show how the optimal costs appearing in (3.9) and (3.11) can be obtained using the constant bearing strategy. In Section 3.6, we suggest how the legibility and predictability scores can be queried from a set of motion primitives for use in a motion planner. In the single agent case, we gave a solution which computed the Pareto front to optimize the scores. Since the Pareto front was shown to degenerate in the multi-agent case, we provided an improved strategy which chooses the primitive that maximizes the minimum score over all agents. This strategy penalizes motions with are illegible to other agents in the environment.

Our approach represents navigation interactions as dynamic goals with which a motion planner can use to reason on its legibility and predictability with respect to a passing side. In Chapter 4, these properties of motion were used to promote legible behavior in ambiguous scenarios and predictable behavior otherwise. We also tested our framework's multi-agent performance, where it is competitive with state-of-the-art approaches in terms of goal-efficiency while remaining collision-free in randomly generated scenarios.

5.2 Future Work and Open Problems

5.2.1 User Study and Hardware Implementation

Future work includes a user study to validate our framework in real-world human-robot navigation scenarios. This would allow us to analyze the legibility and goal-efficiency trade-off in human-shared environments. The main body of this work is focused on the theoretical development of social navigation interactions. Although we provide the necessary generalizations to quantify legibility and predictability during navigation, our work is limited in that we only validate in a simplified simulation environment.

State-of-the-art human motion models struggle to accurately capture human behavior during locomotion. Nonetheless, we should note that our framework does not assume that all agents are homogeneous. Furthermore, our framework does not rely on learning navigation behaviors from simulated agents. Although training on simulated agents can provide promising results in simulation, deploying into the real-world can result in significantly different behaviors. In cases where the human model does not accurately capture human behavior, deployment into the real-world could degrade the performance. Testing our framework through user-studies in real-world scenarios would incorporate the complexities of social navigation which cannot be captured in simulation.

A few extra measures should be taken to deploy our approach in practice. The first of which is the integration with a global planner. Our work focuses on the robot's behavior while interacting with nearby agents during navigation. More specifically, we presented a local trajectory planner which chooses the optimal motion primitive for the robot's immediate actions. This motion planner, as outlined in Algorithm 1, does not currently have the ability to deal with obstacles in the environment. For example, if we consider the scenario in Figure 5.1, where a wall separates the robot from its goal, the robot will get stuck and be unable to reach its goal. A successful deployment into the real-world therefore requires a global planner in addition to our local planner. One such solution is a global planner responsible for setting way-points leading to the robot's global goal. Having reached a way-point, a new one is set to which our local planner will consider to be its new goal. This is illustrated in Figure 5.1, where $\boldsymbol{g}_1$ is the intermediate way-point for which the local planner is currently considering. The global planner chooses the way-points $\boldsymbol{g}_1$, $\boldsymbol{g}_2$ and $\boldsymbol{g}_3$ such that they circumvent the wall, allowing the robot to reach its global goal $\boldsymbol{g}^\star$ while employing our local planner.

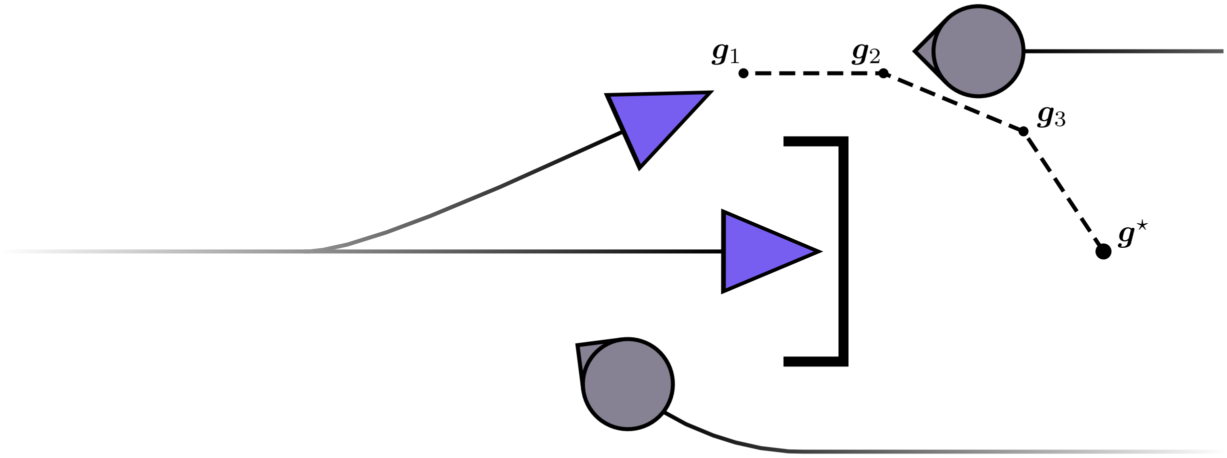


Figure 5.1: *Effect of employing a global planner or not. Without a global planner, the robot goes directly towards its goal,* $\boldsymbol{g}^\star$*, resulting in a stuck state behind the wall. With a global planner, intermediate way-points are set such that they circumvent the wall. In the figure, the local planner is currently considering the way-point* $\boldsymbol{g}_1$ *as its global goal, allowing it to employ our local planner while making further progress towards its global goal* $\boldsymbol{g}^\star$*.*

Secondly, a hardware implementation should require a software framework suitable for robot software development. Currently, our framework is implemented in Python, which is suitable for theoretical and prototyping purposes. The Robot Operating System (ROS) [79] has become the de facto standard to build robot applications. Its first iteration was widely used for research development. With a focus on security, its second generation, ROS2, was redesigned to allow reliable deployment in real-world robot applications [63]. ROS2 Navigation (Nav2) [62] would facilitate the implementation of our planner onto hardware. Its modular design allows the designer to choose an existing global planner, known as a *planner plugin*, alongside an implementation of our local planner, as a *controller plugin*. Since ROS is widely used, this implementation would allow others to easily incorporate our framework into their applications.

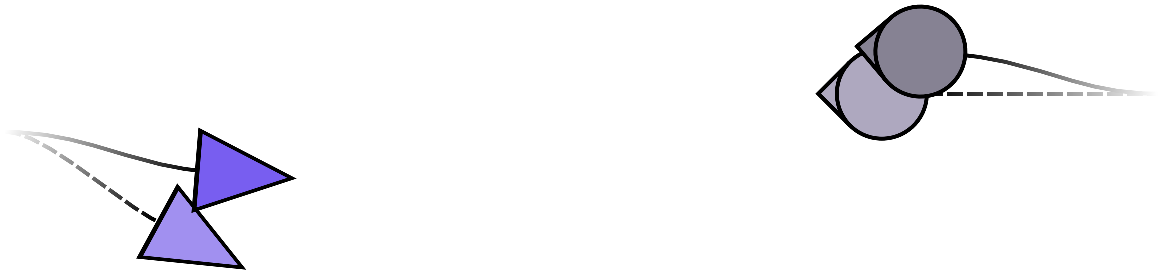


Figure 5.2: *Comparing the trajectory segments while accounting for the other agent's cooperation or not.*

5.2.2 Directions for Future Development

To consider humans as the other dynamic agents in the environment, an improved motion prediction model could aid the transition from simulation to real-world deployment. Existing learning-based prediction models use human data-sets to learn complex social interactions during navigation. That being said, caution should be taken when choosing such approach as there are potential pitfalls in training pedestrian prediction models [83]. Furthermore the complexity of social navigation also makes it difficult to extract the strategies which are learned. As opposed to a learning-based approach, explicit modelling of navigation interactions could be used to improve the prediction model. As humans navigate amongst one another, they also seek to generate motion that is legible to others.

Using our generalizations of legibility and predictability from Section 3.4, the human's cooperation in the interaction could be incorporated into the prediction model in (3.7). As a result, the robot would reason on its legibility and predictability while accounting for the human's cooperation. To illustrate what sort of behavioral changes this would cause, let us consider the swap scenario in Figure 5.2. The lighter agents represent the expected behavior assuming our the robot follows our current implementation using the CVM as the prediction model. Without taking into account the other's cooperation in the interaction, our policy would exhibit an aggressive swerve early on to indicate its intention. However, by taking into account the others cooperation (as indicated by the darker agents), the right passing inference would become more likely sooner. The robot would therefore deviate less since less aggressive motion primitives would be as intent expressive in the cooperative case as the more aggressive motion primitives in the non-cooperative case. Specifically, the λ parameter in (3.26) would increase more rapidly if the robot were to model the other agent's cooperation.

Implementing a cooperation model into our framework would require additional precautions. For example, it would be unreasonable to assume that each human the robot encounters would cooperate equally in a navigation interaction. Furthermore, the human could also be inattentive, in which case it will not cooperate in the interaction. Explicitly modelling social navigation should therefore be able to adapt to the environment and detect to what degree another human is expected and willing to participate in the interaction.

It remains unknown as to what degree a mobile robot should exhibit human-like navigation behavior. Although our work is not directly focused on generating human-like motion, it respects social conventions such as a preferred passing side. A key property of our framework is that the robot should indicate its intention early to disambiguate its navigation strategy to an observer. This is accomplished by setting λ in (3.26) depending on the goal and trajectory inferences. An alternate approach would be to learn how λ varies during human navigation from pedestrian data-sets. In doing so, λ could be set accordingly to generate motion that more closely matches that of humans.

The definition of a navigation interaction is also left unclear. For example, in our work, we use a pre-defined maximum interaction time, $t_{\mathcal{I}}^{\max}$, which determines an interaction interval (i.e. when two agents are considered to be interacting with each other). However, in practice, this is not a fixed value. For example, in social environments, humans are likely to engage in a navigation interaction with another human much sooner in a sparse environment than in a dense crowd. The multi-agent behavior of our approach could therefore be improved by making $t_{\mathcal{I}}^{\max}$ adaptive to the environment.

In Section 3.8, we made r^C from (3.3) adapt to the density of the crowd. Additional insight could be taken into account to redefine the interaction line. In our work, the collision segment is defined as the intersection between the closed disc centered at the agent's position and the interaction line. Existing works have shown that humans seek to maintain a surrounding area taking the form of an ellipse, rather than a circle [39]. In light of these findings, we could instead redefine the collision segment using a closed ellipse centered at $\boldsymbol{p}_A$ and rotated such that its major axis is aligned with $\boldsymbol{v}_H$:

$$\mathcal{I}^C(t) \coloneqq \mathcal{I}(t) \cap \left\{\boldsymbol{p} \in \mathcal{W} : \|\boldsymbol{p} - \boldsymbol{p}_A(t)\| \leq d^C(t)\right\}, \tag{5.1}$$

where

$$d^C(t) = \frac{ab}{\sqrt{(b\cos\theta(t))^2 + (a\sin\theta(t))^2}} \tag{5.2}$$

and a and b are the ellipse's major and minor axes respectively and $\theta(t)$ is the angle measured from $\boldsymbol{v}_H(t)$ to $\boldsymbol{e}(t)$. As a result, the left passing side is redefined as

$$\mathcal{I}^L(t) \coloneqq \left\{\boldsymbol{p}_A(t) + \alpha\boldsymbol{e}(t) : \alpha > d^C(t)\right\} \tag{5.3}$$

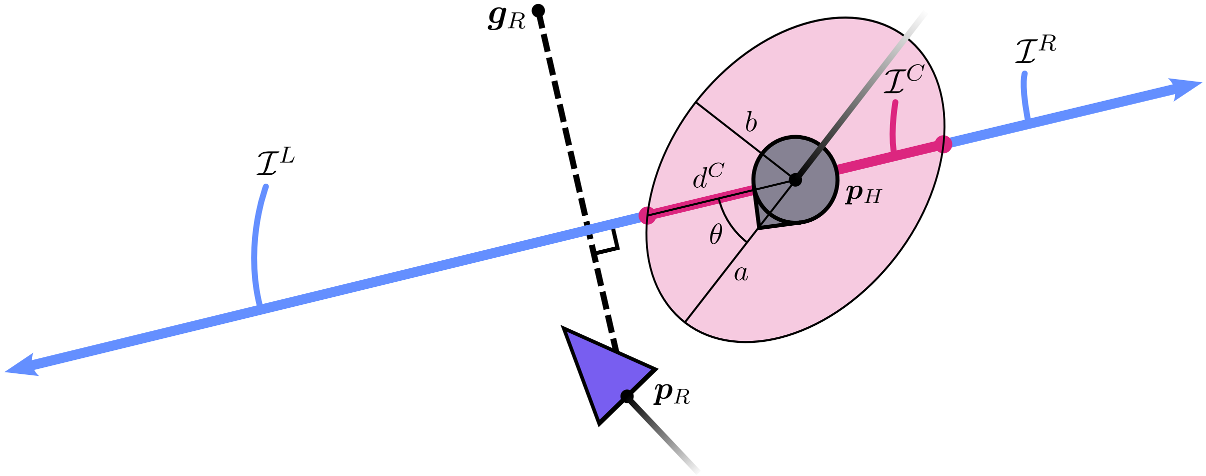


Figure 5.3: *Redefinition of the interaction line with an elliptical personal space.*

and the right passing side is redefined as

$$\mathcal{I}^R(t) \coloneqq \left\{\boldsymbol{p}_A(t) - \alpha\boldsymbol{e}(t) : \alpha > d^C(t)\right\}. \tag{5.4}$$

An example of the interaction line as defined with an elliptical personal space is shown in Figure 5.3.

The authors from [39] also found that the elliptical personal space was adjusted according to different environmental factors, such as the certainty in the other agents motion as well as its speed. Taking these findings into account, we could set the ellipsis' axes, a and b, dynamically in order to adapt to such environmental factors. Assuming $a > b$ as was determined in [39], our policy would give more space when passing in front than when passing on the side.

Further improvement to our framework includes a new cost function (3.6) for the observer's model of the robot's motion. First, the exact cost of the optimal trajectory to the interaction line can be computed as opposed to the approximation provided in Section 3.5. An exact solution would account for the interaction line's rotation in addition to its translation at the expense of additional computation. A drawback of our current approximation can be seen in the t-junction scenario in Figure 4.4 (b). Since the approximation does not account for the interaction line's rotation, the robot gets too close to the human, at which point the collision checker causes it to swerve downwards later. Second, an improved cost function could capture additional social costs. Social navigation is a complex task with many external factors which can influence behavior. For example, motions that require the robot to switch from its current passing side would incur a larger cost. In fact, the *Social Momentum* [65] cost is a good candidate to accomplish this as it represents the certainty over the pairwise avoidance strategies between agents.

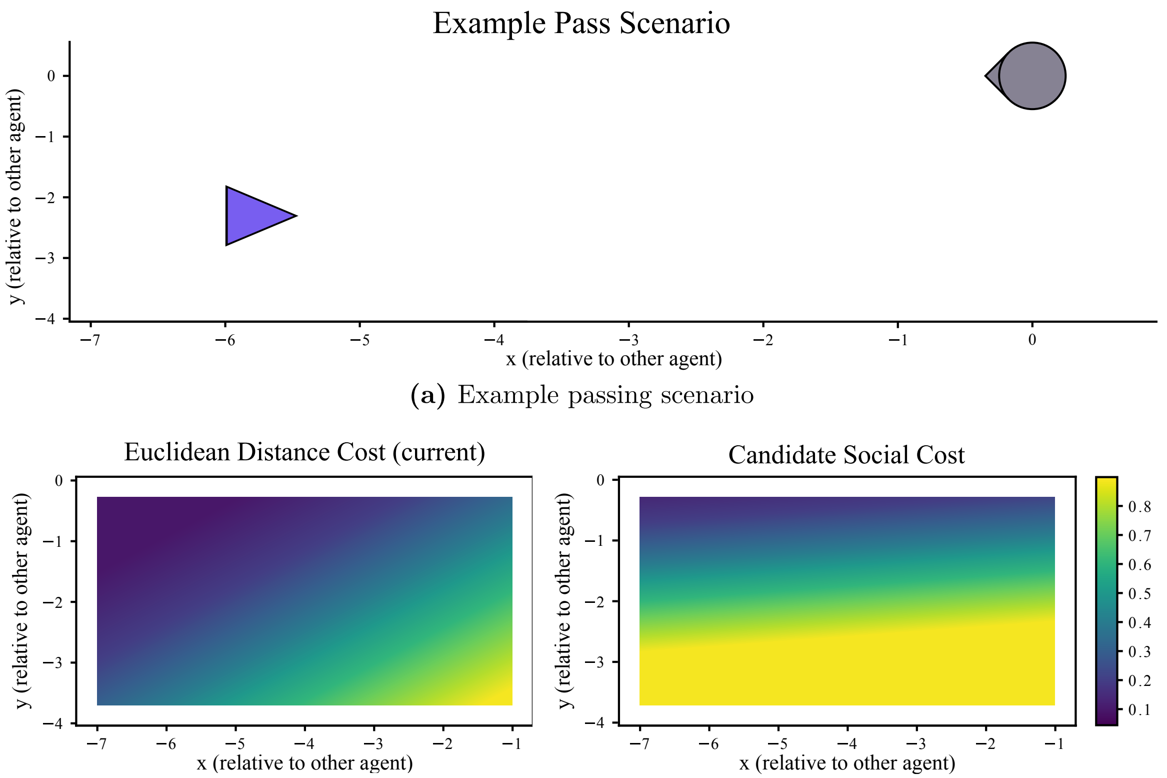
Improvements can also be made directly to the trajectory inference (2.3). In our work, the prior over the interaction goals, $P(\mathcal{I})$, $\mathcal{I} \in \boldsymbol{\mathcal{G}}$, is set assuming a certain passing side is preferred over the other to respect social conventions. However, in reality, there are additional social navigation factors in the environment which can influence these priors. For example, certain candidate global goals in the environment are more frequented than others. The prior over the interaction goals could therefore be set such that goals leading to common areas have a higher probability. These global goals such as doorways and charging stations would then influence the priors on the interaction goals.

5.2.3 Legibility and Predictability

In Section 5.2.2, we suggested that our framework could benefit from an improved cost function. This also becomes apparent when comparing the legibility and predictability of different trajectories. Let us consider a passing scenario (see Figure 5.4 (a) for an example initial configuration). Figure 5.4 (b) is a heatmap representing the difference between the right and left passing inferences for various initial configurations relative to the other agent. Given our existing time-based cost function (3.6), we notice the difference between passing inferences is small when the agents are far apart. Intuitively, this makes sense since the optimal times to reach each of the passing sides are relatively close. As a result, computing the legibility for trajectories of different initial deviations when the agents are far apart will also yield similar scores.

The two robot trajectories in Figure 5.5 are drastically different, yet their legibility as computed by (4.1) gives 54% and 60%. To make these scores more representative of the visual difference between the trajectories, we would need a new cost function for the observer's model of the robot's motion. As previously discussed, this depends on the observer's previous experience with robotic application and will likely incorporate additional social costs, rather than a simple time-based cost function. For example, the observer might model the robot such that it penalizes motions which cross its path in order to switch sides. An example of what such a candidate social cost function could look like is shown in Figure 5.4 (c). Recomputing the legibility with this candidate cost function of the trajectories in Figure 5.5 would result in a more noticeable difference between scores.

Our work is mainly focused on the legibility and predictability of a navigation interaction for the two-agent case. In Section 3.8, we propose an adaptation to the multi-agent



(a) Example passing scenario

(b) Current cost

(c) Candidate social cost

Figure 5.4: *Difference between passing inferences for various passing scenarios using our current cost function (b) and a candidate social cost function (c).*

case. In doing so, we provide an optimization strategy that maximizes the minimum score, as computed using (3.9) and (3.11) which were originally defined for the single-agent case. We have not encountered scenarios in which the robot becomes unstable while employing the proposed multi-agent optimization scheme. A formal analysis would be required to guarantee stability.

The notions of legibility and predictability when simultaneously considering multiple agents remains an open problem. As outlined in our work, the multi-agent behavior is left up to the designer. When navigating a crowd, certain trade-offs have to be made. For example, in a dense crowd, it is very likely the robot will encounter a configuration to which it cannot be legible to any of the agents. In this case, it is up to the designer to choose whether the robot should strictly optimize predictable motion or perhaps continue to try and be legible to a select few of the other agents.

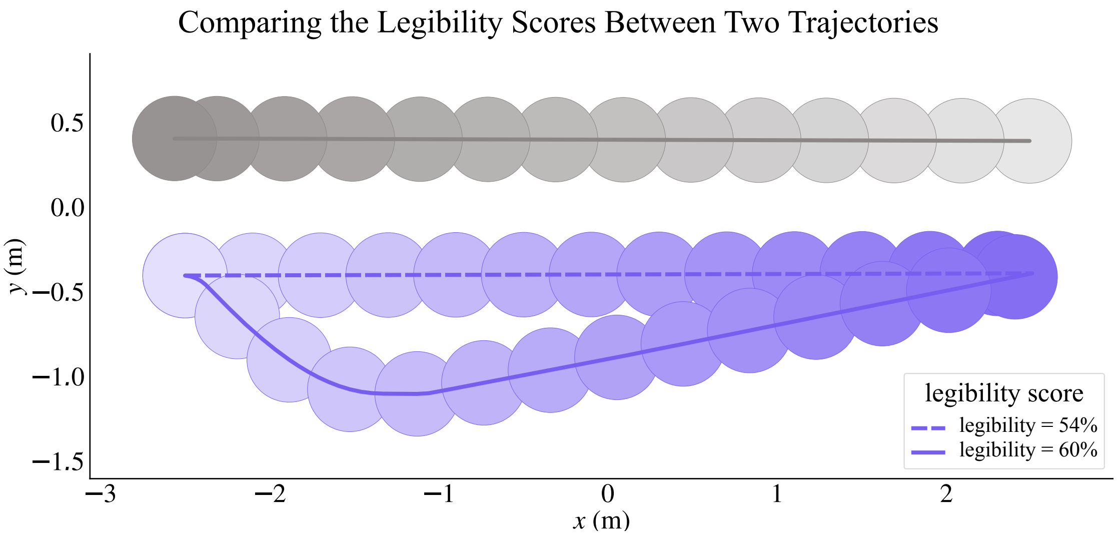


Figure 5.5: *Two different robot trajectories with their corresponding legibility scores.*

References

[1] Y. Abe and M. Yoshiki. Collision avoidance method for multiple autonomous mobile agents by implicit cooperation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 1207–1212, 2001.

[2] John R. Aiello. A further look at equilibrium theory: Visual interaction as a function of interpersonal distance. *Environmental Psychology and Nonverbal Behavior*, 1:122–140, 1977.

[3] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–971, 2016.

[4] Rachid Alami, Aurélie Clodic, Vincent Montreuil, Emrah Sisbot, and Raja Chatila. Toward human-aware robot task planning. In *AAAI Spring Symposium*, pages 39–46, 2006.

[5] Helmut Alt and Michael Godau. Computing the fréchet distance between two polygonal curves. *International Journal of Computational Geometry & Applications*, 5:75–91, 1995.

[6] Joel A E Andersson, Joris Gillis, Greg Horn, James B Rawlings, and Moritz Diehl. CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11:1–36, 2019.

[7] Chris Baker, Joshua Tenenbaum, and Rebecca Saxe. Goal inference as inverse planning. *Annual Meeting of the Cognitive Science Society*, 29:779–784, 2007.

[8] Patrizia Basili, Murat Sağlam, Thibault Kruse, Markus Huber, Alexandra Kirsch, and Stefan Glasauer. Strategies of locomotor collision avoidance. *Gait & Posture*, 37:385–390, 2013.

[9] James C. Baxter. Interpersonal spacing in natural settings. *Sociometry*, 33:444–456, 1970.

[10] Michael Beetz, Freek Stulp, Piotr Esden-Tempski, Andreas Fedrizzi, Ulrich Klank, Ingo Kresse, Alexis Maldonado, and Federico Ruiz-Ugalde. Generality and legibility in mobile manipulation: Learning skills for routine tasks. *Autonomous Robots*, 28:21–44, 2010.

[11] Stephen Bitgood and Stephany Dukes. Not another step! economy of movement and pedestrian choice point behavior in shopping malls. *Environment and Behavior*, 38:394–405, 2006.

[12] Shaunak D. Bopardikar, Stephen L. Smith, Francesco Bullo, and Joao P. Hespanha. Dynamic vehicle routing with moving demands - part i: Low speed demands and high arrival rates. In *American Control Conference*, pages 1454–1459, 2009.

[13] Raymond C. Browning, Emily A. Baker, Jessica A. Herron, and Rodger Kram. Effects of obesity and sex on the energetic cost and preferred speed of walking. *Journal of Applied Physiology*, 100:390–398, 2006.

[14] Baptiste Busch, Jonathan Grizou, Manuel Lopes, and Freek Stulp. Learning legible motion from human–robot interactions. *International Journal of Social Robotics*, 9:765–779, 2017.

[15] Beatrice Capelli, Cristian Secchi, and Lorenzo Sabattini. Communication through motion: Legibility of multi-robot systems. In *International Symposium on Multi-Robot and Multi-Agent Systems*, pages 126–132, 2019.

[16] Beatrice Capelli, Valeria Villani, Cristian Secchi, and Lorenzo Sabattini. Understanding multi-robot systems: on the concept of legibility. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 7355–7361, 2019.

[17] Daniel Carton, Wiktor Olszowy, and Dirk Wollherr. Measuring the effectiveness of readability for mobile robot locomotion. *International Journal of Social Robotics*, 8:721–741, 2016.

[18] Yuhang Che, Allison M. Okamura, and Dorsa Sadigh. Efficient and trustworthy social navigation via explicit and implicit robot–human communication. *IEEE Transactions on Robotics*, 36(3):692–707, 2020.

[19] Changan Chen, Yuejiang Liu, Sven Kreiss, and Alexandre Alahi. Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning. In *International Conference on Robotics and Automation*, pages 6015–6022, 2019.

[20] Yu Fan Chen, Michael Everett, Miao Liu, and Jonathan P. How. Socially aware motion planning with deep reinforcement learning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1343–1350, 2017.

[21] Yujing Chen, Fenghua Zhao, and Yunjiang Lou. Interactive model predictive control for robot navigation in dense crowds. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 52(4):2289–2301, 2022.

[22] Zhuo Chen, Chao Jiang, and Yi Guo. Pedestrian-robot interaction experiments in an exit corridor. In *International Conference on Ubiquitous Robots*, pages 29–34, 2018.

[23] Michael E. Cinelli and Aftab E. Patla. Travel path conditions dictate the manner in which individuals avoid collisions. *Gait & Posture*, 26:186–193, 2007.

[24] A. D. Dragan, K. C. T. Lee, and S. S. Srinivasa. Legibility and predictability of robot motion. In *ACM/IEEE International Conference on Human-Robot Interaction*, pages 301–308, 2013.

[25] Anca Dragan and Siddhartha Srinivasa. Formalizing assistive teleoperation. In *Robotics: Science and Systems*, pages 73–80, 2012.

[26] Anca D. Dragan, Shira Bauman, Jodi Forlizzi, and Siddhartha S. Srinivasa. Effects of robot motion on human-robot collaboration. In *ACM/IEEE International Conference on Human-Robot Interaction*, pages 51–58, 2015.

[27] Anca D. Dragan and Siddhartha S. Srinivasa. Generating legible motion. In *Robotics: Science and Systems*, 2013.

[28] Michael Everett, Yu Fan Chen, and Jonathan P. How. Motion planning among dynamic, decision-making agents with deep reinforcement learning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3052–3059, 2018.

[29] Michael Everett, Yu Fan Chen, and Jonathan P. How. Collision avoidance in pedestrian-rich environments with deep reinforcement learning. *IEEE Access*, 9:10357–10377, 2021.

[30] Miguel Faria, Rui Silva, Patrícia Alves-Oliveira, Francisco S. Melo, and Ana Paiva. “me and you together” movement impact in multi-user collaboration tasks. In *International Conference on Intelligent Robots and Systems*, pages 2793–2798, 2017.

[31] Gonzalo Ferrer, Anaís Garrell, and Alberto Sanfeliu. Robot companion: A social-force based approach with human awareness-navigation in crowded environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1688–1694, 2013.

[32] Paolo Fiorini and Zvi Shiller. Motion planning in dynamic environments using velocity obstacles. *The International Journal of Robotics Research*, 17:760–772, 1998.

[33] Jaime F. Fisac, Chang Liu, Jessica B. Hamrick, Shankar Sastry, J. Karl Hedrick, Thomas L. Griffiths, and Anca D. Dragan. *Generating Plans that Predict Themselves*, volume 13, pages 144–159. Springer International Publishing, 2020.

[34] D. Fox, W. Burgard, and S. Thrun. The dynamic window approach to collision avoidance. *IEEE Robotics Automation Magazine*, 4:23–33, 1997.

[35] György Gergely, Zoltán Nádasdy, Gergely Csibra, and Szilvia Bíró. Taking the intentional stance at 12 months of age. *Cognition*, 56:165–193, 1995.

[36] Francesco Giuliari, Irtiza Hasan, Marco Cristani, and Fabio Galasso. Transformer networks for trajectory forecasting. In *International Conference on Pattern Recognition*, pages 10335–10342, 2021.

[37] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2255–2264, 2018.

[38] Jérôme Guzzi, Alessandro Giusti, Luca M. Gambardella, Guy Theraulaz, and Gianni A. Di Caro. Human-friendly robot navigation in dynamic environments. In *IEEE International Conference on Robotics and Automation*, pages 423–430, 2013.

[39] Martin Gérin-Lajoie, Carol L. Richards, and Bradford J. McFadyen. The negotiation of stationary and moving obstructions during walking: Anticipatory locomotor adaptations and preservation of personal space. *Motor Control*, 9:242 – 269, 2005.

[40] Edward T. (Edward Twitchell) Hall. *The hidden dimension.* Anchor, 1966.

[41] Dirk Helbing and Péter Molnár. Social force model for pedestrian dynamics. *Physical Review E*, 51:4282–4286, 1995.

[42] Dirk Helbing and Péter Molnár. Social force model for pedestrian dynamics. *Physical Review E*, 51:4282–4286, 1995.

[43] Nicholas J. Hetherington, Elizabeth A. Croft, and H.F. Machiel Van der Loos. Hey robot, which way are you going? nonverbal motion legibility cues for human-robot spatial interaction. *IEEE Robotics and Automation Letters*, 6(3):5010–5015, 2021.

[44] Nicholas J. Hetherington, Ryan Lee, Marlene Haase, Elizabeth A. Croft, and H. F. Machiel Van der Loos. Mobile robot yielding cues for human-robot spatial interaction. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3028–3033, 2021.

[45] Mark Hollands, AE Patla, and Joan Vickers. "look where you're going!": Gaze behaviour associated with maintaining and changing the direction of locomotion. *Experimental brain research*, 143:221–30, 2002.

[46] Markus Huber, Yi-Huang Su, Melanie Krüger, Katrin Faschian, Stefan Glasauer, and Joachim Hermsdörfer. Adjustments of speed and path when avoiding collisions with another pedestrian. *PLOS One*, 9, 2014.

[47] Beomjoon Kim and Joelle Pineau. Socially adaptive path planning in human environments using inverse reinforcement learning. *International Journal of Social Robotics*, 8:51–66, 2016.

[48] Lawrence H. Kim and Sean Follmer. Generating legible and glanceable swarm robot motion through trajectory, collective behavior, and pre-attentive processing features. *ACM Transactions on Human-Robot Interaction*, 10(3), 2021.

[49] Rachel Kirby, Reid Simmons, and Jodi Forlizzi. Companion: A constraint-optimizing method for person-acceptable navigation. In *IEEE International Symposium on Robot and Human Interactive Communication*, pages 607–612, 2009.

[50] Alexandra Kirsch, Thibault Kruse, Emrah Sisbot, Rachid Alami, Martin Lawitzky, Drazen Brscic, Sandra Hirche, Patrizia Basili, and Stefan Glasauer. Plan-based control of joint human-robot activities. *KI - Künstliche Intelligenz*, 24:223–231, 2010.

[51] Thibault Kruse, Patrizia Basili, Stefan Glasauer, and Alexandra Kirsch. Legible robot navigation in the proximity of moving humans. In *IEEE Workshop on Advanced Robotics and its Social Impacts*, pages 83–88, 2012.

[52] Markus Kuderer, Henrik Kretzschmar, Christoph Sprunk, and Wolfram Burgard. Feature-based prediction of trajectories for socially compliant navigation. In *Robotics: Science and Systems*, 2012.

[53] Steven M. LaValle. *Planning Algorithms*. Cambridge University Press, 2006.

[54] Namhoon Lee, Wongun Choi, Paul Vernaza, Christopher B. Choy, Philip H. S. Torr, and Manmohan Chandraker. Desire: Distant future prediction in dynamic scenes with interacting agents. In *Conference on Computer Vision and Pattern Recognition*, 2017.

[55] Christina Lichtenthäler and A. Kirsch. Towards legible robot navigation - how to increase the intend expressiveness of robot navigation behavior. In *International Conference on Social Robotics*, 2013.

[56] Christina Lichtenthäler, Tamara Lorenz, and Alexandra Kirsch. Towards a legibility metric: How to measure the perceived value of a robot. In *International Conference on Social Robotics*, 2011.

[57] Christina Lichtenthäler, Tamara Lorenzy, and Alexandra Kirsch. Influence of legibility on perceived safety in a virtual human-robot path crossing task. In *IEEE International Symposium on Robot and Human Interactive Communication*, pages 676–681, 2012.

[58] Yuejiang Liu, Qi Yan, and Alexandre Alahi. Social nce: Contrastive learning of socially-aware motion representations. In *IEEE/CVF International Conference on Computer Vision*, pages 15118–15129, 2021.

[59] Matthias Luber, Luciano Spinello, Jens Silva, and Kai O. Arras. Socially-aware robot navigation: A learning approach. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 902–907, 2012.

[60] Yuanfu Luo and Panpan Cai. Gamma: A general agent motion prediction model for autonomous driving. *CoRR*, 2019.

[61] Sean Dean Lynch, Richard Kulpa, Laurentius Antonius Meerhoff, Julien Pettré, Armel Crétual, and Anne-Hélène Olivier. Collision avoidance behavior between walkers: Global and local motion cues. *IEEE Transactions on Visualization and Computer Graphics*, 24:2078–2088, 2018.

[62] Steve Macenski, Francisco Martín, Ruffin White, and Jonatan Ginés Clavero. The marathon 2: A navigation system. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2718–2725, 2020.

[63] Steven Macenski, Tully Foote, Brian Gerkey, Chris Lalancette, and William Woodall. Robot operating system 2: Design, architecture, and uses in the wild. *Science Robotics*, 7(66), 2022.

[64] Christoforos I Mavrogiannis and Ross A Knepper. Multi-agent path topology in support of socially competent navigation planning. *The International Journal of Robotics Research*, 38:338–356, 2019.

[65] Christoforos I. Mavrogiannis, Wil B. Thomason, and Ross A. Knepper. Social momentum: A framework for legible navigation in dynamic multi-agent environments. In *IEEE International Conference on Human-Robot Interaction*, page 361–369, 2018.

[66] Alexander R. McNeill. Energetics and optimization of human walking and running. *American Journal of Human Biology*, 14:641–648, 2002.

[67] D. Livingston McPherson and S. Shankar Sastry. An efficient understandability objective for dynamic optimal control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 986–992, 2021.

[68] Abduallah Mohamed, Kun Qian, Mohamed Elhoseiny, and Christian Claudel. Social-stgcnn: A social spatio-temporal graph convolutional neural network for human trajectory prediction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14424–14432, 2020.

[69] Ronja Möller, Antonino Furnari, Sebastiano Battiato, Aki Härmä, and Giovanni Maria Farinella. A survey on human-aware robot navigation. *Robotics and Autonomous Systems*, 145, 2021.

[70] Margot M. E. Neggers, Raymond H. Cuijpers, and Peter A. M. Ruijten. Comfortable passing distances for robots. In *Social Robotics*, pages 431–440, 2018.

[71] C.L. Nehaniv, K. Dautenhahn, J. Kubacki, M. Haegele, C. Parlitz, and R. Alami. A methodological approach relating the classification of gesture to identification of human intent in the context of human-robot interaction. In *IEEE International Workshop on Robot and Human Interactive Communication*, pages 371–377, 2005.

[72] Stefanos Nikolaidis, Anca Dragan, and Siddhartha Srinivasa. Viewpoint-based legibility optimization. In *IEEE International Conference on Human-Robot Interaction*, pages 271–278, 2016.

[73] Lauri Nummenmaa, Jukka Hyönä, and Jari K. Hietanen. I'll walk this way: Eyes reveal the direction of locomotion and make passersby look and go the other way. *Psychological Science*, 20:1454–1458, 2009.

[74] Anne-Hélène Olivier, Antoine Marin, Armel Crétual, Alain Berthoz, and Julien Pettré. Collision avoidance between two walkers: Role-dependent strategies. *Gait & posture*, 38:751–756, 2013.

[75] Anne-Hélène Olivier, Antoine Marin, Armel Crétual, and Julien Pettré. Minimal predicted distance: A common metric for collision avoidance during pairwise interactions between walkers. *Gait & Posture*, 36:399–404, 2012.

[76] E. Pacchierotti, H.I. Christensen, and P. Jensfelt. Human-robot embodied interaction in hallway settings: a pilot user study. In *IEEE International Workshop on Robot and Human Interactive Communication*, pages 164–171, 2005.

[77] Stefania Pellegrinelli, Henny Admoni, Shervin Javdani, and Siddhartha Srinivasa. Human-robot shared workspace collaboration via hindsight optimization. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 831–838, 2016.

[78] Noé Pérez-Higueras, Fernando Caballero, and Luis Merino. Learning human-aware path planning with fully convolutional networks. In *IEEE International Conference on Robotics and Automation*, pages 5897–5902, 2018.

[79] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, Andrew Y Ng, et al. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5, 2009.

[80] Andrey Rudenko, Tomasz P. Kucner, Chittaranjan S. Swaminathan, Ravi T. Chadalavada, Kai O. Arras, and Achim J. Lilienthal. ThÖr: Human-robot navigation data collection and accurate motion trajectories dataset. *IEEE Robotics and Automation Letters*, 5:676–682, 2020.

[81] Andrey Rudenko, Luigi Palmieri, Michael Herman, Kris M Kitani, Dariu M Gavrila, and Kai O Arras. Human motion trajectory prediction: a survey. *The International Journal of Robotics Research*, 39:895–935, 2020.

[82] Amir Sadeghian, Vineet Kosaraju, Ali Sadeghian, Noriaki Hirose, Hamid Rezatofighi, and Silvio Savarese. Sophie: An attentive gan for predicting paths compliant to social

and physical constraints. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1349–1358, 2019.

[83] C. Schöller, V. Aravantinos, F. Lay, and A. Knoll. What the constant velocity model can teach us about pedestrian motion prediction. *IEEE Robotics and Automation Letters*, 5:1696–1703, 2020.

[84] Elaine Short, Justin Hart, Michelle Vu, and Brian Scassellati. No fair!! an interaction with a cheating robot. In *ACM/IEEE International Conference on Human-Robot Interaction*, pages 219–226, 2010.

[85] Emrah Akin Sisbot and Rachid Alami. A human-aware manipulation planner. *IEEE Transactions on Robotics*, 28:1045–1057, 2012.

[86] Emrah Akin Sisbot, Luis F. Marin-Urias, Rachid Alami, and Thierry Simeon. A human aware mobile robot motion planner. *IEEE Transactions on Robotics*, 23:874–883, 2007.

[87] Jamie Snape, Jur van den Berg, Stephen J. Guy, and Dinesh Manocha. The hybrid reciprocal velocity obstacle. *IEEE Transactions on Robotics*, 27:696–706, 2011.

[88] Wagner Souza Silva, Gayatri Aravind, Samir Sangani, and Anouk Lamontagne. Healthy young adults implement distinctive avoidance strategies while walking and circumventing virtual human vs. non-human obstacles in a virtual environment. *Gait & Posture*, 61:294–300, 2018.

[89] William A. Sparrow and Karl M. Newell. Metabolic energy expenditure and the regulation of movement economy. *Psychonomic Bulletin & Review*, 5:173–196, 1998.

[90] Yusuke Tamura, Phuoc Dai Le, Kentarou Hitomi, Naiwala P. Chandrasiri, Takashi Bando, Atsushi Yamashita, and Hajime Asama. Development of pedestrian behavior model taking account of intention. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 382–387, 2012.

[91] Ada V. Taylor, Ellie Mamantov, and Henny Admoni. Observer-aware legibility for social navigation. In *IEEE International Conference on Robot & Human Interactive Communication*, 2022.

[92] Pete Trautman, Jeremy Ma, Richard M. Murray, and Andreas Krause. Robot navigation in dense human crowds: Statistical models and experimental studies of human–robot cooperation. *The International Journal of Robotics Research*, 34:335–356, 2015.

[93] Peter Trautman and Andreas Krause. Unfreezing the robot: Navigation in dense, interacting crowds. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 797–803, 2010.

[94] Annemarie Turnwald and D. Wollherr. Human-like motion planning based on game theoretic decision making. *International Journal of Social Robotics*, 11:151–170, 2019.

[95] Jur van den Berg, Stephen J. Guy, Ming Lin, and Dinesh Manocha. Reciprocal n-body collision avoidance. In *Robotics Research*, pages 3–19, 2011.

[96] Jur van den Berg, Ming Lin, and Dinesh Manocha. Reciprocal velocity obstacles for real-time multi-agent navigation. In *IEEE International Conference on Robotics and Automation*, pages 1928–1935, 2008.

[97] Christian Vassallo, Anne-Hélène Olivier, Philippe Souères, Armel Crétual, Olivier Stasse, and Julien Pettré. How do walkers avoid a mobile robot crossing their way? *Gait & Posture*, 51:97–103, 2017.

[98] William H. Whyte. *City: Rediscovering the Center*. University of Pennsylvania Press, 2009.

[99] Brian D. Ziebart, Nathan Ratliff, Garratt Gallagher, Christoph Mertz, Kevin Peterson, J. Andrew Bagnell, Martial Hebert, Anind K. Dey, and Siddhartha Srinivasa. Planning-based prediction for pedestrians. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3931–3936, 2009.