Deceptive Decision-Making Under Uncertainty

Yagiz Savas, Christos K. Verginis, Ufuk Topcu

The University of Texas at Austin, Austin, TX
yagiz.savas@utexas.edu, christos.verginis@austin.utexas.edu, utopcu@utexas.edu

Abstract

We study the design of autonomous agents that are capable of deceiving outside observers about their intentions while carrying out tasks in stochastic, complex environments. By modeling the agent's behavior as a Markov decision process, we consider a setting where the agent aims to reach one of multiple potential goals while deceiving outside observers about its true goal. We propose a novel approach to model observer predictions based on the principle of maximum entropy and to efficiently generate deceptive strategies via linear programming. The proposed approach enables the agent to exhibit a variety of tunable deceptive behaviors while ensuring the satisfaction of probabilistic constraints on the behavior. We evaluate the performance of the proposed approach via comparative user studies and present a case study on the streets of Manhattan, New York, using real travel time distributions.

Introduction

Deception is an important capability that is present in many human activities, ranging from sports (Jackson and Cañal-Bruland 2019) to business (Chelliah and Swamy 2018) and military (Tsu 2016). By making deceptive decisions, e.g., by hiding information or conveying false information, teams win games, companies secretly develop new products, and troops gain strategic advantage during battles. Although the outcomes of decisions are typically uncertain, e.g., due to incomplete knowledge and imperfect predictions, humans are still able to deceive one another effectively.

In this paper, we develop a novel approach that enables autonomous systems to make deceptive decisions under uncertainty. Such a deception capability has the potential to improve security in adversarial environments, increase success rates in competitive settings, and create more engaging interactions in games. For example, a delivery drone may protect itself from attacks by deceiving potentially hostile observers about its destination.

We consider an autonomous agent that carries out a task in a complex, stochastic environment. We model the agent's behavior as a Markov decision process (MDP) and express its task as reaching one of multiple potential goal states in the MDP. Being aware of the potential goals, the observer aims to predict the agent's true goal from its trajectories. The agent aims to follow a deceptive strategy that misleads the observer about the true goal either by exaggerating its behavior towards a decoy goal or by creating ambiguity.

The main contribution of this paper is a novel approach that systematically generates globally optimal deceptive strategies in stochastic environments by combining the principle of maximum entropy with stochastic control. The proposed approach involves a number of parameters that enables the agent to exhibit tunable deceptive behaviors and allows the integration of probabilistic resource constraints into the formulation. An overview of the proposed approach is shown in Fig. 1.

We express the observer's predictions on the agent's true goal by developing a prediction model based on the principle of maximum entropy (Ziebart et al. 2008; Ziebart, Bagnell, and Dey 2010). The model is based on three factors, namely, the observer's prior beliefs on the agent's true goal, a cost function expressing the agent's expected goal-directed behavior, and a constant expressing how much efficiency the observer expects from the agent.

We synthesize deceptive strategies for the agent by developing a planning model based on stochastic optimal control (Puterman 2014). The model takes the observer's predictions as an input and constructs a constrained optimization problem that is solved via linear programming. The model is based on three factors, namely, the agent's true goal, a function expressing the type of deception, e.g., exaggeration or ambiguity, and a discount factor controlling the trade-off between the trajectory length and deception.

We present three experiments. Firstly, we illustrate the effects of different parameters in the proposed approach on the agent's deceptive behavior. Secondly, we present online user studies and compare the proposed approach to two recently proposed deception methods (Masters and Sardina 2017; Dragan, Holladay, and Srinivasa 2015) as well as a baseline. Finally, we present a large-scale case study on the streets of Manhattan, New York with real travel time distributions and illustrate the use of deception in realistic scenarios under probabilistic constraints on travel time.

Related Work Deception in autonomous systems has been studied in the literature from different perspectives. In (Masters and Sardina 2017), the authors generate deceptive plans in deterministic environments. For exaggerated

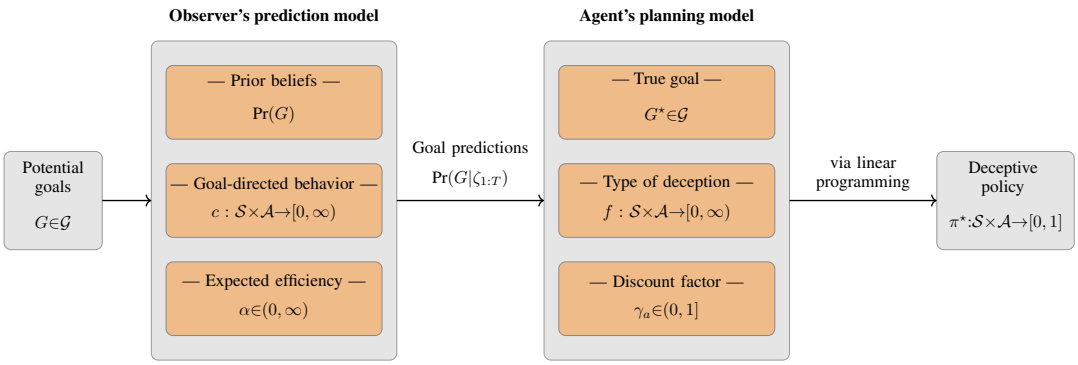

Figure 1: The overview of the proposed deceptive policy synthesis approach. Given a set $\mathcal{G}$ of potential goals, the observer's prediction model assigns a probability $\Pr(G|\zeta_{1:T})$ to each potential goal $G \in \mathcal{G}$ based on the agent's partial trajectory $\zeta_{1:T}$. Utilizing $\Pr(G|\zeta_{1:T})$, the agent synthesizes a deceptive policy via linear programming.

behaviors, their method corresponds to a simple heuristic, i.e., reaching a decoy goal before reaching the true goal. In (Dragan, Holladay, and Srinivasa 2015), a robot with deterministic dynamics is considered and deceptive trajectories are generated using an approach based on functional gradient descent. The work (Kulkarni, Srivastava, and Kambhampati 2019) synthesizes obfuscated plans in deterministic environments by exploiting observation sequences. These approaches are different from the one proposed in this paper as they consider deterministic systems and the synthesized strategies are based on heuristics or local approaches.

In (Ornik and Topcu 2018), the authors synthesize deceptive strategies by expressing the evolution of observer predictions as a stochastic transition system over potential goals, which is constructed using the agent's relative distance to potential goals. Unlike (Ornik and Topcu 2018), we generate observer predictions as probability distributions over potential goals using the principle of maximum entropy. Nature-inspired deception strategies for social robots are developed in (Shim and Arkin 2012; Pettinati and Arkin 2019). Although these approaches are effective, their generality is limited as they lack a mathematical foundation.

Deception has also been studied from the perspective of game theory. In (Wagner and Arkin 2011) and (Nguyen et al. 2019), the authors generate deceptive strategies in single stage games and finitely repeated games, respectively. These strategies are different from the ones synthesized in this work as we focus on stochastic and dynamic settings. The works (Anwar and Kamhoua 2020; Çeker et al. 2016; Kulkarni et al. 2020) study deception for cybersecurity using game-theoretic formulations. The proposed strategies are, in general, restricted to small-scale problems due to the complexity of computing equilibria in dynamic games.

Deception is also related to the problem of goal recognition in which an observer aims to infer an agent's goal based on its past behavior (Ramírez and Geffner 2010; Ramirez and Geffner 2011; Shvo and McIlraith 2020). We consider an observer that aims to infer the agent's goal by using a prediction model based on the principle of maximum entropy. Utilizing this model, we develop a planning algorithm for deceiving the observer regarding the agent's goal.

Background

We model the agent's behavior in a stochastic environment as a Markov decision process (MDP). An MDP is a tuple $\mathcal{M}=(\mathcal{S}, s_1, \mathcal{A}, P)$ where $\mathcal{S}$ is a finite set of states, s_1 is a unique initial state, $\mathcal{A}$ is a finite set of actions, and $P{:}\mathcal{S}\times\mathcal{A}\times\mathcal{S}\to[0,1]$ is a transition function such that $\sum_{s'\in\mathcal{S}} P(s,a,s')=1$ for all $s\in\mathcal{S}$ and $a\in\mathcal{A}$. In an MDP, the agent follows a policy to achieve a task. Formally, a policy $\pi{:}\mathcal{S}\times\mathcal{A}\to[0,1]$ is a mapping such that $\sum_{a\in\mathcal{A}} \pi(s,a)=1$ for all $s\in\mathcal{S}$. We denote the set of all possible policies by Π.

We aim to develop an algorithm such that the agent reaches its goal in an environment while deceiving an outside observer about its goal. Hence, we consider a set of potential goals $\mathcal{G}\subset\mathcal{S}$ in the MDP and denote the agent's true goal by $G^\star\in\mathcal{G}$. For simplicity, we assume that all potential goal states are absorbing, i.e., $P(G,a,G)=1$ for all $G\in\mathcal{G}$.

A trajectory ζ is a sequence $(s_1, a_1, s_2, a_2, s_3, \ldots)$ of states and actions that satisfy $P(s_t, a_t, s_{t+1})>0$ for all $t\in\mathbb{N}$. A *partial* trajectory $\zeta_{1:T}$ of length $T\in\mathbb{N}$ is a sequence $(s_1, a_1, s_2, \ldots, s_T)$. Let $\mathcal{T}_\pi$ denote the set of all admissible trajectories that are generated under the policy π, and $\zeta[t]{:=}s_t$ denote the state visited at the t-th step along ζ. For a given goal state $G\in\mathcal{G}$ and a policy π, we denote by

$$\Pr^\pi(Reach[G]) := \Pr\{\zeta \in \mathcal{T}_\pi : \exists t \in \mathbb{N}, \zeta[t] = G\}$$

the probability with which the agent reaches the goal G under the policy π. Furthermore, we denote by $R_{\max}(G) := \max_{\pi\in\Pi} \Pr^\pi(Reach[G])$ the *maximum* probability of reaching the goal G under any policy. We note that the value of $R_{\max}(G)$ can be efficiently computed via value iteration (Baier and Katoen 2008).

For an MDP $\mathcal{M}$, let $G_\mathcal{M}=(\mathcal{S}, E_\mathcal{M})$ be a directed graph where $\mathcal{S}$ is the set of vertices and $E_\mathcal{M}\subseteq\mathcal{S}\times\mathcal{S}$ is the set of edges such that $(s,s')\in E_\mathcal{M}$ if and only if $\sum_{a\in\mathcal{A}} P_{s,a,s'}>0$. For the graph $G_\mathcal{M}$, we denote by $T_{\min}(s)$ the length of the shortest partial trajectory $\zeta_{1:T_{\min}(s)}$ such that $s_{T_{\min}(s)}=s$. Informally, $T_{\min}(s)$ indicates the minimum number of steps to reach the state s from the initial state s_1. We use the convention $T_{\min}(s)=\infty$ if the state s is not reachable from the initial state. Note that $T_{\min}(s)$ can be efficiently computed, e.g., using Dijkstra's algorithm (Dijkstra et al. 1959).

Modeling Observer Predictions

To deceive an observer about its true goal, the agent should know how the observer associates the agent's partial trajectories with potential goals. In this section, we provide a prediction model that formally expresses the observer's inference method using the principle of maximum entropy. Specifically, we present a prediction model that assigns a probability $\Pr(G|\zeta_{1:T})$ to each potential goal $G{\in}\mathcal{G}$ for a given partial trajectory $\zeta_{1:T}$.

An overview of the observer's prediction model is shown in Fig. 1. We formally characterize the observer with its prior beliefs $\Pr(G)$ on the agent's true goal, the cost function $c{:}\mathcal{S}{\times}\mathcal{A}{\to}[0,\infty)$ that expresses the agent's expected goal-directed behavior from the observer's perspective, and the efficiency parameter $\alpha{\in}(0,\infty)$ that expresses the agent's expected degree of optimality from the observer's perspective.

The prior beliefs constitute a probability distribution over the agent's potential goals and formalize *where* the observer expects the agent to reach when the agent is at the initial state s_1. When the observer interacts with the agent only once, the prior beliefs are typically represented by a uniform distribution. In repeated interactions, Bayesian approaches can be used to construct the prior beliefs from historical data (Ziebart et al. 2009). Note that, given the prior beliefs, we can express the observer's predictions $\Pr(G|\zeta_{1:T})$ as

$$\Pr(G|\zeta_{1:T}) = \frac{\Pr(\zeta_{1:T}|G)\Pr(G)}{\sum_{G'\in\mathcal{G}}\Pr(\zeta_{1:T}|G')\Pr(G')}, \tag{1}$$

where $\Pr(\zeta_{1:T}|G)$ denotes the probability with which the agent follows a partial trajectory $\zeta_{1:T}$ for reaching the goal G. In other words, the probability $\Pr(\zeta_{1:T}|G)$ expresses *how* the observer expects the agent to reach a goal. We formally characterize the probability $\Pr(\zeta_{1:T}|G)$ using the cost function c and the efficiency parameter α. Specifically, to reach a goal $G{\in}\mathcal{G}$, we assume that the observer *expects* the agent to follow a policy $\overline{\pi}_G{\in}\Pi$ that satisfies

$$\overline{\pi}_G \in \arg\min_{\pi\in\Pi} \mathbb{E}^{\pi}\left[\sum_{t=1}^{\infty}\gamma_o^{t-1}\Big(c(s_t,a_t) - \alpha H(\pi(s_t,\cdot))\Big)\right]$$
$$\text{s.t.}\quad \Pr^{\pi}(Reach[G]) = R_{\max}(G).$$

In the above equation, the term $H(\pi(s_t,\cdot))$ measures the entropy of the policy π in the state $s_t{\in}S$ and is defined as $H(\pi(s_t,\cdot)){=}-\sum_{a\in\mathcal{A}}\pi(s_t,a)\log\pi(s_t,a)$. The entropy term quantifies the randomness in the agent's policy and enables the observer to reason about suboptimal trajectories. The parameter $\gamma_o{\in}(0,1)$ is the observer's discount factor, which is introduced only to ensure the finiteness of the solution and can be chosen arbitrarily close to one.

The cost function c expresses the expected goal-directed behavior of a rational agent. For example, in motion planning, the cost function corresponds to the distance between state pairs as observers typically expect the agent to reach its goal through the shortest feasible trajectory (Gergely et al. 1995). We make the standard assumption (Dragan, Lee, and Srinivasa 2013; Sreedharan et al. 2021; Masters and Vered 2021) that the cost function c is known to the agent. In scenarios that involve a cooperative observer, the cost function c can also be learned from demonstrations using existing learning approaches (Ziebart et al. 2008).

The parameter $\alpha{\in}(0,\infty)$ controls how much efficiency the observer expects from the agent. For example, as $\alpha{\to}0$, the agent is expected to be perfectly efficient and follow only the trajectories that minimize its total cost. On the other extreme, as $\alpha{\to}\infty$, the agent is expected to have no efficiency concerns and reach its goal by following random trajectories. We assume that the parameter α is also known to the agent. In practice, one can incorporate the parameter α into the cost function c by defining the costs as $\widetilde{c}(s,a){=}c(s,a)/\alpha$ and learn the function $\widetilde{c}$ from demonstrations.

We can now derive the observer's prediction model from the agent's *expected* policy $\overline{\pi}_G$ as follows. It is known (Haarnoja et al. 2017; Ziebart et al. 2009) that the policy $\overline{\pi}_G$ satisfies $\overline{\pi}_G(s,a){=}e^{(Q_G(s,a)-V_G(s))/\alpha}$ where

$$Q_G(s,a) = -c(s,a) + \gamma_o\sum_{s'\in\mathcal{S}}P(s,a,s')V_G(s')$$
$$V_G(s) = \operatorname*{softmax}_a Q_G(s,a).$$

In the above equations, the softmax operator is defined as $\operatorname{softmax}_x f(x){=}\alpha\log\sum_x e^{f(x)/\alpha}$. The values of $V_G(s)$ and $Q_G(s,a)$ can be iteratively computed via softmax value iteration using the initialization $V_G(G){=}0$ and $V_G(s){=}{-}C$ for all $s{\in}\mathcal{S}\backslash\{G\}$, where C is an arbitrarily large constant.

It is known (Ziebart et al. 2008) that $\Pr(\zeta_{1:T}|G)$ satisfies

$$\Pr(\zeta_{1:T}|G) \approx \frac{e^{-\frac{1}{\alpha}\sum_{t=1}^{T}c(s_t,a_t)+V_G(s_T)}}{e^{V_G(s_1)}}\prod_{t=1}^{T}P(s_t,a_t,s_{t+1})$$

when the transition randomness has a limited effect on the agent's behavior and the discount factor γ_o is large enough. Note that for MDPs with deterministic transitions, the above expression implies that $\Pr(\zeta|G){\propto}e^{-\frac{1}{\alpha}\sum_{t=1}^{\infty}c(s_t,a_t)}$. Hence, in the maximum entropy distribution, the probability of a trajectory exponentially decreases with increasing total cost. Finally, plugging $\Pr(\zeta_{1:T}|G)$ into (1) and simplifying terms, we obtain the observer's prediction model as

$$\Pr(G|\zeta_{1:T}) \approx \frac{e^{V_G(s_T)-V_G(s_1)}\Pr(G)}{\sum_{G'\in\mathcal{G}}e^{V_{G'}(s_T)-V_{G'}(s_1)}\Pr(G')}. \tag{2}$$

Note that the observer's prediction $\Pr(G|\zeta_{1:T})$ is only a function of the agent's initial state s_1 and the current state s_T, i.e., $\Pr(G|\zeta_{1:T}){=}\Pr(G|s_1,s_T)$. Hence, the observer's predictions can be computed *offline* by computing the value of $V_G(s)$ for all $G{\in}\mathcal{G}$ and $s{\in}S$. This computation can be performed by running the softmax value iteration $|\mathcal{G}|$ times.

As the efficiency parameter $\alpha{\to}\infty$, for any given partial trajectory $\zeta_{1:T}$, we have $\Pr(G|\zeta_{1:T}){=}\Pr(G'|\zeta_{1:T})$. This implies that, if the observer expects the agent's goal-directed behavior to be inefficient, then the observer predicts all goals to be equally likely even after the agent's partial trajectory is revealed. In such a scenario, it is impossible to mislead the observer about the true goal. Accordingly, we will see in the experiments that the agent's deceptive behavior corresponds to reaching the true goal via shortest trajectories when the observer expects the agent to be inefficient.

Synthesizing Deceptive Policies

Being aware of the observer's prediction model, the agent aims to synthesize a policy that deceives the observer about its true goal $G^\star$. Formally, we propose to synthesize a deceptive policy $\pi^\star \in \Pi$ under which the agent maximizes the deceptiveness of its trajectory while reaching its true goal with maximum probability, i.e.,

$$\pi^\star \in \arg\min_{\pi\in\Pi} \mathbb{E}^\pi \left[\sum_{t=1}^{\infty} g(s_t, a_t) \right] \tag{3a}$$

$$\text{s.t.} \quad \Pr^\pi(Reach[G^\star]) = R_{\max}(G^\star). \tag{3b}$$

In (3a)-(3b), we express the agent's deception objective through the generic cost function $g{:}\mathcal{S}\times\mathcal{A}\to[0,\infty)$. In particular, we consider a class of functions of the form

$$g(s,a) = \gamma_a^{T_{\min}(s)} f(s,a) \tag{4}$$

where $\gamma_a \in (0,1]$ is a discount factor and $f{:}\mathcal{S}\times\mathcal{A}\to[0,\infty)$ is a mapping that formalizes the type of deception. Recall that the constant $T_{\min}(s)$ is the minimum number of steps to reach the state s from the initial state s_1 in the graph $G_{\mathcal{M}}$. We introduce the term $\gamma_a^{T_{\min}(s)}$ in (4) as a scaling factor to obtain tunable agent behavior. As we will see in the experiments, as γ_a decreases, the cost for states that are further away from the initial state becomes smaller, which encourages the agent to follow longer trajectories for deception.

Mathematical Representation of Deception

We design the mapping f to achieve two common types of deception, namely, exaggeration and ambiguity.

Exaggeration: One of the most common strategies to deceive an observer about the true goal is exaggeration (Dragan, Holladay, and Srinivasa 2015). In this strategy, the agent exhibits an exaggerated behavior by pretending to reach a decoy goal, i.e., a goal that is not the true goal. We express the exaggeration behavior by defining f as

$$f(s,a) = 1 + \Pr(G^\star|s_1,s) - \max_{G\in\mathcal{G}\setminus\{G^\star\}} \Pr(G|s_1,s) \tag{5}$$

if $s\in\mathcal{S}\setminus\mathcal{G}$, and $f(s,a)=0$ otherwise.

In (5), the value of $f(s,a)$ linearly increases with the difference $\Pr(G^\star|s_1,s) - \max_{G\in\mathcal{G}\setminus\{G^\star\}} \Pr(G|s_1,s)$, i.e., the relative likelihood of the true goal with respect to a decoy goal. Hence, the smaller the value of $f(s,a)$, the more likely it is for the agent to reach a decoy goal. Additionally, we have $f(s,a)=0$ if $\Pr(G^\star|s_1,s)=0$ and $\Pr(G|s_1,s)=1$ for some $G\in\mathcal{G}\setminus\{G^\star\}$. That is, the agent incurs no cost in a state if the observer almost surely expects the agent to reach a decoy goal from that state.

Ambiguity: Another possible strategy to deceive an observer about the true goal is to behave ambiguously. In this strategy, the agent exhibits an ambiguous behavior by keeping the likelihood of all potential goals similar along its trajectory. Similar to exaggeration, we express ambiguity by defining the mapping f as

$$f(s,a) = \sum_{G\in\mathcal{G}} \sum_{G'\in\mathcal{G}} \Big| \Pr(G|s_1,s) - \Pr(G'|s_1,s) \Big| \tag{6}$$

if $s\in\mathcal{S}\setminus\mathcal{G}$, and $f(s,a)=0$ otherwise.

In (6), the value of $f(s,a)$ at a state s increases as the relative likelihood of a goal with respect to any other one increases. Hence, the smaller the value of $f(s,a)$, the less likely it is for the agent to try and reach a specific goal. Additionally, we have $f(s,a)=0$ if $\Pr(G|s_1,s)=\Pr(G'|s_1,s)$ for all $G,G'\in\mathcal{G}$, i.e., the agent incurs no cost in a state if the observer expects the agent to reach all goals equally likely.

Synthesis via Linear Programming

We now synthesize deceptive policies by solving a series of linear programs (LPs). For a given MDP $\mathcal{M}$, let $\mathcal{S}_0\subseteq\mathcal{S}$ be a set of states from which there is no trajectory reaching a potential goal $G\in\mathcal{G}$. The set $\mathcal{S}_0$ can be efficiently computed through standard graph search algorithms (Baier and Katoen 2008). Moreover, let $\mathcal{S}_r=\mathcal{S}\setminus(\mathcal{G}\cup\mathcal{S}_0)$.

To obtain the deceptive policy $\pi^\star$, we first solve the following LP:

$$\underset{x(s,a)\geq 0}{\text{minimize}} \sum_{s\in\mathcal{S}_r}\sum_{a\in\mathcal{A}} g(s,a)x(s,a) \tag{7a}$$

subject to:

$$\sum_{a\in\mathcal{A}} x(s,a) - \sum_{s'\in\mathcal{S}}\sum_{a\in\mathcal{A}} P(s',a,s)x(s',a) = \beta_s, \ \forall s\in\mathcal{S}_r \tag{7b}$$

$$\sum_{s\in\mathcal{S}_r}\sum_{a\in\mathcal{A}} x(s,a)r(s,a) = R_{\max}(G^\star). \tag{7c}$$

In the above LP, $x(s,a)$ is a decision variable that corresponds to the agent's expected number of visits to the state-action pair (s,a) (Puterman 2014). The function β_s indicates the initial state distribution, i.e., $\beta_s=1$ if $s=s_1$, and $\beta_s=0$ otherwise. Finally, the function $r{:}\mathcal{S}\times\mathcal{A}\to[0,\infty)$ is the transition probability to the true goal from a given state, i.e., $r(s,a)=P(s,a,G^\star)$ for $s\in\mathcal{S}_r$, and $r(s,a)=0$ otherwise.

The objective function in (7a) corresponds to the agent's expected total cost given in (3a). The constraint in (7b) represents the balance equation (Altman 1999), i.e., the expected number of times the agent enters a state is equal to the expected number of times the agent leaves that state. Finally, the constraint in (7c) ensures that the agent reaches its true goal $G^\star$ with maximum probability $R_{\max}(G^\star)$.

It is possible to extract the deceptive policy $\pi^\star$ from the optimal solution of the LP in (7a)-(7c). However, under the extracted policy, the agent may visit the states with zero cost too many times before reaching its true goal since such states do not affect the objective function. Let $v^\star$ be the optimal value of the LP in (7a)-(7c). To ensure that that the agent reaches its true goal as quickly as possible while achieving its deception objective, we solve the following second LP:

$$\underset{x(s,a)\geq 0}{\text{minimize}} \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} x(s,a) \tag{8a}$$

$$\text{subject to:} \sum_{s\in\mathcal{S}_r}\sum_{a\in\mathcal{A}} g(s,a)x(s,a) = v^\star \tag{8b}$$

$$\text{(7b)} - \text{(7c)}. \tag{8c}$$

Let $\{x^\star(s,a)\geq 0 : s\in\mathcal{S}, a\in\mathcal{A}\}$ be the set of optimal variables for the LP in (8a)-(8c). It follows from (Altman 1999) that the deceptive policy $\pi^\star$ satisfying the condition in (3a)-(3b) can be synthesized by choosing

$$\pi^\star(s,a) = \begin{cases} \frac{x^\star(s,a)}{\sum_{a'\in\mathcal{A}} x^\star(s,a')} & \text{if } \sum_{a'\in\mathcal{A}} x^\star(s,a') > 0, \\ 1/|\mathcal{A}| & \text{otherwise.} \end{cases}$$

Experiments

We now demonstrate the performance of the proposed approach through numerical simulations and user studies. We run all computations on a 3.2 GHz desktop with 8 GB RAM and employ the Gurobi solver (Gurobi Optimization 2021) for optimization. Approvals for user studies are obtained from the University of Texas at Austin IRB (Study #1368).

Generating Tunable Agent Behavior

We first illustrate how to generate a range of deceptive behaviors by tuning α and γ_a. We consider the environment shown in Fig. 2. The initial state is labeled with **S** and the two potential goals are labeled with **G1** and **G2**, with **G1** being the true goal. Black regions indicate the obstacles. The agent has four actions $\{right, left, up, down\}$. Under a given action, the agent transitions to the state in the corresponding direction with probability one.

The agent's expected goal-directed behavior is to follow shortest trajectories to the goal, which we express by setting $c(s,a)=10$ for all $s\in\mathcal{S}$ and $a\in\mathcal{A}$ and $\gamma_o=0.95$. Note that any positive cost expresses the same goal-directed behavior; the value of 10 is chosen to obtain distinct behaviors for a wide range of α values. Recall that, as α gets smaller, the observer expects the agent to be more efficient and follow shorter trajectories to reach its goal. In Fig. 2 (left), the shaded region indicates all the states that a perfectly efficient agent, $\alpha=0$, can potentially visit along its trajectory to the goal **G1**.

In Fig. 2 (left), we generate 5 trajectories to represent the agent's exaggeration behavior for various α and γ_a combinations. As can be seen from the figure, for $\alpha\leq 1$ and $\gamma_a=1$, the agent's exaggerated trajectory reaches the true goal while avoiding the shaded region. This trajectory is deceptive because the observer *expects* the agent to be highly efficient and visit *only* the states in the shaded region while reaching the goal **G1**. As α increases, the observer expects the agent to be less efficient. In that case, to deceive the observer, the agent starts exaggerating its behavior by getting closer to the decoy goal **G2**. As we keep increasing the α value, the observer expects the agent's behavior to be less goal-directed and more random. In that case, it becomes impossible to deceive the observer since any random behavior is expected. Accordingly, for $\alpha\geq 20$, the agent does not try to deceive the observer and follows a shortest trajectory to its goal.

A simple heuristic to achieve exaggeration-type deceptive behavior is to first reach the decoy goal and then the true goal (Masters and Sardina 2017). In the environment shown in Fig. 2 (left), the states that are further away from the initial state have high costs $g(s,a)$ when the discount factor is $\gamma_a=1$. Therefore, the agent has no incentive to follow longer trajectories and pretend to reach the decoy goal. However,

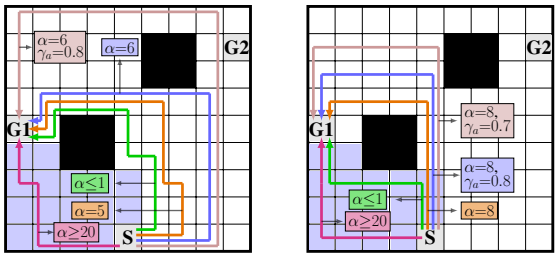

Figure 2: An illustration of deceptive trajectories generated by the proposed approach under various efficiency parameters (α) and discount factors (γ_a). The agent starts from the state **S**. The true goal and decoy goal are **G1** and **G2**, respectively. $\gamma_a=1$ if it is not written explicitly. (Left) Exaggeration behavior. (Right) Ambiguity behavior.

when $\gamma_a=0.8$ and $\alpha=6$, the agent's exaggeration behavior starts exploiting those states as well and replicates the trajectory generated by the aforementioned heuristic approach.

We also generate 5 ambiguous trajectories, shown in Fig. 2 (right). In this environment, ambiguity corresponds to being at the same horizontal distance to both potential goals. Accordingly, to achieve ambiguity for $\alpha\leq 1$, the agent stays at the same horizontal distance to both potential goals for as long as possible while ensuring to visit only the states in the shaded region along its trajectory. As α increases, e.g., $\alpha=8$, the agent is expected to be less efficient, which enables the agent to generate ambiguity for longer. As we keep increasing the value of α, the observer expects the agent to behave randomly. In that case, deception becomes impossible, and the agent reaches its goal by following a shortest trajectory.

The effect of the discount factor γ_a on ambiguity is also illustrated in Fig. 2 (right). As we decrease the value of γ_a, the cost $g(s,a)$ of the states that are further away from the initial state decreases. Consequently, the agent starts exploiting those states to achieve better ambiguity by staying at the same horizontal distance to the potential goals for longer.

User Studies

We conduct two user studies to evaluate the performance of the proposed approach and compare the deceptiveness of the exaggerated trajectories with a baseline and two other algorithms. We consider only exaggerated trajectories since such trajectories are known to be more deceptive than ambiguous trajectories (Dragan, Holladay, and Srinivasa 2015).

We consider the shortest trajectory to the true goal as the baseline (base) algorithm, which we generate by choosing $c(s,a)=1$, $\gamma_o=\gamma_a=0.95$, and $\alpha=20$ in a given environment. For comparison, we generate deceptive trajectories using the algorithms proposed in (Dragan, Holladay, and Srinivasa 2015) and (Masters and Sardina 2017). We note that, unlike the algorithm proposed in this paper, these algorithms are proposed for *deterministic* systems and environments.

In (Dragan, Holladay, and Srinivasa 2015), the authors generate exaggerated (continuous) trajectories for robots. They utilize a functional gradient descent-based (GD) algorithm which *locally* maximizes the cumulative goal probabilities for a decoy goal. By following (Dragan, Holladay,

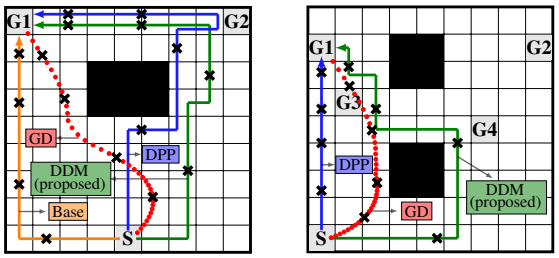

Figure 3: Environments and trajectories in the user studies. Crosses indicate the points up to which a trajectory is shown to the users. In study 2, the baseline is the same with DPP. (Left) User study 1. (Right) User study 2.

and Srinivasa 2015), we initialize the GD algorithm with the baseline trajectory. In (Masters and Sardina 2017), the authors present a deceptive path planning (DPP) algorithm to generate exaggerated trajectories by first reaching a decoy goal. In the case of multiple potential decoys, they choose the decoy goal using a heuristic which corresponds to visiting the decoy goal that is in closest distance to the true goal.

Study 1: the importance of global optimality In the first study, we consider the 9×9 grid world shown in Fig. 3 (left). The agent starts from the state labeled with **S** and has two potential goals **G1** and **G2**, with **G1** being the true goal. Black regions indicate the obstacles. Under each action $a\in\{right, left, up, down\}$, the agent transitions to the state in the corresponding direction with probability one.

Recall that the algorithm proposed in this paper, i.e., deceptive decision-making (DDM), generates *globally* optimal deceptive trajectories via linear programming. In complex environments involving obstacles, as the one considered here, we expect the DDM to be more deceptive than local approaches, e.g., GD. Additionally, since the decoy goal is far away from the true goal, we also expect the "first reach a decoy goal" heuristic (as in DPP) to perform well in this environment. Hence, we hypothesize the following.

$\mathbf{H}_1$: *DDM and DPP generate significantly more deceptive trajectories than GD and baseline.*

We manipulated two independent factors: the *algorithm* (with 4 levels: DDM, DPP, GD, and baseline) and the *segment* at which the trajectory is evaluated (with 4 levels: 25%, 50%, 75%, and 90% of the total length, shown in Fig. 3 (left)), leading to a total of 16 conditions. We used two dependent variables to measure deceptiveness: (i) *goal prediction incorrectness* and (ii) *incorrect prediction confidence*. We used a between-subjects design and recruited 320 users (20 per condition) on Amazon's Mechanical Turk. For each condition, we showed users the corresponding trajectory segment and asked them (i) to predict the agent's goal and (ii) to state their confidence on a 5-point Likert scale.

A factorial ANOVA on *goal prediction incorrectness* (considered to be robust to dichotomous data (D'Agostino 1971)) revealed significant main effects for both *algorithm* ($F(2, 228)$=27.344, p<0.001) and *segment* ($F(3, 228)$=59.817, p<0.001) as well as significant interaction effects ($F(6, 228)$=4.949, p<0.001). A factorial

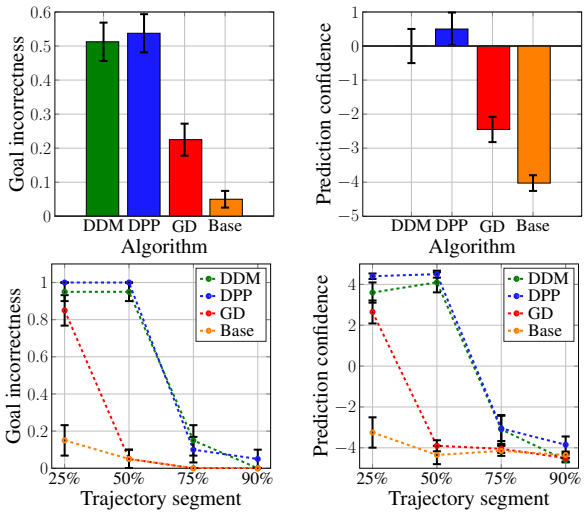

Figure 4: Statistics of the user responses in study 1. (Top) The *algorithm* factor. (Bottom) The *segment* factor.

ANOVA on *incorrect prediction confidence* revealed similar significant main and interaction effects.

In line with $\mathbf{H}_1$, a post-hoc analysis with Tukey HSD that marginalizes over segments showed that DDM and DPP are significantly more deceptive than GD and baseline (p<0.001 for all pairwise comparisons). Fig. 4 echoes these findings, where we plot the means and the standard errors of the dependent variables. Note that DDM induces wrong goal predictions 10 times more often than the baseline and 2 times more often than GD. Moreover, both DDM and DPP induce wrong predictions up to 50% segment of the trajectories, whereas GD reveals the true goal with high confidence after 25% segment of the trajectory.

Study 2: the importance of prediction-awareness Next, we consider the environment shown in Fig. 3 (right) which includes 4 potential goals **G1**,...,**G4**. The true goal is **G1**.

Recall that the DDM algorithm systematically generates exaggerated trajectories using prediction probabilities. In complex environments with multiple decoy goals and obstacles, we expect DDM to be more deceptive than heuristic approaches, e.g., DPP. We also expect the GD algorithm's local optimality to limit its deceptiveness in this complex environment. The trajectory generated by the DPP algorithm first pretends to reach the decoy goal **G3**, which is the closest decoy to the true goal. Hence, DDM coincides with the baseline. In this study, we hypothesize the following.

$\mathbf{H}_2$: *DDM generates significantly more deceptive trajectories than DPP and GD.*

We manipulated two independent factors: the *algorithm* (with 3 levels: DDM, DPP, and GD) and the *segment* at which the trajectory is evaluated (with 4 levels shown in Fig. 3 (right)), leading to a total of 12 conditions. We used a between-subjects design and recruited 240 users (20 per condition) on Amazon's Mechanical Turk. To measure deceptiveness, we used the two dependent variables from the previous study and *two-goal prediction incorrectness*. For

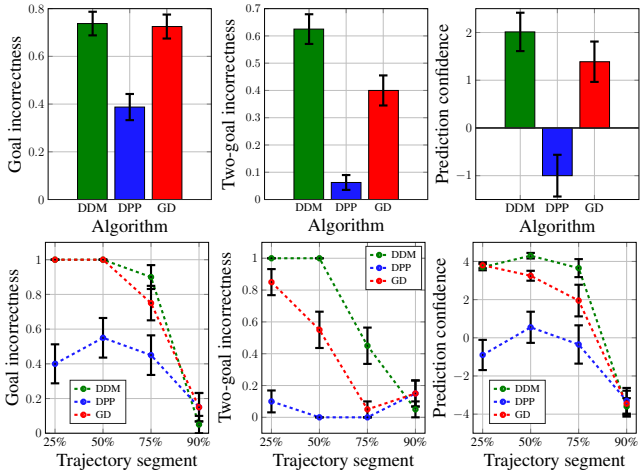

Figure 5: Statistics of the user responses in study 2. (Top) The *algorithm* factor. (Bottom) The *segment* factor.

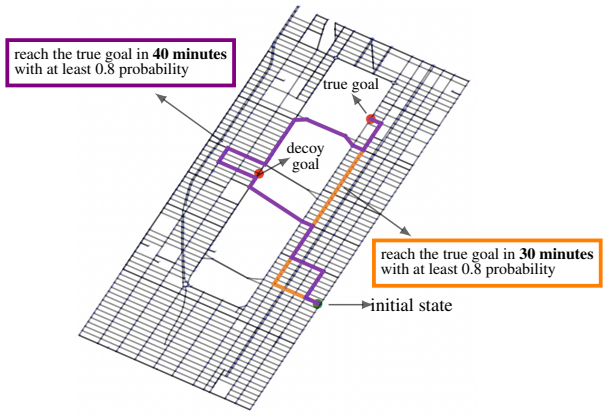

Figure 6: An illustration of deceptive trajectories in Manhattan case study. The agent exaggerates its behavior by moving towards the decoy goal only when the probabilistic constraint on arrival time allows such a behavior.

each condition, we asked users (i) to predict the agent's goal, (ii) to state their confidence on a 5-point Likert scale, and (iii) to predict the agent's second most likely goal. A two-goal prediction is incorrect if the goal prediction *and* the second most likely goal prediction are different than the true goal. Note that we asked the users second most likely goal to understand the effect of the decoy goal **G3** on predictions.

A factorial ANOVA analysis yielded significant main and interaction effects for all dependent variables. In line with $\mathbf{H}_2$, a post-hoc analysis with Tukey HSD that marginalizes over segments revealed that DDM is significantly more deceptive than DPP with respect to all three dependent variables ($p<0.001$ for all comparisons). There is no significant difference between the DDM and GD with respect to *goal prediction incorrectness* and *incorrect prediction confidence* variables. However, the comparison with respect to *two-goal prediction incorrectness* variable revealed that DDM is significantly more deceptive than GD ($p<0.001$). The means and standard errors depicted in Fig. 5 also reflect these findings. Note in the figure that the deceptiveness of the DDM only slightly changes when the users' second most likely goal prediction is included in the analysis, whereas the deceptiveness of GD and DPP dramatically decreases.

Deception Under Probabilistic Constraints

We now consider a large-scale example and demonstrate how the proposed algorithm can generate deceptive trajectories while respecting probabilistic constraints on travel time.

We consider the graph given in Fig. 6, which represents the road network in Manhattan, New York. We utilize the real-world speed data provided in the open source database (Uber Technologies 2021) to express realistic travel times. We generate a continuous travel time distribution on each edge by assuming that the speed follows a lognormal distribution, which is a common assumption in transportation networks (Rakha, El-Shawarby, and Arafeh 2010). To construct the MDP model expressing stochastic travel times, we discretize the travel time distributions and take the Cartesian product of the graph with the set $\{0.5, 1, 1.5, \dots, T_{\max}\}$ of states, where $T_{\max}$ is the maximum travel time in minutes. In this MDP, the agent's transition from a state (s,t) to (s',t') with probability p expresses that the agent's travel from s to s' takes $t'-t$ minutes with probability p.

We consider two potential goals shown in Fig. 6 and synthesize two exaggerated trajectories ensuring that the agent reaches its true goal in $T_{\max}\in\{30, 40\}$ minutes with 0.8 probability. Note that one can encode this constraint in the proposed framework by slightly changing the constraint in (3b) and defining it as $\Pr(Reach[(G^\star, T_{\max})])\geq 0.8$ on the constructed MDP. We choose the value 0.8 to more clearly illustrate the effect of probabilistic time constraints on the agent's deceptive behavior. Finally, we use the parameters $c(s,a)=5$ for all $s\in\mathcal{S}$ and $a\in\mathcal{A}$, $\gamma_o=0.95$, $\alpha=1$, and $\gamma_a=1$.

The two trajectories shown in Fig. 6 demonstrates that the proposed algorithm enables the agent to adjust its exaggeration behavior with respect to the desired travel time. As can be seen from the figure, when the agent is required to arrive its goal in 30 minutes with at least 0.8 probability, it simply follows a shortest trajectory to the goal. This is because the agent's stochastic travel time constraint prevents it from exaggerating its behavior. Indeed, when the agent is required to arrive its goal in 40 minutes instead of 30, it pretends to reach the decoy goal before reaching the true goal.

Conclusions

We consider an autonomous agent that aims to reach one of multiple potential goals in a stochastic environment and propose a novel approach to generate globally optimal deceptive strategies via linear programming. We evaluate the performance of the proposed approach via comparative user studies and present a case study on the streets of Manhattan, New York illustrating the use of deception in realistic scenarios under probabilistic constraints. Future work will focus on characterizing the sensitivity of the deceptive strategies to the knowledge of the observer's prediction model.

Ethical Statement

Although deceptive capabilities can provide benefits both to autonomous systems and the society, their deployment without proper regulations and public education may yield undesirable outcomes. For example, (Arkin 2018) argues, through an ethical discussion, that deceptive behaviors can hinder the public trust towards robots. The author in (Danaher 2020) presents a categorization to systematically discuss deception ethics, and in (Sætra 2021), the author stresses the need for regulation and ethical conduct by producers to avoid the degradation of public trust towards intelligent systems. We believe that theoretical studies, such as this work, are an important first step for understanding the potential drawbacks of deceptive capabilities in autonomy and developing counter-measures against systems with such capabilities. We hope that these studies will also help inform decision-makers and lead to the development of necessary regulatory actions.

Acknowledgements

This work is supported in part by the grants ARL W911NF-17-2-0181, ARL ACC-APG-RTP W911NF1920333, and AFRL FA9550-19-1-0169. We thank Emilie Thome for her contributions to the software implementation.

References

Altman, E. 1999. *Constrained Markov decision processes*. CRC Press.

Anwar, A. H.; and Kamhoua, C. 2020. Game theory on attack graph for cyber deception. In *International Conference on Decision and Game Theory for Security*, 445–456.

Arkin, R. C. 2018. Ethics of robotic deception [opinion]. *IEEE Technology and Society Magazine*, 37(3): 18–19.

Baier, C.; and Katoen, J.-P. 2008. *Principles of Model Checking*. MIT Press.

Çeker, H.; Zhuang, J.; Upadhyaya, S.; La, Q. D.; and Soong, B.-H. 2016. Deception-based game theoretical approach to mitigate DoS attacks. In *International Conference on Decision and Game Theory for Security*, 18–38.

Chelliah, J.; and Swamy, Y. 2018. Deception and lies in business strategy. *Journal of Business Strategy*.

D'Agostino, R. B. 1971. A second look at analysis of variance on dichotomous data. *Journal of Educational Measurement*, 8(4): 327–333.

Danaher, J. 2020. Robot Betrayal: a guide to the ethics of robotic deception. *Ethics and Information Technology*, 22(2): 117–128.

Dijkstra, E. W.; et al. 1959. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1): 269–271.

Dragan, A.; Holladay, R.; and Srinivasa, S. 2015. Deceptive robot motion: synthesis, analysis and experiments. *Autonomous Robots*, 39(3): 331–345.

Dragan, A. D.; Lee, K. C.; and Srinivasa, S. S. 2013. Legibility and predictability of robot motion. In *International Conference on Human-Robot Interaction*, 301–308.

Gergely, G.; Nádasdy, Z.; Csibra, G.; and Bíró, S. 1995. Taking the intentional stance at 12 months of age. *Cognition*, 56(2): 165–193.

Gurobi Optimization, L. 2021. Gurobi Optimizer Reference Manual.

Haarnoja, T.; Tang, H.; Abbeel, P.; and Levine, S. 2017. Reinforcement learning with deep energy-based policies. In *International Conference on Machine Learning*, 1352–1361.

Jackson, R. C.; and Cañal-Bruland, R. 2019. Deception in sport. In *Anticipation and decision making in sport*, 99–116.

Kulkarni, A.; Srivastava, S.; and Kambhampati, S. 2019. A unified framework for planning in adversarial and cooperative environments. In *AAAI Conference on Artificial Intelligence*, 2479–2487.

Kulkarni, A. N.; Luo, H.; Leslie, N. O.; Kamhoua, C. A.; and Fu, J. 2020. Deceptive labeling: hypergames on graphs for stealthy deception. *IEEE Control Systems Letters*, 5(3): 977–982.

Masters, P.; and Sardina, S. 2017. Deceptive Path-Planning. *International Joint Conference on Artificial Intelligence*, 4368–4375.

Masters, P.; and Vered, M. 2021. What's the Context? Implicit and Explicit Assumptions in Model-Based Goal Recognition. In *International Joint Conference on Artificial Intelligence,*, 4516–4523. Survey Track.

Nguyen, T. H.; Wang, Y.; Sinha, A.; and Wellman, M. P. 2019. Deception in finitely repeated security games. In *AAAI Conference on Artificial Intelligence*, 2133–2140.

Ornik, M.; and Topcu, U. 2018. Deception in optimal control. In *Annual Allerton Conference on Communication, Control, and Computing*, 821–828.

Pettinati, M. J.; and Arkin, R. C. 2019. Push and Pull: Shepherding Multi-Agent Robot Teams in Adversarial Situations. In *International Conference on Advanced Robotics and its Social Impacts*, 407–414.

Puterman, M. L. 2014. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

Rakha, H.; El-Shawarby, I.; and Arafeh, M. 2010. Trip travel-time reliability: issues and proposed solutions. *Journal of Intelligent Transportation Systems*, 14(4): 232–250.

Ramírez, M.; and Geffner, H. 2010. Probabilistic plan recognition using off-the-shelf classical planners. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*.

Ramirez, M.; and Geffner, H. 2011. Goal recognition over POMDPs: Inferring the intention of a POMDP agent. In *International joint conference on artificial intelligence*.

Shim, J.; and Arkin, R. C. 2012. Biologically-inspired deceptive behavior for a robot. In *International Conference on Simulation of Adaptive Behavior*, 401–411.

Shvo, M.; and McIlraith, S. A. 2020. Active goal recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 9957–9966.

Sreedharan, S.; Kulkarni, A.; Smith, D.; and Kambhampati, S. 2021. A Unifying Bayesian Formulation of Measures of Interpretability in Human-AI Interaction. In *International*

Joint Conference on Artificial Intelligence,, 4602–4610. Survey Track.

Sætra, H. S. 2021. Social robot deception and the culture of trust. *Paladyn, Journal of Behavioral Robotics*, 12: 276–286.

Tsu, S. 2016. *The art of war*. Cosimo Classics.

Uber Technologies, I. 2021. Uber movement.

Wagner, A. R.; and Arkin, R. C. 2011. Acting deceptively: Providing robots with the capacity for deception. *International Journal of Social Robotics*, 3(1): 5–26.

Ziebart, B. D.; Bagnell, J. A.; and Dey, A. K. 2010. Modeling interaction via the principle of maximum causal entropy. In *International Conference on International Conference on Machine Learning*, 1255–1262.

Ziebart, B. D.; Maas, A. L.; Bagnell, J. A.; Dey, A. K.; et al. 2008. Maximum entropy inverse reinforcement learning. In *AAAI Conference on Artificial Intelligence*, volume 8, 1433–1438.

Ziebart, B. D.; Ratliff, N.; Gallagher, G.; Mertz, C.; Peterson, K.; Bagnell, J. A.; Hebert, M.; Dey, A. K.; and Srinivasa, S. 2009. Planning-based prediction for pedestrians. In *International Conference on Intelligent Robots and Systems*, 3931–3936.