Autonomous Robots (2018) 42:1405–1426
https://doi.org/10.1007/s10514-018-9746-1

Planning for cars that coordinate with people: leveraging effects on human actions for planning and active information gathering over human internal state

Dorsa Sadigh[1] · Nick Landolfi[1] · Shankar S. Sastry[1] · Sanjit A. Seshia[1] · Anca D. Dragan[1]

Received: 16 February 2017 / Accepted: 2 April 2018 / Published online: 4 May 2018

Abstract
Traditionally, autonomous cars treat human-driven vehicles like moving obstacles. They predict their future trajectories and plan to stay out of their way. While physically safe, this results in defensive and opaque behaviors. In reality, an autonomous car's actions will actually affect what other cars will do in response, creating an opportunity for coordination. Our thesis is that we can leverage these responses to plan more efficient and communicative behaviors. We introduce a formulation of interaction with human-driven vehicles as an underactuated dynamical system, in which the robot's actions have consequences on the state of the autonomous car, but also on the human actions and thus the state of the human-driven car. We model these consequences by approximating the human's actions as (noisily) optimal with respect to some utility function. The robot uses the human actions as observations of her underlying utility function parameters. We first explore learning these parameters offline, and show that a robot planning in the resulting underactuated system is more efficient than when treating the person as a moving obstacle. We also show that the robot can target specific desired effects, like getting the person to switch lanes or to proceed first through an intersection. We then explore estimating these parameters online, and enable the robot to perform active information gathering: generating actions that purposefully probe the human in order to clarify their underlying utility parameters, like driving style or attention level. We show that this significantly outperforms passive estimation and improves efficiency. Planning in our model results in *coordination* behaviors: the robot inches forward at an intersection to see if can go through, or it reverses to make the other car proceed first. These behaviors result from the optimization, without relying on hand-coded signaling strategies. Our user studies support the utility of our model when interacting with real users.

Keywords Planning for human–robot interaction · Mathematical models of human behavior · Autonomous driving

1 Introduction

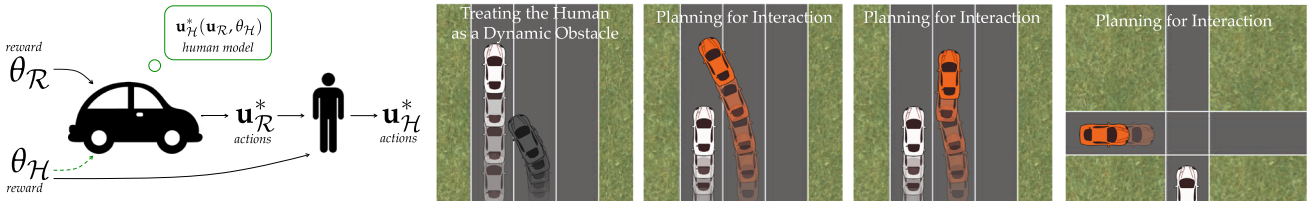
Currently, autonomous cars tend to be overly *defensive* and obliviously *opaque*. When needing to merge into another lane, they will patiently wait for another driver to pass first. When stopped at an intersection and waiting for the driver on the right to go, they will sit there unable to wave them by. They are very capable when it comes to obstacle avoidance, lane keeping, localization, active steering and braking (Urmson et al. 2008; Levinson et al. 2011; Falcone et al. 2007, 2008, 2007; Dissanayake et al. 2001; Leonard et al. 2008). But when it comes to other human drivers, they tend to rely on simplistic models: for example, assuming that other drivers will be bounded disturbances (Gray et al. 2013; Raman et al. 2015), they will keep moving at the same velocity (Vitus and Tomlin 2013; Luders et al. 2010; Sadigh and Kapoor 2015),

This is one of several papers published in *Autonomous Robots* comprising the "Special Issue on Robotics Science and Systems".

This paper combines work from Sadigh et al. (2016b, c). It adds a general formulation of the problem as a game, discusses its limitations, and lays out the assumptions we make to reduce it to a tractable problem. On the experimental side, it adds an analysis of the adaptivity of the behaviors produced to initial conditions for both offline and active estimation, an analysis of the benefits of active estimation on the robot's actual reward, and results on actively estimating user intentions as opposed to just driving style.

✉ Dorsa Sadigh
dorsa@cs.stanford.edu

[1] Department of Computer Science, Stanford University, Stanford, USA



(a) Formulation of Interaction: H computes a best response to R's plan

(b) Defensive R Waits for H before Merging

(c) *attentive H:* *R* merges in front, H brakes

(d) *distracted H:* *R* nudges in, but retracts

(e) *R* Backs Up to Encourage *H* to Cross

Fig. 1 We equip autonomous cars with a model of how humans will react to the car's actions (**a**). We test the planner in user studies, where the car figures out that it can nudge into the human's lane to check their driving style (**b**, **c**): if it gets evidence that they are attentive it merges in front, expecting that the human will slow down; else, it retracts back to its lane (**c**). We also see coordination behavior at an intersection, with the car planning to inch forward to find out more about how the human will act, or even inch backwards to incentivize the human to go through first (**d**)

or they will approximately follow one of a set of known trajectories (Vasudevan et al. 2012; Hermes et al. 2009).

These models predict the trajectory of other drivers as if those drivers act in isolation. They reduce human–robot interaction to obstacle avoidance: the autonomous car's task is to do its best to stay out of the other drivers' way. It will not nudge into the lane to test if the other driver yields, nor creep into the intersection to assert its turn for crossing.

> In reality, the actions of an autonomous car affect the actions of other drivers. Our goal is to leverage these effects in planning to improve efficiency and coordination with human drivers.

We are not the first to propose that robot actions influence human actions. Work in social navigation recognized this, and addressed it by treating the human and the robot as being part of a team—a team that works together to make sure that each agent reaches their goal, and everyone avoids each other (Trautman and Krause 2010; Trautman et al. 2013; Trautman 2013; Kuderer et al. 2015). The robot computes a *coupled plan* by optimizing the team's objective jointly over the human and robot plans, assuming the human will follow their part of the plan, and re-planning at every step. Nikolaidis et al. (2015) recognized that the human might stray away from the plan that the robot computed, and modeled the human as switching to the robot's plan at every time step with some probability.

We propose that when people stray away from the coupled plan, there is a fundamental reason for this: they have *a different objective altogether*. This is particularly true in driving, where coupled planning would assume that the person is just as interested in the robot reaching its goal as they are in themselves reaching theirs—selfish drivers are likely just trying to get home and not get into an accident; they are optimizing for something different.

In this work, we explicitly account for the robot's influence on the person by modeling the person as optimizing their *own* objective or utility function *in response* to the robot's actions. We develop an optimization-based method for planning an autonomous vehicle's behavior in a manner that is cognizant of the effects it will have on human driver actions via this model. This optimization leads to plans like the ones in Fig. 1.

Assuming a universal human model that the robot learns offline, we see that the the orange car in (c) decides to *cut in front* of the human driver in order to more efficiently reach its goal. It arrives at this plan by anticipating that taking this action will cause the human to brake and make space for it. This comes in contrast to treating the person as an obstacle that moves (b) and being more defensive. Since not all drivers are the same, we also arm the robot with the ability to collect information about the human driver online. It then decides to nudge into the person's lane (d) and only merges if it gets evidence that the person is paying attention; otherwise it retreats back to its lane. We find fascinating behaviors at intersections (e): the car inches forward to test human attention, or even inches backwards to get the person to cross first through the intersection. These can be interpreted as signaling behaviors, but they emerge out of optimizing to affect human actions, without ever explicitly modeling human inferences.

We achieve this by planning in an *underactuated dynamical system*: the robot's actions change not only robot state, but also influence human actions and thus human state. We model other drivers as acting approximately optimally according to some reward function that depends on state, human actions, as well as robot actions. We explore learning this human reward function offline, as well as online during the interaction: here, the robot has the opportunity to use its own actions to actively gather more information about the underlying reward. Our contributions are as follows[1]:

[1] A preliminary version of our results was reported in Sadigh et al. (2016a, b). This paper extends that work by providing more detailed discussion and experiments...

1. Formalism of Interaction with Drivers We begin by formalizing the interaction between a human and a robot as a partially observable stochastic game (POSG) in Sect. 2. The human and the robot can both act to change the state of the world, they have partial information because they don't know each other's reward functions, and they arrive at some tuple of policies that are at an equilibrium.

This formulation has two issues: intractability, especially in continuous state and action spaces, and failing to capture human behavior, because humans tend to not follow Nash equilibria in day to day tasks (Hedden and Zhang 2002).

We introduce a simplification of this formulation to an underactuated system. We assume that the robot decides on a trajectory $\mathbf{u}_{\mathcal{R}}$, and the human computes a best response to $\mathbf{u}_{\mathcal{R}}$ (as opposed to trying to influence $\mathbf{u}_{\mathcal{R}}$ as would happen in a game). This reduction enforces turn-taking, and provides us with a dynamics model in which the human observes (or predicts) the robot's actions prior to selecting her actions. It maintains our ability to model the effects of the robot's actions on the human, because the robot can influence which actions the human takes by selecting actions that force a particular response.

2. Approximate Optimization Solution for Known Human Model Assuming a known reward function for the human [which we learn offline through Inverse Reinforcement Learning (Ng et al. 2000; Abbeel and Ng 2005; Ziebart et al. 2008; Levine and Koltun 2012)], we derive an approximate solution for our system based on Model Predictive Control and a quasi-newton optimization in Sect. 3. At every step, the robot replans a trajectory $\mathbf{u}_{\mathcal{R}}$ by reasoning about the optimization that the human would do based on a candidate $\mathbf{u}_{\mathcal{R}}$. We use implicit differentiation to obtain a gradient of the human's trajectory with respect to the robot's. This enables the robot to compute a plan in close to real-time.

3. Extension to Online Inference of Human Reward The solution based on estimating human reward offline from training data is useful in some cases, but ultimately every driver is different, and even the same driver is sometimes more or less aggressive, more or less attentive, and so on. In Sect. 6, we thus also explore estimating the human reward function online.

This turns the problem into a partially observable Markov decision process (POMDP), with the human reward parameters as the hidden state. Prior work that incorporates some notion of human state into planning has thus far separated estimation and planning, always using the current estimate of the human state to determine what the robot should do (Javdani et al. 2015; Fern et al. 2007; Bandyopadhyay et al. 2013). Although efficient, these approximations sacrifice an important aspect of POMDPs: the ability to *actively gather information*.

In recent years, many efforts have developed more efficient algorithms for solving continuous POMDPs (Chaudhari et al. 2013; Seiler et al. 2015; Agha-Mohammadi et al. 2014); however, computing the transition function for our POMDP is costly as it involves solving for human's best response. This renders the aforementioned approaches ineffective as they usually require either a succinct transition function or an easily computable one.

Our work takes advantage of the underactuated system to gather information about the human reward parameters. Rather than relying on passive observations, the robot actually accounts for the fact that humans will react to their actions: it uses this knowledge to select actions that will trigger human reactions which in turn will clarify the internal state. This is in line with work in active information gathering in manipulation, safe exploration, or patrolling (Javdani et al. 2013; Atanasov et al. 2014; Atanasov 2015), now used over human state as opposed to physical state.

4. Analysis of Planning in Human–Robot Driving Scenarios We present the consequences of planning in this dynamical system in Sect. 4, showcasing behaviors that emerge when rewarding the robot for certain effects on human state, like making the human slow down, change lanes, or go first through an intersection. We also show that such behaviors can emerge from simply rewarding the robot for reaching its goal state fast—the robot becomes more aggressive by leveraging its possible effects on human actions. This does not happen always: in Sect. 7, we show the robot maintains a belief over the human driver model, and starts nudging into their lane to figure out if they are going to let the robot merge, or inching forward at a 4-way stop.

We test the planner in two user studies in a driving simulator: one with a human reward learned offline in Sect. 5, and one with online estimation in Sect. 8. Our results suggest that despite the approximation our algorithm makes, it significantly outperforms the "people as moving obstacles" interaction paradigm. We find that the robot achieves significantly higher reward when planning in the underactuated system, and that active information gathering does outperform passive estimation with real users.

Overall, this paper takes a step towards robots that account for their effects on human actions in situations that are not entirely cooperative, and leverage these effects to coordinate with people. Natural coordination and interaction strategies are not hand-coded, but emerge out of planning in our model.

2 General formalism for human–robot interaction as a game

To enable a robot to autonomously generate behavior for interaction and coordination with a human, we need to set up

a model that goes beyond a single agent acting in the physical world among moving obstacles, and consider two-agent models. One aspect that sets interaction with people apart from typical multi-agent models like those used for multi-robot planning is that we do not know or have direct control over what the human is trying to do and how. Furthermore, the human also does not know what the robot is trying to do nor how the robot is going to do it.

Even the simplest formulation of the problem becomes a two player game, and in what follows we introduce such a formulation and expose some of its issues, including the computational complexity of planning in such a model as well as the model's inaccuracy in capturing human behavior. We then propose an approximation that simplifies planning in the game to planning in an *underactuated* system: the robot had direct control over its actions, but also has a model for how its actions will influence those of the human.

Much of robotics research focuses on how to enable a robot to achieve physical tasks, often times in the face of perception and movement error—of partially observable worlds and nondeterministic dynamics (Prentice and Roy 2009; Javdani et al. 2013; Patil et al. 2015). Part of what makes human–robot interaction difficult is that even if we assume the physical world to be fully observable and deterministic, we are still left with a problem as complex as an *incomplete information repeated two player game*. It is a game because it involves multiple rational (or rather, approximately rational) agents who can take actions to maximize their own (expected) utilities. It is repeated because unlike single shot games, the agents act over a time horizon. It is incomplete information because the agents do not necessarily know each others' reward functions (Aumann et al. 1995).

Partially Observable Stochastic Game Extrapolating from the (PO)MDP formulation of a robot acting in isolation to human–robot interaction, we can formulate interaction as a special case of a Partially-Observable Stochastic Game (POSG) (Hansen et al. 2004): there are two "players", the robot $\mathcal{R}$ and the human $\mathcal{H}$; at every step t, they can apply control inputs $u^t_{\mathcal{R}} \in \mathcal{U}_{\mathcal{R}}$ and $u^t_{\mathcal{H}} \in \mathcal{U}_{\mathcal{H}}$; they each have a reward function, $r_{\mathcal{R}}$ and $r_{\mathcal{H}}$; and there is a state space $\mathcal{S}$ with states s consisting of both the physical state x, as well as *reward parameters* θ_R and θ_H.

Including the reward parameters in the state is an unusual trick, necessary here in order to ensure that the human and the robot do not have direct access to each others' reward functions, while maintaining a well defined POSG: $\mathcal{R}$ does not observe $\theta_{\mathcal{H}}$, and $\mathcal{H}$ does not observe $\theta_{\mathcal{R}}$, but both agents can technically evaluate each reward at any state-control tuple $(s^t, u^t_{\mathcal{R}}, u^t_{\mathcal{H}})$ just because s contains the needed reward parameter information: $s = (x, \theta_{\mathcal{R}}, \theta_{\mathcal{H}})$—if an agent knew the state, it could evaluate the other agent's reward.

To simplify the physical world component and focus on the interaction component, we assume fully observable physical states with deterministic dynamics. The observations in the game are thus the physical state x and the controls that each agent applies at each time step. The dynamics model is deterministic, affecting x through the control inputs and leaving the reward parameters unchanged (a reasonable assumption for relatively short interactions).

Aside 1 The POSG above is not necessarily the most direct extension of single-agent planning in deterministic fully observable state spaces to interactions. Collaborative interactions have been modeled simply as a centralized multi-agent system with a *shared* reward function (Kuderer et al. 2015) (i.e. $r_{\mathcal{R}} = r_{\mathcal{H}}$), but this reduces to treating the human as another robot that the system has full control over—essentially the system has more degrees of freedom to control, and there is no difference between the human DoFs and the robot DoFs. Unlike the POSG, it assumes that the human and the robot know each other's reward, and even more, that their rewards are identical. This can be true of specific collaborative tasks, but not of interactions in general: robots do not know exactly what humans want, humans do not know exactly what robots have been programmed to optimize for, and their rewards might have common terms but will not be identical. This happens when an autonomous car interacts with other drivers or with pedestrians, and it even happens in seemingly collaborative scenarios like rehabilitation, in which very long horizon rewards might be aligned but not short-term interaction ones. The POSG formulation captures this intricacy of general interaction.

Aside 2 The POSG above is the simplest extension of the single-agent models to interactions between a human and a robot that do not share a reward function or know each others' rewards. More complex models might include richer state information (such as human's beliefs about the robot, trust, estimation of capability, mood, etc.), and might allow it to change over time.

Limitations of the Game Formulation The POSG formulation is a natural way to characterize interaction from the perspective of MDP-like models, but is limited in two fundamental ways: (1) its computational complexity is prohibitive even in discrete state and action spaces (Bernstein et al. 2002; Hansen et al. 2004) and no methods are known to handle continuous spaces, and (2) it is not a good model for how people actually work—people do not solve games in everyday tasks when they are not playing chess (Hedden and Zhang 2002). Furthermore, solutions here are tuples of policies that are in a Nash equilibrium, and it is not clear what equilibrium to select.

3 Approximate solution as an underactuated system

To alleviate the limitations from above, we introduce an approximate close to real-time solution, with a model of human behavior that does not assume that people compute equilibria of the game.

3.1 Assumptions that simplify the POSG

Our approximation makes several simplifying assumptions that turn the game into an offline learning phase in which the robot learns the human's reward function, followed by an online planning phase in which the robot is solving an underactuated control problem:

Separation of Estimation & Control We separate the process of computing actions for the robot into two stages. First, the robot estimates the human reward function parameters $\theta_{\mathcal{H}}$ offline. Second, the robot exploits this estimate as a fixed approximation to the human's true reward parameters during planning. In the offline phase, we estimate $\theta_{\mathcal{H}}$ from user data via Inverse Reinforcement Learning (Ng et al. 2000; Abbeel and Ng 2005; Ziebart et al. 2008; Levine and Koltun 2012). This method relies heavily on the approximation of all humans to a constant set of reward parameters, but we will relax this separation of estimation and control in Sect. 6.

Model Predictive Control (MPC) Solving the POSG requires planning to the end of the full-time horizon. We reduce the computation required by planning for a shorter horizon of N time steps. We execute the control only for the first time step, and then re-plan for the next N at the next time step (Camacho and Alba 2013).

Let $\mathbf{x} = (x^1, \ldots, x^N)^\top$ denote a sequence of states over a finite horizon, N, and let $\mathbf{u}_{\mathcal{H}} = (u^1_{\mathcal{H}}, \ldots, u^N_{\mathcal{H}})^\top$ and $\mathbf{u}_{\mathcal{R}} = (u^1_{\mathcal{R}}, \ldots, u^N_{\mathcal{R}})^\top$ denote a finite sequence of continuous control inputs for the human and robot, respectively. We define $R_{\mathcal{R}}$ as the robot's reward over the finite MPC time horizon:

$$R_{\mathcal{R}}(x^0, \mathbf{u}_{\mathcal{R}}, \mathbf{u}_{\mathcal{H}}) = \sum_{t=1}^{N} r_{\mathcal{R}}(x^t, u^t_{\mathcal{R}}, u^t_{\mathcal{H}}), \tag{1}$$

where x^0 denotes the present physical state at the current iteration, and each state thereafter is obtained from the previous state and the controls of the human and robot using a given dynamics model, f.

At each iteration, we desire to find the sequence $\mathbf{u}_{\mathcal{R}}$ which maximizes the reward of the robot, but this reward ostensibly depends on the actions of the human. The robot might attempt to influence the human's actions, and the human, optimizing its own reward functions, might likewise attempt to influence the actions of the robot. Despite our reduction to a finite time horizon, the game formulation still demands computing equilibria to the problem. Our core assumption, which we discuss next, is that this is not required for most interactions: that a simpler model of what the human does suffices.

Simplification of the Human Model To avoid computing these equilibria, we propose to model the human as responding rationally to some fixed extrapolation of the robot's actions. At every time step, t, $\mathcal{H}$ computes a simple estimate of $\mathcal{R}$'s plan for the remaining horizon, $\tilde{\mathbf{u}}^{t+1:N}_{\mathcal{R}}$, based on the robot's previous actions $\mathbf{u}^{\mathbf{0:t}}_{\mathcal{R}}$. Then the human computes its plan $\mathbf{u}_{\mathcal{H}}$ as a *best response* (Fudenberg and Tirole 1991) to this estimate. We remark that with this assumption the human is not modeled as a passive agent since the reward function can model the behavior of any active agent, if allowed to have arbitrary complexity. With this simplification, we reduce the general game to a Stackelberg competition: the human computes its best outcome while holding the robots plan fixed. This simplification is justified in practice because usually the agents in driving scenarios are competing with each other rather than collaborating. Solutions to Stackelberg games are in general more conservative in competitive regimes since an information advantage is given to the second player. As a result, the quality of a good solution found in a Stackelberg game model does not decrease in practice. We additionally use a short time horizon, so not much is lost by this approximation. But fundamentally, we do assume that people would not try to influence the robot's actions, and this is a limitation when compared to solving the POSG.

Let $R_{\mathcal{H}}$ be the human reward over the time horizon:

$$R_{\mathcal{H}}(x^0, \mathbf{u}_{\mathcal{R}}, \mathbf{u}_{\mathcal{H}}) = \sum_{t=1}^{N} r_{\mathcal{H}}(x^t, u^t_{\mathcal{R}}, u^t_{\mathcal{H}}), \tag{2}$$

then we can compute the control inputs of the human from the remainder of the horizon by:

$$u^t_{\mathcal{H}}(x^0, \mathbf{u}^{0:t}_{\mathcal{R}}, \tilde{\mathbf{u}}^{t+1:N}_{\mathcal{R}}) = \arg\max_{\mathbf{u}^{t+1:T}_{\mathcal{H}}} R_{\mathcal{H}}(x^t, \tilde{\mathbf{u}}^{t+1:N}_{\mathcal{R}}, u^{t+1:N}_{\mathcal{H}}). \tag{3}$$

This human model would certainly not work well in adversarial scenarios, but our hypothesis, supported by our results, is that it is useful enough in day-to-day tasks to enable robots to be more effective and more fluent interaction partners.

In our work, we propose to make the human's estimate $\tilde{\mathbf{u}}_{\mathcal{R}}$ equal to the actual robot control sequence $\mathbf{u}_{\mathcal{R}}$. Our assumption that the time horizon is short enough that the human can effectively extrapolate the robot's course of action motivates this decision. With this presumption, the human's plan becomes a function of the initial state and robot's true plan:

$$\mathbf{u}^*_{\mathcal{H}}(x^0, \mathbf{u}_{\mathcal{R}}) = \arg\max_{\mathbf{u}_{\mathcal{H}}} R_{\mathcal{H}}(x^t, \mathbf{u}_{\mathcal{R}}, \mathbf{u}_{\mathcal{H}}). \tag{4}$$

This is now an underactuated system the robot has direct control over (can actuate) $\mathbf{u}_{\mathcal{R}}$ and indirect control over (cannot actuate but does affect) $\mathbf{u}_{\mathcal{H}}$. However, the dynamics model in our setup is more sophisticated than in typical underactuated systems because it models the response of the humans to the robot's actions. Evaluating the dynamics model requires solving for the optimal human response, $\mathbf{u}^*_{\mathcal{H}}$.

The system is also a special case of an MDP, with the state as in the POSG, the actions being the actions of the robot, and the world dynamics being dictated by the human's response and the resulting change on the world from both human and robot actions.

The robot can now plan in this system to determine which $\mathbf{u}_{\mathcal{R}}$ would lead to the best outcome for itself:

$$\mathbf{u}^*_{\mathcal{R}} = \arg\max_{\mathbf{u}_{\mathcal{R}}} R_{\mathcal{R}}\left(x^0, \mathbf{u}_{\mathcal{R}}, \mathbf{u}^*_{\mathcal{H}}(x^0, \mathbf{u}_{\mathcal{R}})\right). \tag{5}$$

3.2 Planning with Quasi-Newton optimization

Despite the reduction to a single agent complete information underactuated system, the dynamics remain too complex to solve in real-time. We lack an analytical form for $\mathbf{u}^*_{\mathcal{H}}(x^0, \mathbf{u}_{\mathcal{R}})$ which forces us to solve (4) each time we evaluate the dynamics.

Assuming a known human reward function $r_{\mathcal{H}}$ [which we will obtain later through Inverse Reinforcement Learning (IRL), see Ng et al. (2000), Abbeel and Ng (2005), Ziebart et al. (2008) and Levine and Koltun (2012)], we can solve (5) locally, using gradient-based methods. Our main contribution is agnostic to the particular optimization method, but we use L-BFGS (Andrew and Gao 2007), a quasi-Newton method that stores an approximate inverse Hessian implicitly resulting in fast convergence.

To perform the local optimization, we need the gradient of (2) with respect to $\mathbf{u}_{\mathcal{R}}$:

$$\frac{\partial R_{\mathcal{R}}}{\partial \mathbf{u}_{\mathcal{R}}} = \frac{\partial R_{\mathcal{R}}}{\partial \mathbf{u}_{\mathcal{H}}} \frac{\partial \mathbf{u}^*_{\mathcal{H}}}{\partial \mathbf{u}_{\mathcal{R}}} + \frac{\partial R_{\mathcal{R}}}{\partial \mathbf{u}_{\mathcal{R}}} \tag{6}$$

We can compute both $\frac{\partial R_{\mathcal{R}}}{\partial \mathbf{u}_{\mathcal{H}}}$ and $\frac{\partial R_{\mathcal{R}}}{\partial \mathbf{u}_{\mathcal{R}}}$ symbolically through back-propagation because we have a representation of $R_{\mathcal{R}}$ in terms of $\mathbf{u}_{\mathcal{H}}$ and $\mathbf{u}_{\mathcal{R}}$.

What remains, $\frac{\partial \mathbf{u}^*_{\mathcal{H}}}{\partial \mathbf{u}_{\mathcal{R}}}$, is difficult to compute because $\mathbf{u}^*_{\mathcal{H}}$ is technically the outcome of a global optimization. To compute $\frac{\partial \mathbf{u}^*_{\mathcal{H}}}{\partial \mathbf{u}_{\mathcal{R}}}$, we use the method of implicit differentiation. Since $R_{\mathcal{H}}$ is a smooth function whose minimum can be attained, we conclude that for the unconstrained optimization in (4), the gradient of $R_{\mathcal{H}}$ with respect to $\mathbf{u}_{\mathcal{H}}$ evaluates to 0 at its optimum $\mathbf{u}^*_{\mathcal{H}}$:

$$\frac{\partial R_{\mathcal{H}}}{\partial \mathbf{u}_{\mathcal{H}}}\left(x^0, \mathbf{u}_{\mathcal{R}}, \mathbf{u}^*_{\mathcal{H}}(x^0, \mathbf{u}_{\mathcal{R}})\right) = 0 \tag{7}$$

Now, we differentiate the expression in (7) with respect to $\mathbf{u}_{\mathcal{R}}$:

$$\frac{\partial^2 R_{\mathcal{H}}}{\partial \mathbf{u}^2_{\mathcal{H}}} \frac{\partial \mathbf{u}^*_{\mathcal{H}}}{\partial \mathbf{u}_{\mathcal{R}}} + \frac{\partial^2 R_{\mathcal{H}}}{\partial \mathbf{u}_{\mathcal{H}} \partial \mathbf{u}_{\mathcal{R}}} \frac{\partial \mathbf{u}_{\mathcal{R}}}{\partial \mathbf{u}_{\mathcal{R}}} = 0 \tag{8}$$

Finally, we solve for a symbolic expression of $\frac{\partial \mathbf{u}^*_{\mathcal{H}}}{\partial \mathbf{u}_{\mathcal{R}}}$:

$$\frac{\partial \mathbf{u}^*_{\mathcal{H}}}{\partial \mathbf{u}_{\mathcal{R}}} = \left[-\frac{\partial^2 R_{\mathcal{H}}}{\partial \mathbf{u}_{\mathcal{H}} \partial \mathbf{u}_{\mathcal{R}}}\right]\left[\frac{\partial^2 R_{\mathcal{H}}}{\partial \mathbf{u}^2_{\mathcal{H}}}\right]^{-1} \tag{9}$$

and insert it into (6), providing an expression for the gradient $\frac{\partial R_{\mathcal{R}}}{\partial \mathbf{u}_{\mathcal{R}}}$.

3.3 Offline estimation of human reward parameters

Thus far, we have assumed access to $r_{\mathcal{H}}(x^t, u^t_{\mathcal{R}}, u^t_{\mathcal{H}})$. In our implementation, we learn this reward function from human data. We collect demonstrations of a driver in a simulation environment, and use Inverse Reinforcement Learning (Ng et al. 2000; Abbeel and Ng 2005; Ziebart et al. 2008; Levine and Koltun 2012; Shimosaka et al. 2014; Kuderer et al. 2015) to recover a reward function that explains the demonstrations.

To handle continuous state and actions space, and cope with noisy demonstrations that are perhaps only locally optimal, we use continuous inverse optimal control with locally optimal examples (Levine and Koltun 2012).

We parametrize the human reward function as a linear combination of features:

$$r_{\mathcal{H}}(x^t, u^t_{\mathcal{R}}, u^t_{\mathcal{H}}) = \theta^\top_{\mathcal{H}} \phi(x^t, u^t_{\mathcal{R}}, u^t_{\mathcal{H}}), \tag{10}$$

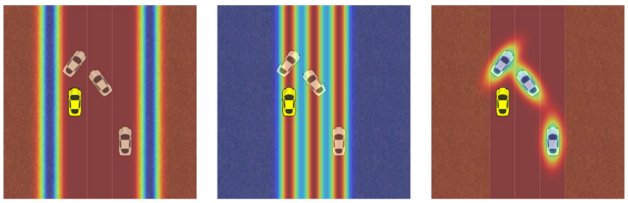
and apply the principle of maximum entropy (Ziebart et al. 2008; Ziebart 2010) to define a probability distribution over human demonstrations $\mathbf{u}_{\mathcal{H}}$, with trajectories that have higher reward being more probable:

$$P(\mathbf{u}_{\mathcal{H}} \mid x^0, \theta_{\mathcal{H}}) = \frac{\exp\left(R_{\mathcal{H}}(x^0, \mathbf{u}_{\mathcal{R}}, \mathbf{u}_{\mathcal{H}})\right)}{\int \exp\left(R_{\mathcal{H}}(x^0, \mathbf{u}_{\mathcal{R}}, \tilde{\mathbf{u}}_{\mathcal{H}})\right) d\tilde{\mathbf{u}}_{\mathcal{H}}}. \tag{11}$$

We then optimize the weights $\theta_{\mathcal{H}}$ in the reward function that make the human demonstrations the most likely:

$$\max_{\theta_{\mathcal{H}}} P\left(\mathbf{u}_{\mathcal{H}} \mid x^0, \theta_{\mathcal{H}}\right) \tag{12}$$

We approximate the partition function in (11) following (Levine and Koltun 2012), by computing a second order Taylor approximation around the demonstration:



(a) Features for the road boundaries **(b)** Features for staying inside the lanes **(c)** Features for avoiding other vehicles

Fig. 2 Features used in IRL for the human driven vehicle; warmer colors correspond to higher reward. We illustrate features corresponding to (**a**) respecting road boundaries, **b** holding lanes, and **c** avoiding collisions with other cars

$$R_{\mathcal{H}}(x^0, \mathbf{u}_{\mathcal{R}}, \tilde{\mathbf{u}}_{\mathcal{H}}) \simeq R_{\mathcal{H}}(x^0, \mathbf{u}_{\mathcal{R}}, \mathbf{u}_{\mathcal{H}}) + (\tilde{\mathbf{u}}_{\mathcal{H}} - \mathbf{u}_{\mathcal{H}})^\top \frac{\partial R_{\mathcal{H}}}{\partial \mathbf{u}_{\mathcal{H}}} + (\tilde{\mathbf{u}}_{\mathcal{H}} - \mathbf{u}_{\mathcal{H}})^\top \frac{\partial^2 R_{\mathcal{H}}}{\partial \mathbf{u}_{\mathcal{H}}^2} (\tilde{\mathbf{u}}_{\mathcal{H}} - \mathbf{u}_{\mathcal{H}}), \tag{13}$$

which results in the the integral in (11) reducing to a Gaussian integral, with a closed form solution. See Levine and Koltun (2012) for more details.

We display a heat map of features we used in Fig. 2. The warmer colors correspond to higher rewards. In addition to the features shown in the figure, we include a quadratic function of the speed to capture efficiency in the objective. The five features include:

- $\phi_1 \propto c_1 \cdot \exp(-c_2 \cdot d^2)$: distance to the boundaries of the road, where d is the distance between the vehicle and the road boundaries and c_1 and c_2 are appropriate scaling factors as shown in Fig. 2a.
- ϕ_2: distance to the middle of the lane, where the function is specified similar to ϕ_1 as shown in Fig. 2b.
- $\phi_3 = (v - v_{\max})^2$: higher speed for moving forward through, where v is the velocity of the vehicle, and $v_{\max}$ is the speed limit.
- $\phi_4 = \beta_{\mathcal{H}} \cdot \mathbf{n}$: heading; we would like the vehicle to have a heading along with the road using a feature, where $\beta_{\mathcal{H}}$ is the heading of $\mathcal{H}$, and $\mathbf{n}$ is a normal vector along the road.
- ϕ_5 corresponds to collision avoidance, and is a non-spherical Gaussian over the distance of $\mathcal{H}$ and $\mathcal{R}$, whose major axis is along the robot's heading as shown in Fig. 2c.

We collected demonstrations of a single human driving for approximately an hour in an environment with multiple autonomous cars, which followed precomputed routes. Despite the simplicity of our features and robot actions during the demonstrations, the learned human model proved sufficient for the planner to produce human-interpretable behavior (case studies in Sect. 4), and actions which affected human action in the desired way (user study in Sect. 5).

3.4 Implementation details

In our implementation, we used the software package Theano (Bergstra et al. 2010; Bastien et al. 2012) to symbolically compute all Jacobians and Hessians. Theano optimizes the computation graph into efficient C code, which is crucial for real-time applications.

This implementation enables us to solve each step of the optimization in equation (5) in approximately 0.3 s for horizon length $N = 5$ on a 2.3 GHz Intel Core i7 processor with 16 GB RAM. Future work will focus on achieving better computation time and a longer planning horizon.

4 Case studies with offline estimation

We noted earlier that the state of the art autonomous driving plans conservatively because of its simple assumptions regarding the environment and vehicles on the road. In our experiments, we demonstrate that an autonomous vehicle can purposefully affect human drivers, and can use this ability to gather information about the human's driving style and goals.

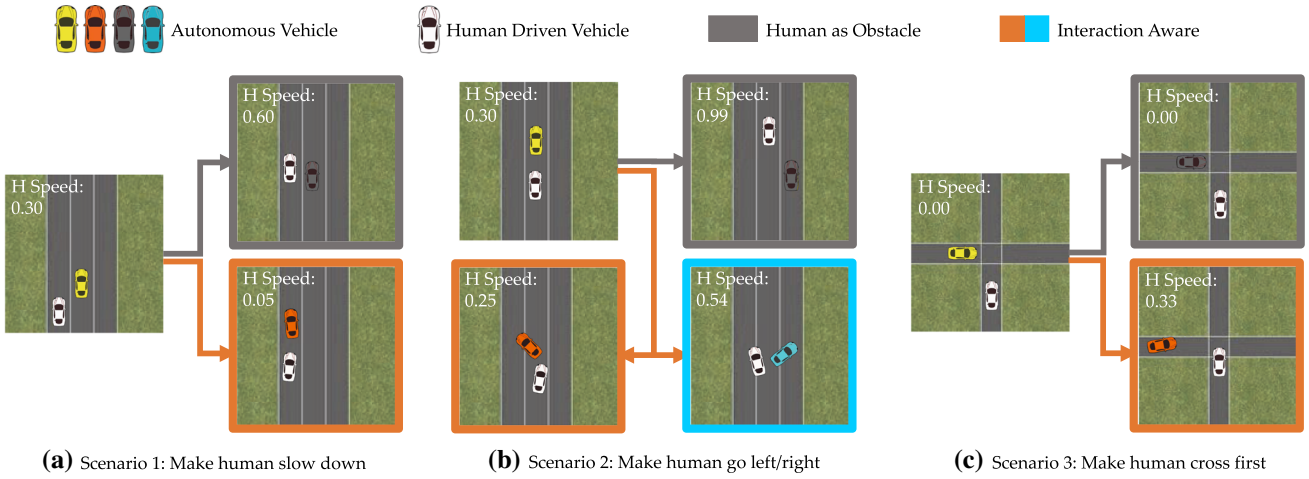
In this section, we introduce 3 driving scenarios, and show the result of our planner assuming a simulated human driver, highlighting the behavior that emerges from different robot reward functions. In the next section, we test the planner with real users and measure the effects of the robot's plan. Figure 3 illustrates our three scenarios, and contains images from the actual user study data.

4.1 Conditions for analysis across scenarios

In all three scenarios, we start from an initial position of the vehicles on the road, as shown in Fig. 3. In the ***control*** condition, we give the car the reward function to avoid collisions and have high velocity. We refer to this as R_{control}. In the ***experimental*** condition, we augment this reward function with a term corresponding to a desired human action (e.g. low speed, lateral position, etc.). We refer to this as $R_{\text{control}} + R_{\text{affect}}$. Sections 4.3 through 4.5 contrast the two plans for each of our three scenarios. Section 4.6 shows what happens when instead of explicitly giving the robot a reward function designed to trigger certain effects on the human, we simply task the robot with reaching a destination as quickly as possible.

4.2 Driving simulator

We use a simple point-mass model of the car's dynamics. We define the physical state of the system $\mathbf{x} = [x \ y \ \psi \ v]^\top$, where x, y are the coordinates of the vehicle, ψ is the heading, and v is the speed. We let $\mathbf{u} = [u_1 \ u_2]^\top$ represent the control input, where u_1 is the steering input and u_2 is the acceleration.



(a) Scenario 1: Make human slow down **(b)** Scenario 2: Make human go left/right **(c)** Scenario 3: Make human cross first

Fig. 3 Driving scenarios. In (**a**), the car plans to merge in front of the human in order to make them slow down. In (**b**), the car plans to direct the human to another lane, and uses its heading to choose which lane the human will go to. In (**c**), the car plans to back up slightly in order to make the human proceed first at the intersection. None of these plans use any hand coded strategies. They *emerge* out of optimizing with a learned model of how humans react to robot actions. In the training data for this model, the learned was *never exposed to situations* where another car stopped at an orientation as in (**b**), or backed up as in (**c**). However, by capturing human behavior in the form of a reward, the model is able to generalize to these situations, enabling the planner to find creative ways of achieving the desired effects

We denote the friction coefficient by μ. We can write the dynamics model:

$$[\dot{x}\ \ \dot{y}\ \ \dot{\psi}\ \ \dot{v}] = [v \cdot \cos(\psi) \quad v \cdot \sin(\psi) \quad v \cdot u_1 \quad u_2 - \mu \cdot v] \quad (14)$$

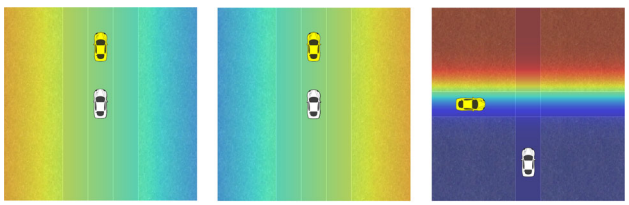
4.3 Scenario 1: Make human slow down

In this highway driving setting, we demonstrate that an autonomous vehicle can plan to cause a human driver to slow down. The vehicles start at the initial conditions depicted on left in Fig. 3a, in separate lanes. In the experimental condition, we augment the robot's reward with the negative of the square of the human velocity, which encourages the robot to slow the human down.

Figure 3a contrasts our two conditions. In the control condition, the human moves forward uninterrupted. *In the experimental condition, however, the robot plans to move in front of the person, anticipating that this will cause the human to brake.*

4.4 Scenario 2: Make human go left/right

In this scenario, we demonstrate that an autonomous vehicle can plan to affect the human's lateral location, making the human switch lanes. The vehicles start at the initial conditions depicted on left in Fig. 3b, in the same lane, with the robot ahead of the human. In the experimental condition, we augment the robot's reward with the lateral position of the human, in two ways, to encourage the robot to make the human go either left (orange border image) or right (blue border image). The two reward additions are shown in Fig. 4a, b.



(a) Scenario 2 Reward: force human left **(b)** Scenario 2 Reward: force human right **(c)** Scenario 3 Reward: force human to cross first

Fig. 4 Heat map of the reward functions in scenarios 2 and 3. The warmer colors show higher reward values. In (**a**, **b**), the reward function of the autonomous vehicle is plotted, which is a function of the human driven vehicle's position. In order to affect the driver to go left, the reward is higher on the left side of the road in (**a**), and to affect the human to go right in (**b**), the rewards are higher on the right side of the road. In (**c**), the reward of the autonomous vehicle is plotted for scenario 3 with respect to the position of the human driven car. Higher rewards correspond to making the human cross the intersection

Figure 3b contrasts our two conditions. In the control condition, the human moves forward, and might decide to change lanes. *In the experimental condition, however, the robot plans to intentionally occupy two lanes (using either a positive or negative heading), anticipating this will make the human avoid collision by switching into the unoccupied lane.*

4.5 Scenario 3: Make human go first

In this scenario, we demonstrate that an autonomous vehicle can plan to cause the human to proceed first at an intersection.

The vehicles start at the initial conditions depicted on the left in Fig. 3c, with both human and robot stopped at the 4-way intersection. In the experimental condition, we augment the robot's reward with a feature based on the y position of the human car $y_{\mathcal{H}}$ relative to the middle of the intersection y_0. In particular, we used the hyperbolic tangent of the difference, $\tanh(y_{\mathcal{H}} - y_0)$. The reward addition is shown in Fig. 4c.

Figure 3c contrasts our two conditions. In the control condition, the robot proceeds in front of the human. *In the experimental condition, however, the robot plans to intentionally reverse slightly, anticipating that this will induce the human cross first.* We might interpret such a trajectory as communicative behavior, but communication was never explicitly encouraged in the reward function. Instead, *the goal of affecting human actions led to this behavior.*

Reversing at an intersection is perhaps the most surprising result of the three scenarios, because it is not an action human drivers take. In spite of this novelty, our user study suggests that human drivers respond in the expected way: they proceed through the intersection. Further, pedestrians sometimes exhibit behavior like the robot's, stepping back from an intersection in order to let a car pass first.

4.6 Behaviors also emerge from efficiency

Thus far, we have explicitly encoded a desired effect on human actions, and optimized it as a component of the robot's reward. We have also found, however, that behaviors like those we have seen so far can emerge out of the need for efficiency.

Figure 5 (bottom) shows the generated plan for when the robot is given the goal to reach a point in the left lane as quickly as possible (reward shown in Fig. 6). By modeling the effects its actions have on the human actions, the robot plans to merge in front of the person, expecting that they will slow down.

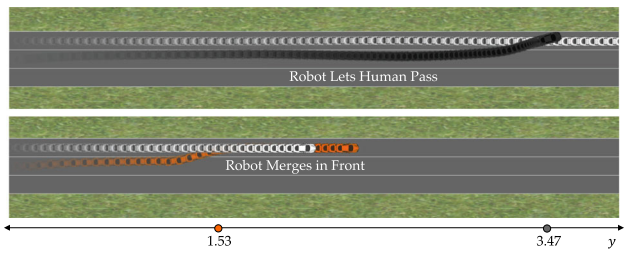
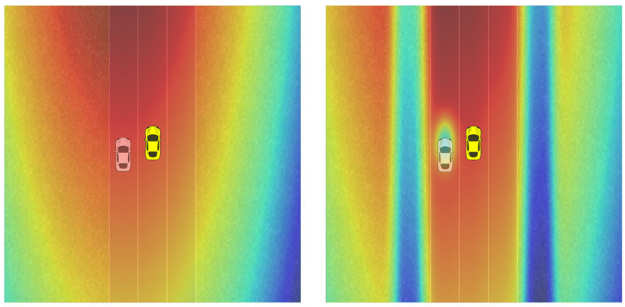


Fig. 5 A time lapse where the autonomous vehicle's goal is to reach a final point in the left lane. In the top scenario, the autonomous vehicle has a simple model of the human driver that does not account for the influence of its actions on the human actions, so it acts more defensively, waiting for the human to pass first. In the bottom, the autonomous vehicle uses the learned model of the human driver, so it acts less defensively and reaches its goal faster



(a) Single goal feature **(b)** Superposition of all features

Fig. 6 Heat map of reward function for reaching a final goal at the top left of the road. As shown in the figure, the goal position is darker showing more reward for reaching that point

In contrast, the top of the figure shows the generated plan for when the robot uses a simple (constant velocity) model of the person. In this case, the robot assumes that merging in front of the person can lead to a collision, and defensively waits for the person to pass, merging behind them.

We hear about this behavior often in autonomous cars today: they are defensive. Enabling them to plan in a manner that is cognizant that they can affect other driver actions can make them more efficient at achieving their goals.

4.7 The robot behavior adapts to the situation

Throughout the case studies, we see examples of coordination behavior that emerges out of planning in our system: going in front of someone knowing they will brake, slowing and nudging into another lane to incentivize a lane change, or backing up to incentivize that the human proceeds first through an intersection. Such behaviors could possibly be hand-engineered for those particular situations rather than autonomously planned. However, we advocate that the need to plan comes from versatility: from the fact that the planner can adapt the exact strategy to each situation.

With this work, we are essentially shifting the design burden from designing policies to designing reward functions. We still need to decide on what we want the robot to do: do we want it to be selfish, do we want it to be extra polite and try to get every human to go first, or do we want it to be somewhere in between? Our paper does not answer this question, and designing good reward functions remains challenging. However, this work does give us the tools to autonomously generate (some of) the strategies needed to then optimize such reward functions when interacting with people. With policies, we'd be crafting that the car should inch forward or backwards from the intersection, specifying by how much and at what velocity, and how this depends on where the other cars are, and how it depends on the type of intersection. With this work, barring local optima issues and the fact that human models could always be improved, all that we need to specify

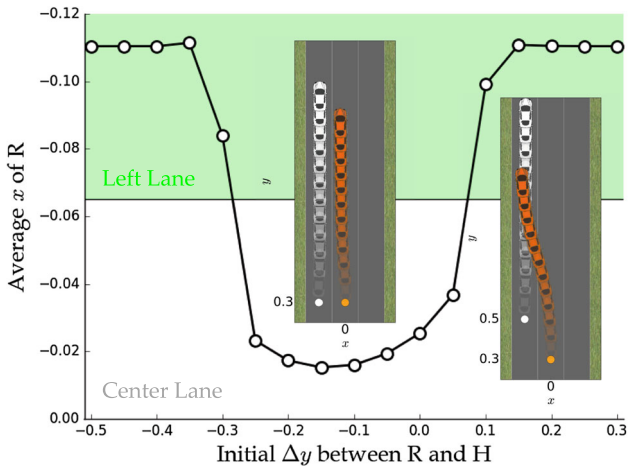


Fig. 7 The robot adapts its merging behavior depending on the relative position of the person: it does not always cut the person off: sometimes it merges behind the person, and if it starts too close (depending on how the reward function is set up) it will not merge at all

is the regular reward that we would give an autonomous car, or, if we want it to purposefully influence human behavior, the desired outcome. The car figures out how to achieve it, adapting its actions to the different settings.

Figure 7 shows a spectrum of behaviors for the robot depending on where it starts relative to the human: from merging behind the person, to not merging, to merging in front.

5 User study with offline estimation

The previous section showed the robot's plans when interacting with a simulated user that perfectly fits the robot's model of the human. Next, we present the results of a user study that evaluates whether the robot can successfully have the desired effects on real users.

5.1 Experimental design

We use the same 3 scenarios as in the previous section.

Manipulated Factors We manipulate a single factor: the ***reward*** that the robot is optimizing, as described in Sect. 4.1. This leads to two conditions: the ***experimental*** condition where the robot is encouraged to have a particular effect on human state though the reward $R_{\text{control}} + R_{\text{affect}}$, and the ***control*** condition where that aspect is left out of the reward function and the robot is optimizing only R_{control} (three conditions for Scenario 2, where we have two experimental conditions, one for the left case and one for the right case).

Dependent Measures For each scenario, we measure the value along the user trajectory of the feature added to the reward function for that scenario, R_{affect}. Specifically, we measure the human's negative squared velocity in Scenario 1, the human's x axis location relative to center in Scenario 2, and whether the human went first or not through the intersection in Scenario 3 (i.e. a filtering of the feature that normalizes for difference in timing among users and measures the desired objective directly).

Hypothesis We hypothesize that our method enables the robot to achieve the effects it desires not only in simulation, but also when interacting with real users:

> The reward function that the robot is optimizing has a significant effect on the measured reward during interaction. Specifically, R_{affect} is higher, as planned, when the robot is optimizing for it.

Subject Allocation We recruited 10 participants (2 female, 8 male). All the participants owned drivers license with at least 2 years of driving experience. We ran our experiments using a 2D driving simulator, we have developed with the driver input provided through driving simulator steering wheel and pedals.

5.2 Analysis

Scenario 1 A repeated measures ANOVA showed the square speed to be significantly lower in the experimental condition than in the control condition ($F(1, 160) = 228.54$, $p < 0.0001$). This supports our hypothesis: the human moved slower when the robot planned to have this effect on the human.

We plot the speed and latitude profile of the human driven vehicle over time for all trajectories in Fig. 8. Figure 8a shows the speed profile of the control condition trajectories in gray, and of the experimental condition trajectories in orange. Figure 8b shows the mean and standard error for each condition. In the control condition, human squared speed keeps increasing. In the experimental condition however, by merging in front of the human, the robot is triggering the human to brake and reduce speed, as planned. The purple trajectory represents a simulated user that perfectly matches the robot's model, showing the ideal case for the robot. The real interaction moves significantly in the desired direction, but does not perfectly match the ideal model, since real users do not act exactly as the model would predict.

The figure also plots the y position of the vehicles along time, showing that the human has not travelled as far forward in the experimental condition.

Scenario 2 A repeated measures ANOVA showed a significant effect for the reward factor ($F(2, 227) = 55.58$, $p < 0.0001$). A post-hoc analysis with Tukey HSD showed that both experimental conditions were significantly different from the control condition, with the user car going more to the left than in the control condition when R_{affect} rewards

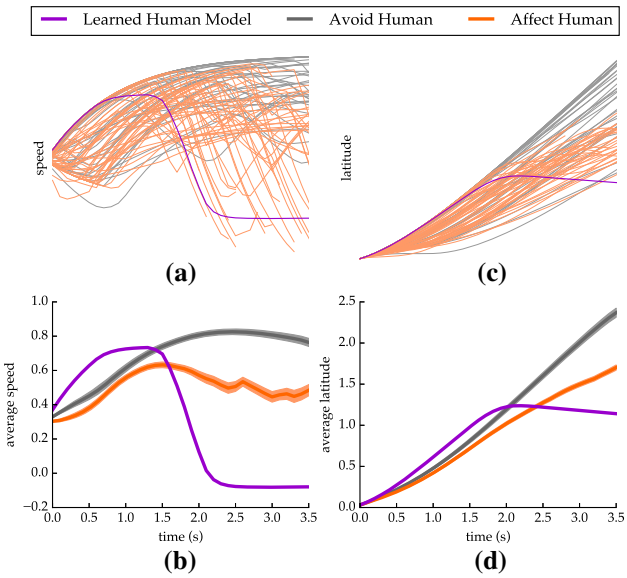


Fig. 8 Speed profile and latitude of the human driven vehicle for Scenario 1. The first column shows the speed of all trajectories with its mean and standard errors in the bottom graph. The second column shows the latitude of the vehicle over time; similarly, with the mean and standard errors. The gray trajectories correspond to the control condition, and the orange trajectories correspond to the experimental condition: the robot decides to merge in front of the users and succeeds at slowing them down. The purple plot corresponds to a simulated user that perfectly matches the model that the robot is using

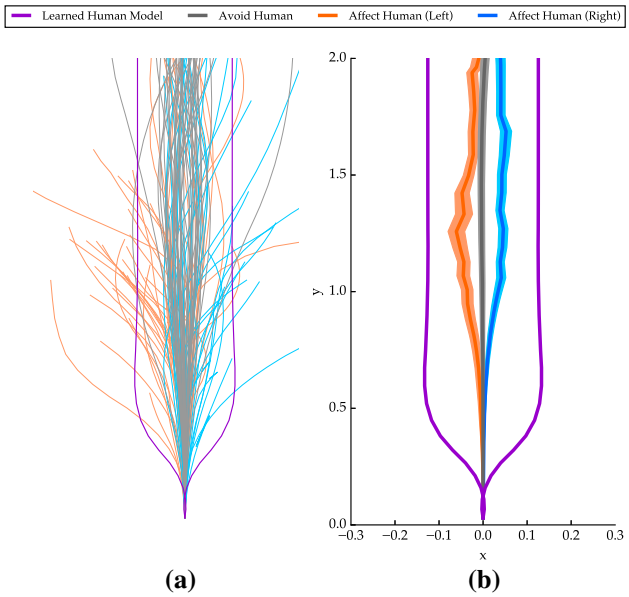


Fig. 9 Trajectories of the human driven vehicle for Scenario 2. The first column **a** shows all the trajectories, and the second column **b** shows the mean and standard error. Orange (blue) indicates conditions where the reward encouraged the robot to affect the user to go left (right)

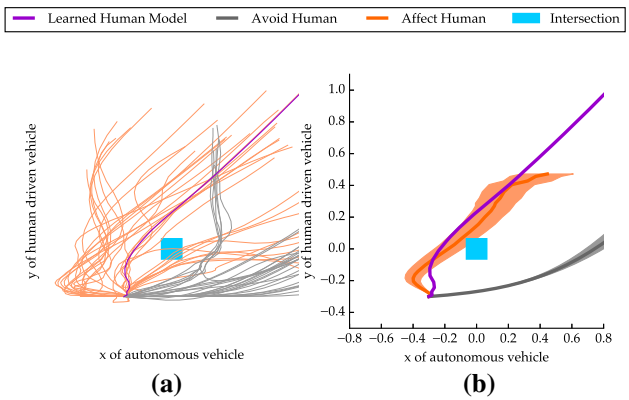


Fig. 10 Plot of $y_{\mathcal{H}}$ with respect to $x_{\mathcal{R}}$. The orange curves correspond to when the autonomous vehicle affects the human to cross the intersection first. The gray curves correspond to the nominal setting

left user positions ($p < 0.0001$), and more to the right in the other case ($p < 0.001$). This supports our hypothesis.

We plot all the trajectories collected from the users in Fig. 9. Figure 9a shows the control condition trajectories in gray, while the experimental conditions trajectories are shown in orange (for left) and blue (for right). By occupying two lanes, the robot triggers an avoid behavior from the users in the third lane. Here again, purple curves show a simulated user, i.e. the ideal case for the robot.

Scenario 3 An ordinal logistic regression with user as a random factor showed that significantly more users went first in the intersection in the experimental condition than in the baseline ($\chi^2(1, 129) = 106.41$, $p < 0.0001$). This supports our hypothesis.

Figure 10 plots the y position of the human driven vehicle with respect to the x position of the autonomous vehicle. For trajectories that have a higher y position for the human vehicle than the x position for the robot, the human car has crossed the intersection before the autonomous vehicle. The lines corresponding to these trajectories travel above the origin, which is shown with a blue square in this figure. The mean of the orange lines travel above the origin, which means that the autonomous vehicle has successfully affected the humans to cross first. The gray lines travel below the origin, i.e. the human crossed second.

Overall, our results suggest that the robot was able to affect the human state in the desired way, even though it does not have a perfect model of the human.

6 Extension to online estimation of the human model

We have thus far in our approximate solution treated the human's reward function as estimated once, offline. This has worked well in our user study on seeking specific coordination effects on the human, like slowing down or going first through the intersection. But in general, this is bound to run

into problems, because not all people behave according to the same estimated $\theta_{\mathcal{H}}$.

Different drivers have different *driving styles*. Some are very defensive, more so than our learned model. Others are much more aggressive, and for instance would not actually brake when the car merges in front of them. Even for the same driver, their style might change over time, for instance when they get distracted on their phone.

In this section we relax our assumption of an offline estimation of the human's reward parameters $\theta_{\mathcal{H}}$. Instead, we explore estimating this online. We introduce an algorithm which maintains a belief over a space of candidate reward functions, and enable the robot to perform inference over this space throughout the interaction. We maintain tractability by clustering possible $\theta_{\mathcal{H}}$s into a few options that the robot maintains a belief over.

6.1 A POMDP with human reward as the hidden variable

The human's actions are influenced by their internal reward parameters $\theta_{\mathcal{H}}$ that the robot does not directly observe. So far, we estimated $\theta_{\mathcal{H}}$ offline and solved an underactuated system, a special case of an MDP. Now, we want to be able to adapt our estimate of $\theta_{\mathcal{H}}$ online, during interaction. This turns the problem into a partially observable markov decision process (POMDP) with $\theta_{\mathcal{H}}$ as the hidden state. By putting $\theta_{\mathcal{H}}$ in the state, we now have a know dynamics model like in the underactuated system before for the robot and the human state, and we assume $\theta_{\mathcal{H}}$ to remain fixed regardless of the robot's actions.

If we could solve the POMDP, the robot would estimate $\theta_{\mathcal{H}}$ from the human's actions, optimally trading off between exploiting it's current belief over $\theta_{\mathcal{H}}$ and actively taking information gathering actions intended to cause human reactions, which result in a better estimate of $\theta_{\mathcal{H}}$.

Because POMDPs cannot be solved tractably, several approximations have been proposed for similar problem formulations (Javdani et al. 2015; Lam et al. 2015; Fern et al. 2007). These approximations are *passively* estimating the human internal state, and exploiting the belief to plan robot actions.[2]

In this work, we take the opposite approach: we focus explicitly on active information gathering Our formulation enables the robot to choose to actively probe the human, and thereby improve its estimate of $\theta_{\mathcal{H}}$. We leverage this method in conjunction with exploitation methods, but the algorithm we present may also be used alone if human internal state (reward parameters) estimation is the robot's primary objective.

[2] One exception is Nikolaidis et al. (2016), who propose to solve the full POMDP, albeit for discrete and not continuous state and action spaces.

6.2 Simplification to information gathering

We denote a belief in the value of the hidden variable, θ, as a distribution $b(\theta)$, and update this distribution according to the likelihood of observing a particular human action, given the state of the world and the human internal state:

$$b^{t+1}(\theta) \propto b^t(\theta) \cdot P(u^t_{\mathcal{H}} \mid x^t, u_{\mathcal{R}}, \theta). \tag{15}$$

In order to update the belief b, we require an observation model. Similar to before, we assume that actions with lower reward are exponentially less likely, building on the principle of maximum entropy (Ziebart et al. 2008):

$$P(u_{\mathcal{H}} \mid x, u_{\mathcal{R}}, \theta) \propto \exp\left(R^{\theta}_{\mathcal{H}}(x^0, \mathbf{u}_{\mathcal{R}}, \mathbf{u}_{\mathcal{H}})\right). \tag{16}$$

To make explicit our emphasis on taking actions which effectively estimate θ, we redefine the robot's reward function to include an information gain term, i.e., the difference between entropies of the current and updated beliefs: $H(b^t) - H(b^{t+1})$. The entropy over the belief $H(b)$ evaluates to:

$$H(b) = -\frac{\sum_{\theta} b(\theta)\log(b(\theta))}{\sum_{\theta} b(\theta)}. \tag{17}$$

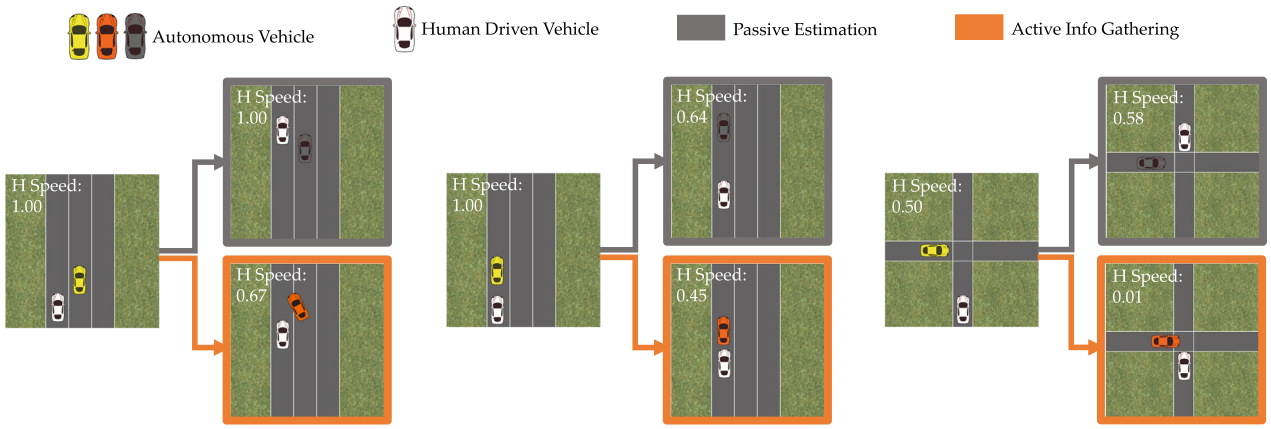
We now optimize our expected reward with respect to the hidden state θ, and this optimization explicitly entails reasoning about the effects that the robot actions will have on the observations, i.e., the actions that the human will take in response, and how useful these observations will be in shattering ambiguity about θ.

6.3 Explore-exploit trade-off

In practice, we use information gathering in conjunction with exploitation. We do not solely optimize the information gain term $H(b^t) - H(b^{t+1})$, but optimize it in conjunction with the robot's actual reward function *assuming the current estimate of* θ:

$$\begin{aligned} r^{augmented}_{\mathcal{R}}(x^t, \mathbf{u}_{\mathcal{R}}, \mathbf{u}_{\mathcal{H}}) = \lambda(H(b^t) - H(b^{t+1})) \\ + r_{\mathcal{R}}(x^t, \mathbf{u}_{\mathcal{R}}, \mathbf{u}_{\mathcal{H}}, b^t) \end{aligned} \tag{18}$$

At the very least, we do this as a measure of safety, e.g., we want an autonomous car to keep avoiding collisions even when it is actively probing a human driver to test their reactions. We choose λ experimentally, though existing techniques that can better adapt λ over time (Vanchinathan et al. 2014).



(a) Scenario 1: Nudging in to Explore on a Highway **(b)** Scenario 2: Braking to Explore on a Highway **(c)** Scenario 3: Nudging in to Explore at an Intersection

Fig. 11 Our three scenarios, along with a comparison of robot plans for passive estimation (gray) versus active information gathering (orange). In the active condition, the robot is purposefully nudging in or braking to test human driver attentiveness. The color of the autonomous car in the initial state is yellow, but changes to either gray or orange in cases of passive and active information gathering respectively

6.4 Solution via model predictive control

To find the control inputs for the robot we locally solve:

$$\mathbf{u}_{\mathcal{R}}^{*} = \arg\max_{\mathbf{u}_{\mathcal{R}}} \mathbb{E}_{\theta}\left[R_{\mathcal{R}}\left(x^{0}, \mathbf{u}_{\mathcal{R}}, \mathbf{u}_{\mathcal{H}}^{*,\theta}(x^{0}, \mathbf{u}_{\mathcal{R}})\right)\right] \quad (19)$$

over a finite horizon N, where $\mathbf{u}_{\mathcal{H}}^{*,\theta}(x^0, \mathbf{u}_{\mathcal{R}})$ corresponds to the actions the human *would* take from state x^0 if the robot executed actions $\mathbf{u}_{\mathcal{R}}$. This objective generalizes (5) with an expectation over the current belief over θ, b^0.

We still assume that the human maximizes their own reward function, $r_{\mathcal{H}}^{\theta}(x^t, u_{\mathcal{R}}^t, u_{\mathcal{H}}^t)$; we add the superscript θ to indicate the dependence on the hidden state. We can write the sum of human rewards over horizon N as:

$$R_{\mathcal{H}}^{\theta}(x^0, \mathbf{u}_{\mathcal{R}}, \mathbf{u}_{\mathcal{H}}) = \sum_{t=0}^{N-1} r_{\mathcal{H}}^{\theta}(x^t, u_{\mathcal{R}}^t, u_{\mathcal{H}}^t) \quad (20)$$

Computing this over the continuous space of possible reward parameters θ is intractable even with discretization. Instead, we learn clusters of θs offline via IRL, and online use estimation to figure out which cluster best matches the human.

Despite optimizing the trade-off in (18), we do not claim that our method as-is can better solve the general POMDP formulation: only that it can be used to get better estimates of human internal state. Different tradeoffs λ will result in different performance. Our results below emphasize the utility of gathering information, but also touch on the implications for active information gathering on $R_{\mathcal{R}}$.

7 Case studies with online estimation

In this section, we show simulation results that use the method from the previous section to estimate human driver type in the interaction between an autonomous vehicle and a human-driven vehicle. We consider three different autonomous driving scenarios. In these scenarios, the human is either distracted or attentive during different driving experiments. The scenarios are shown in Fig. 11, where the yellow car is the autonomous vehicle, and the white car is the human driven vehicle. Our goal is to plan to actively estimate the human's driving style in each one of these scenarios, by using the robot's actions.

7.1 Attentive versus distracted human driver models

Our technique requires reward functions $r_{\mathcal{H}}^{\theta}$ that model the human behavior for a particular internal state θ. We obtain a generic driver model via Continuous Inverse Optimal Control with Locally Optimal Examples (Levine and Koltun 2012) from demonstrated trajectories in a driving simulator in an environment with multiple autonomous cars, which followed precomputed routes.

We then adjust the learned weights to model attentive versus distractive drivers. Specifically, we modify the weights of the collision avoidance features, so the distracted human model has less weight for these features. Therefore, the distracted driver is more likely to collide with the other cars while the attentive driver has high weights for the collision avoidance feature. In future work, we plan to investigate ways of automatically clustering learned $\theta_{\mathcal{H}}$s from data from different users, but we show promising results even with these simple options.

7.2 Manipulated factors

We manipulate the *reward* function that the robot is optimizing. In the ***passive*** condition, the robot optimizes a simple reward function for collision avoidance based on the current belief estimate. It then updates this belief passively, by observing the outcomes of its actions at every time step. In the ***active*** condition, the robot trades off between this reward function and information gain in order to explore the human's driving style.

We also manipulate the *human internal reward parameters* to be ***attentive*** or ***distracted***. The human is simulated to follow the ideal model of reward maximization for our two rewards.

7.3 Scenarios and qualitative results

Scenario 1: Nudging In to Explore on a Highway In this scenario, we show an autonomous vehicle actively exploring the human's driving style in a highway driving setting. We contrast the two conditions in Fig. 11a. In the passive condition, the autonomous car drives on its own lane without interfering with the human throughout the experiment, and updates its belief based on passive observations gathered from the human car. *However, in the active condition, the autonomous car actively probes the human by nudging into her lane in order to infer her driving style. An attentive human significantly slows down (timid driver) or speeds up (aggressive driver) to avoid the vehicle, while a distracted driver might not realize the autonomous actions and maintain their velocity, getting closer to the autonomous vehicle.* It is this difference in reactions that enables the robot to better estimate θ.

Scenario 2: Braking to Explore on a Highway In the second scenario, we show the driving style can be explored by the autonomous car probing the human driver behind it. The two vehicles start in the same lane as shown in Fig. 11b, where the autonomous car is in the front. In the passive condition, the autonomous car drives straight without exploring or enforcing any interactions with the human driven vehicle. *In the active condition, the robot slows down to actively probe the human and find out her driving style. An attentive human would slow down and avoid collisions while a distracted human will have a harder time to keep safe distance between the two cars.*

Scenario 3: Nudging In to Explore at an Intersection In this scenario, we consider the two vehicles at an intersection, where the autonomous car actively tries to explore human's driving style by nudging into the intersection. The initial conditions of the vehicles are shown in Fig. 11c. In the passive condition, the autonomous car stays at its position without probing the human, and only optimizes for collision avoidance. This provides limited observations from the human car resulting in a low confidence belief distribution. *In the active condition, the autonomous car nudges into the intersection to probe the driving style of the human. An attentive human would slow down to stay safe at the intersection while a distracted human will not slow down.*

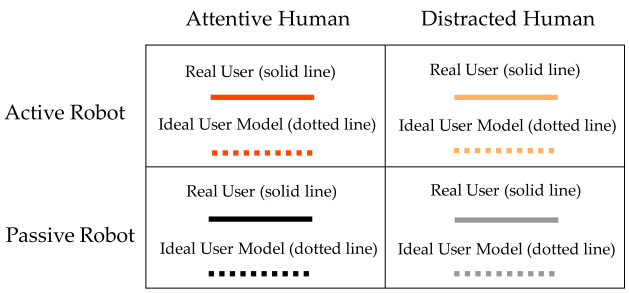


Fig. 12 Legends indicating active/passive robots, attentive/distracted humans, and real user/ideal model used for all following figures

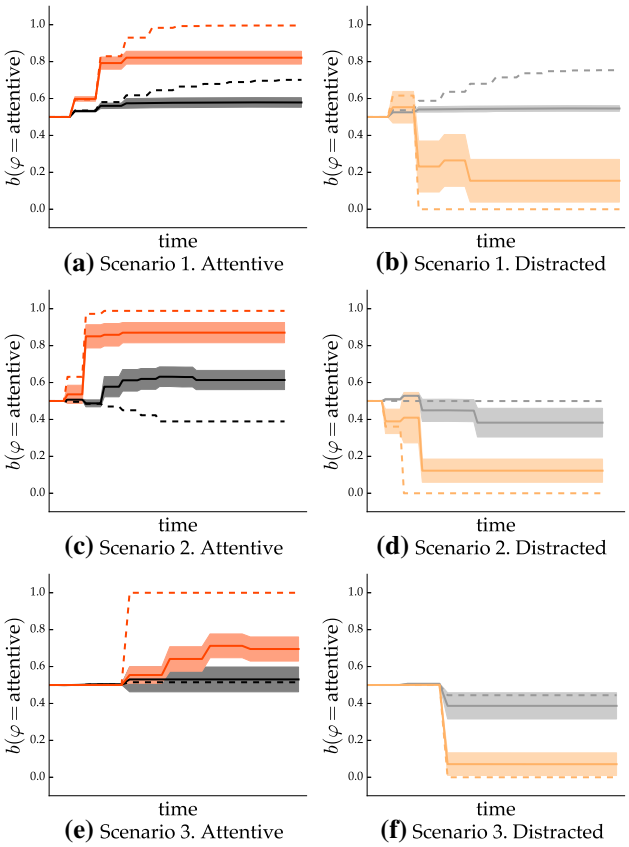


Fig. 13 The probability that the robot assigns to attentive as a function of time, for the attentive (left) and distracted (right). Each plot compares the active algorithm to passive estimation, showing that active information gathering leads to more accurate state estimation, in simulation and with real users

7.4 Quantitative results

Throughout the remainder of the paper, we use a common color scheme to plot results for our experimental conditions. We show this common scheme in Fig. 12: darker colors (black and red) correspond to attentive humans, and lighter colors (gray and orange) correspond to distracted humans. Further, the shades of orange correspond to active information gathering, while the shades of gray indicate passive information gathering. We also use solid lines for real users, and dotted lines for scenarios with an ideal user model learned through inverse reinforcement learning.

Figure 13 plots, using dotted lines, the beliefs over time for the attentive (left) and distracted (right) conditions, comparing in each the passive (dotted black and gray respectively) with the active method (dotted dark orange and light orange respectively). In every situation, the active method achieves a more accurate belief (higher values for attentive on the left, when the true θ is attentive, and lower values on the right, when the true θ is distracted). In fact, passive estimation sometimes incorrectly classifies drivers as attentive when they are distracted and vice-versa.

The same figure also shows (in solid lines) results from our user study of what happens when the robot no longer interacts with an ideal model. We discuss these in the next section.

Figures 14 and 15 plot the corresponding robot and human trajectories for each scenario. The important takeaway from these figures is that there tends to be a larger gap between attentive and distracted human trajectories in the active condition (orange shades) than in the passive condition (gray shades), especially in scenarios 2 and 3. It is this difference that helps the robot better estimate θ: *the robot in the active condition is purposefully choosing actions that will lead to large differences in human reactions*, in order to more easily determine the human driving style.

7.5 Robot behavior adapts to the situation

As Fig. 14 suggests, active info gathering results in interesting coordination behavior. In Scenario 1, the robot decides to nudge into the person's lane. But what follows next nicely reacts to the person's driving style. The robot proceeds with the merge if the person is attentive, but actually *goes back to its lane* if the person is distracted. Even more interesting is what happens in Scenario 3 at the 4way stop. The robot inches forward into the intersection, and proceeds if the person is attentive, but actually *goes back* to allow the person through if they are distracted! These all emerge as the optima in our system.

The behavior also naturally changes as the initial state of the system changes. Figure 16 shows different behaviors arising from an attentive driver model but different initial

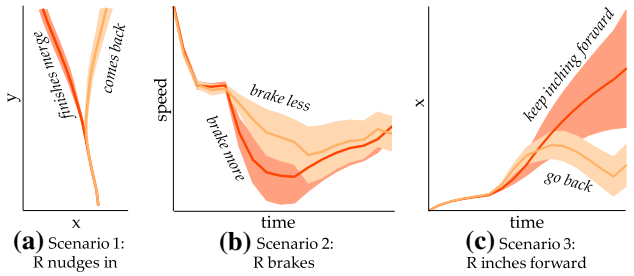


Fig. 14 Robot trajectories for each scenario in the active information gathering condition. The robot acts differently when the human is attentive (dark orange) versus when the human is distracted (light orange) due to the trade-off with safety

position of the human driver. This shows that even for the same driver model, the robot intelligently adapts its coordination behavior to the situation, sometimes deciding to merge but sometimes not.

This is particularly important, because it might be easy to handcode these coordination strategies for a particular situation. *Much like in the onffline estimation case, the robot not only comes up with these strategies, but actually adapts them depending on the situation—the driver style, the initial state, and so on.*

7.6 Active information gathering helps the robot's actual reward

So far, we have looked at how active information gathering improves estimation of the driver model. This is useful in itself in situations where human internal state estimation is the end-goal. But it is also useful for enabling the robot to better achieve its goal.

Intuitively, knowing the human driving style more accurately should improve the robot's ability to collect reward. For instance, if the robot starts unsure of whether the human is paying attention or not, but collects enough evidence that she is, then the robot can safely merge in front of the person and be more efficient. Of course, this is not always the case. If the person is distracted, then the information gathering actions could be a waste because the robot ends up not merging anyway.

Figure 17 shows what happens in the merging scenario: the robot gains more reward compared to passive estimation by doing information gathering with attentive drivers, because it figures out it is safe to merge in front of them; the robot looses some reward compared to passive estimation with distracted drivers, because it makes the effort to nudge in but has to retreat back to its lane anyway because it cannot merge.

Of course, all this depends on choosing λ, the trade-off between exploitation and exploration (information gain). Figure 18 shows the effect of λ has on the robot's goal reward (not its information gain reward), which shows that not all λs

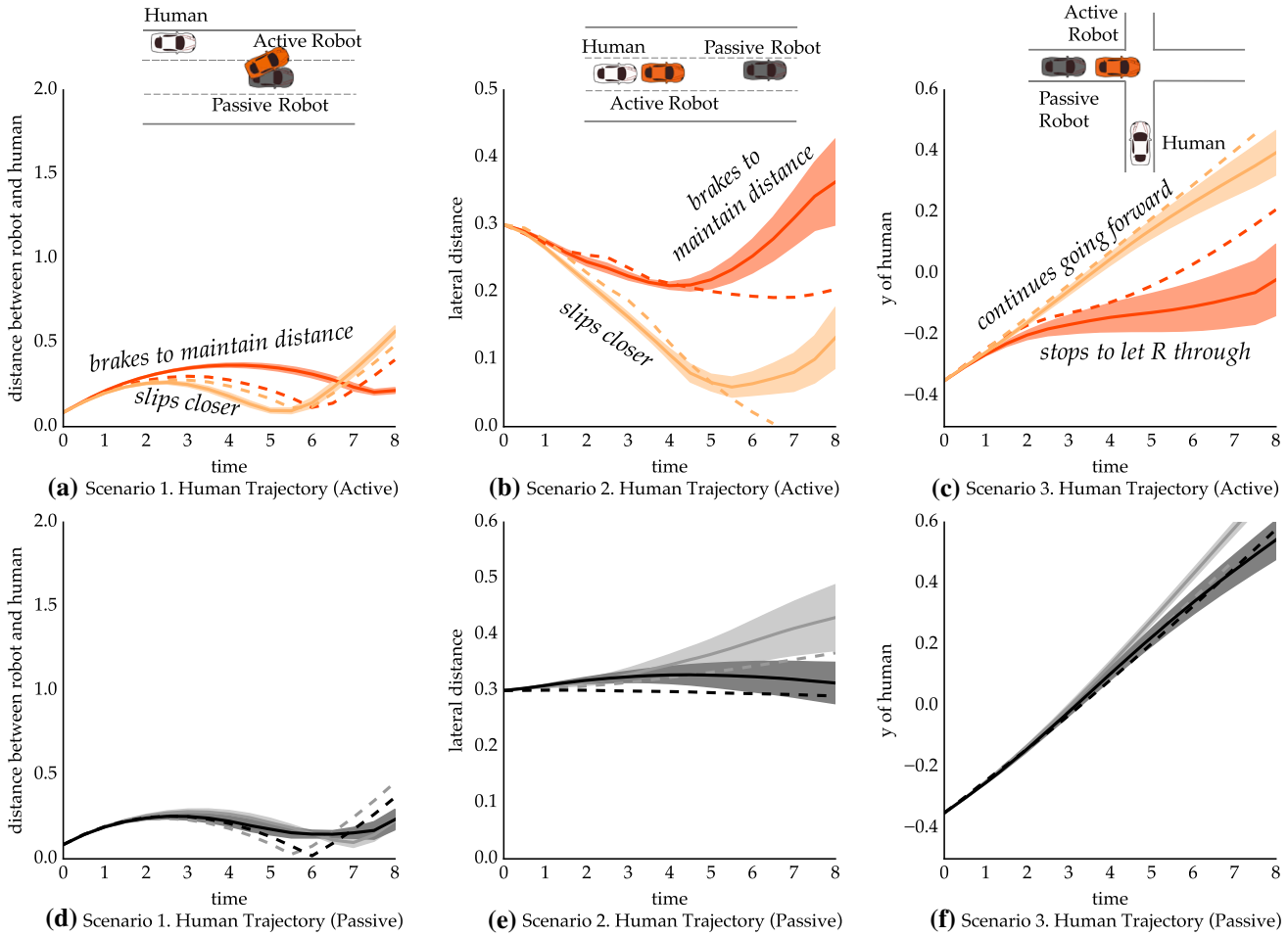


Fig. 15 The user trajectories for each scenario. The gap between attentive and distracted drivers' actions is clear in the active information gathering case (first row)

are useful. In other situations, we would also expect to see a large decrease in reward from too much weight on information gain.

7.7 Beyond driving style: active intent inference

Driving style is not the only type of human internal state that our method enables robots to estimate. If the human has a goal, e.g. of merging into the next lane or not, or of exiting the highway or not, the robot could estimate this as well using the same technique.

Each possible goal corresponds to a feature. When estimating which goal the human has, the robot is deciding among θs which place weight on only one of the possible goal features, and 0 on the others. Figure 19 shows the behavior that emerges from estimating whether the human wants to merge into the robot's lane. In the passive case, the human is side by side with the robot. Depending on the driving style, they might slow down slightly, accelerate slightly, or start nudging into the robot's lane, but since the observation model is noisy the robot does not get quite enough confidence in the human's intent early on. Depending on the

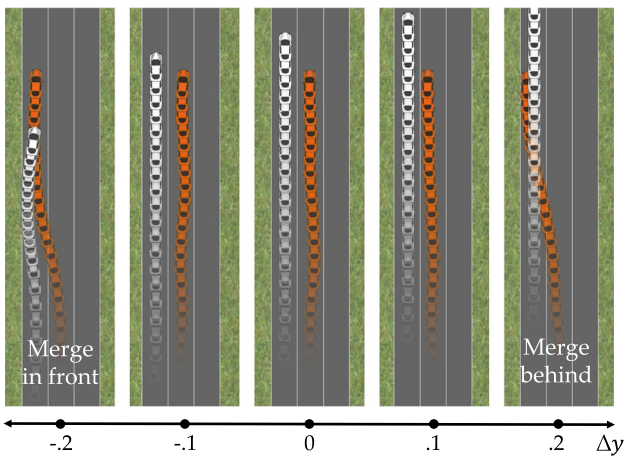


Fig. 16 Effect of varying the initial condition (relative y position) in the active merge scenario. The robot adapts to merge when feasible and avoid otherwise. The human is attentive in all cases

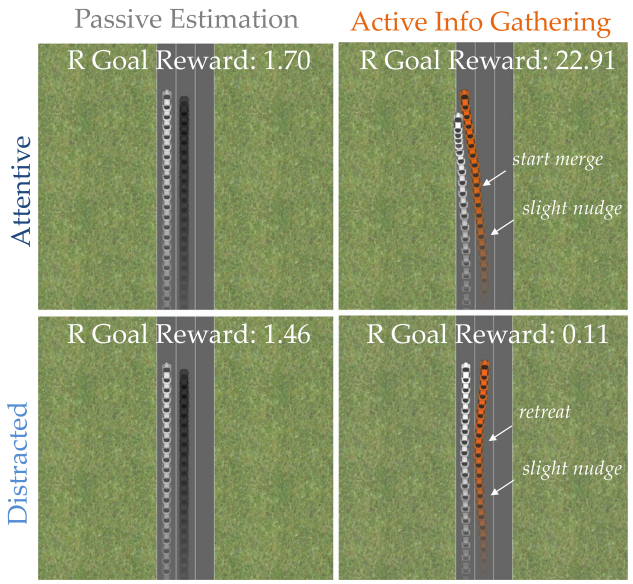


Fig. 17 Active info gathering improves the robot's ability to efficiently achieve its goal in the case when the human is attentive: where a passive estimator never gets enough information to know that the person is paying attention, an active estimator nudges in, updates its belief, and proceeds with the merge. At the same time, active info gathering does not hurt too much when the person is distracted: the robot nudges in slightly (this does decrease its reward relative to the passive case, but not by much), updates its belief, and retreats to its lane

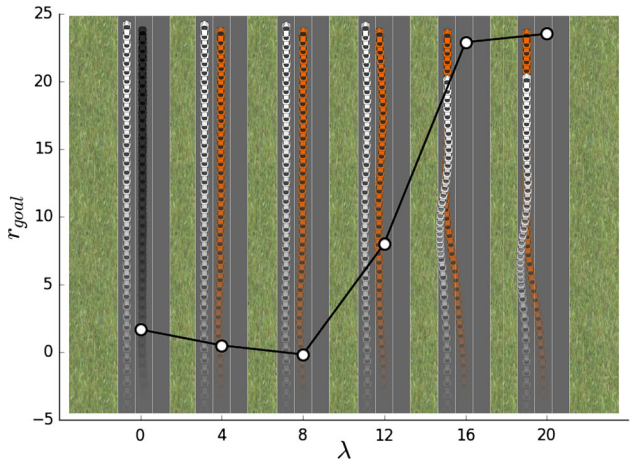
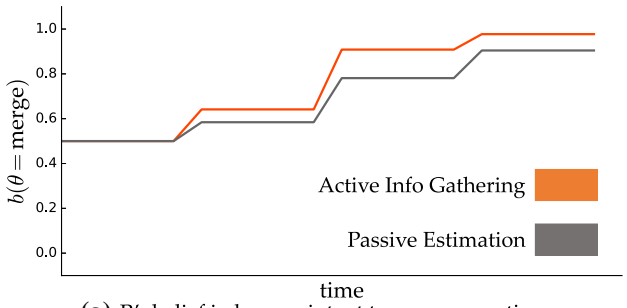
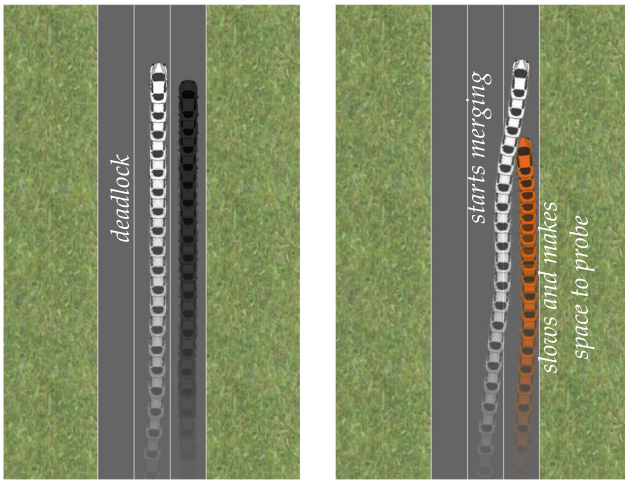


Fig. 18 Active information gathering behavior when the robot's goal is to merge into the left lane for different values of λ, together with the reward the robot obtains. $\lambda = 0$ results in low reward because the robot does not figure out that the person is attentive and does not merge. A small λ hurts the reward because the information gathering costs but does not buy anything. For higher values, the robot gets enough information that it forces a merge in front of the human



(a) R's belief in human intent to merge over time



(b) Interaction from passive (left) and active (right) estimation

Fig. 19 Actively estimating the human's intent (whether they want to merge in the right lane or not). The robot slows down and shifts slightly away from the person, which would make someone who wants to merge proceed. This could be useful for robots trying to optimize for the good of all drivers (rather than for their selfish reward function)

robot's reward, it might take a long time before the person can merge. In the active case, the robot decides to probe the person by slowing down and shifting away from the person in order to make room. It then becomes optimal for the person wanting to merge to start shifting towards the robot's lane, giving the robot enough information now to update its belief. In our experiment, we see that this is enough for the person to be able to complete the merge faster, despite the robot not having any incentive to help the person in its reward.

8 User study with online estimation

In the previous section, we explored planning for an autonomous vehicle that actively probes a human's driving style, by braking or nudging in and expecting to cause reactions from the human driver that would be different depending on their style. We showed that active exploration does significantly better at distinguishing between attentive and distracted drivers using simulated (ideal) models of drivers. Here, we show the results of a user study that evaluates this active exploration for attentive and distracted human drivers.

8.1 Experimental design

We use the same three scenarios discussed in the previous section.

Manipulated Factors We manipulated the same two factors as in our simulation experiments: the *reward* function that the robot is optimizing (whether it is optimizing its reward through passive state estimation, or whether it is trading off with active information gathering), and the *human internal state* (whether the user is attentive or distracted). We asked our users to pay attention to the road and avoid collisions for the attentive case, and asked our users to play a game on a mobile phone during the distracted driving experiments.

Dependent Measure We measured the probability that the robot assigned along the way to the human internal state.

Hypothesis *The active condition will lead to more accurate human internal state estimation, regardless of the true human internal state.*

Subject Allocation We recruited 8 participants (2 female, 6 male) in the age range of 21–26 years old. All participants owned a valid driver license and had at least 2 years of driving experience. We ran the experiments using a 2D driving simulator with the steering input and acceleration input provided through a steering wheel and a pedals as shown in Fig. 1. We used a within-subject experiment design with counterbalanced ordering of the four conditions.

8.2 Analysis

We ran a factorial repeated-measures ANOVA on the probability assigned to "attentive", using reward (active vs. passive) and human internal state (attentive vs. distracted) as factors, and time and scenario as covariates. As a manipulation check, attentive drivers had significantly higher estimated probability of "attentive" associated than distracted drivers (0.66 vs 0.34, $F = 3080.3$, $p < 0.0001$). More importantly, there was a signifiant interaction effect between the factors ($F = 1444.8$, $p < 0.000$). We ran a post-hoc analysis with Tukey HSD corrections for multiple comparisons, which showed all four conditions to be significantly different from each other, all contrasts with $p < 0.0001$. In particular, the active information gathering did end up with higher probability mass on "attentive" than the passive estimation for the attentive users, and lower probability mass for the distracted user. This supports our hypothesis that our method works, and active information gathering is better at identifying the correct state.

Figure 13 compares passive (grays and blacks) and active (light and dark oranges) across scenarios and for attentive (left) and distracted (right) users. It plots the probability of attentive over time, and the shaded regions correspond to standard error. From the first column, we can see that our algorithm in all cases detects human's attentiveness with much higher probably than the passive information gathering technique shown in black. From the second column, we see that our algorithm places significantly lower probability on attentiveness, which is correct because those users were distracted users. These are in line with the statistical analysis, with active information gathering doing a better job estimating the true human internal state.

Figure 14 plots the robot trajectories for the active information gathering setting. Similar to Fig. 13, the solid lines are the mean of robot trajectories and the shaded regions show the standard error. We plot a representative dimension of the robot trajectory (like position or speed) for attentive (dark orange) or distracted (light orange) cases. The active robot probed the user, but ended up taking different actions when the user was attentive versus distracted in order to maintain safety. For example, in Scenario 1, the trajectories show the robot is nudging into the human's lane, but the robot decides to move back to its own lane when the human drivers are distracted (light orange) in order to stay safe. In Scenario 2, the robot brakes in front of the human, but it brakes less when the human is distracted. Finally, in Scenario 3, the robot inches forward, but again it stops when if the human is distracted, and even backs up to make space for her.

Figure 15 plots the user trajectories for both active information gathering (first row) and passive information gathering (second row) conditions. We compare the reactions of distracted (light shades) and attentive (dark shades) users. There are large differences directly observable, with user reactions tending to indeed cluster according to their internal state. These differences are much smaller in the passive case (second row, where distracted is light gray and attentive is black). For example, in Scenario 1 and 2, the attentive users (dark orange) keep a larger distance to the car that nudges in front of them or brakes in front of them, while the distracted drivers (light orange) tend to keep a smaller distance. In Scenario 3, the attentive drivers tend to slow down and do not cross the intersection, when the robot actively inches forward. None of these behaviors can be detected clearly in the passive information gathering case (second row). This is the core advantage of active information gathering: the actions are purposefully selected by the robot such that users would behave drastically differently depending on their internal state, clarifying to the robot what this state actually is.

Overall, these results support our simulation findings, that our algorithm performs better at estimating the true human internal state by leveraging purposeful information gathering actions.

9 Discussion

Summary In this paper, we took a step towards autonomously producing behavior for interaction and coordination between

autonomous cars and human-driven vehicles. We formulated a dynamical system in which the robot accounts for how its actions are going to influence those of the human as a simplification to a partially observable stochastic game. We introduced approximations for optimizing in the dynamical system that bring the robot's computation close to real time (.3 s/time step). We showed in an empirical analysis that when the robot estimates the human model offline, it produces behavior that can purposefully modify the human behavior: merging in front of them to get them to slow down, or pulling back at an intersection to incentivize them to proceed first through. We also showed that these behaviors can emerge out of directly optimizing for the robot's efficiency.

We further introduced an online estimation algorithm in which the robot actively uses its actions to gather information about the human model so that it can better plan its own actions. Our analysis again shows coordination strategies arising out of planning in our formulation: the robot nudges into someones's lane to check if the human is paying attention, and only completes the merge if they are; the robot inches forward at an intersection, again to check if the human is paying attention, and proceeds if they are, but backs up to let them through if they are not; the robot slows down slightly and shifts in its lane away from the human driver to check if they want to merge into its lane or not.

Importantly, these behaviors change with the human driver style and with the initial conditions—the robot takes different actions in different situations, emphasizing the need to start generating such coordination behavior autonomously rather than relying on hand coded strategies. Even more importantly, the behaviors seem to work when the robot is planning and interacting with real users.

Limitations All this work happened in a simple driving simulator. To put this on the road, we will need more emphasis on safety, as well as a longer planning horizon.

While performing these experiments, we found the robot's nominal reward function (trading off between safety and reaching a goal) to be insufficient—in some cases it led to getting dangerously close to the human vehicle and even collisions, going off the road, oscillating in the lane due to minor asymmetries in the environment, etc.

Figure 20 shows an example of such behavior that comes from the 4way stop domain. For the most part, the car plans to back up to incentivize the human to go through first. But for some values of the human's initial velocity, we observed bad behavior, likely due to convergence to local maxima: the car did not figure out to slow down or back up, and instead if proceeded forward—then it tried to avoid collisions with the person and went off the road, and in the wrong direction (i.e. in the person's way).

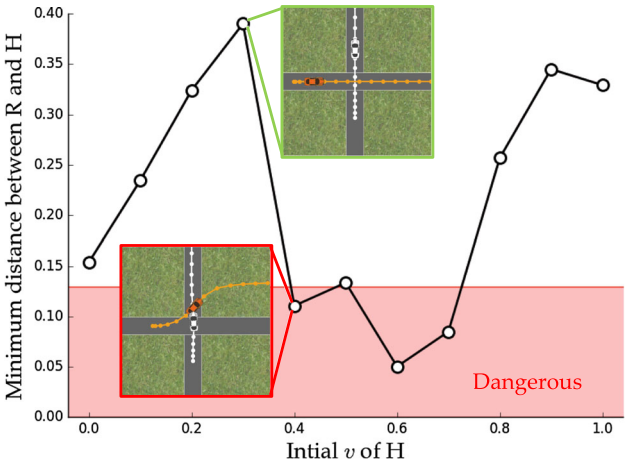


Fig. 20 Example of bad local optima occurring for certain initial velocities of the human in the 4way intersection scenario

It seems like while a reward function might be a good enough model for the human, it might be difficult to devise such a universal function for the robot, and the use of hard constraints to ensure safe control would be welcome.

Another limitation is that we currently focus on a single human driver. Looking to the interaction among multiple vehicles is not just a computational challenge, but also a modeling one—it is not immediately clear how to formulate the problem when multiple human-driven vehicles are interacting and reacting to each other.

Conclusion Despite these limitations, we are encouraged to see autonomous cars generate human-interpretable behaviors through optimization, without relying on hand-coded heuristics. Even though in this work we have focused on modeling the interaction between an autonomous car and a human-driven car, the same framework of *underactuated systems* can be applied to modeling the interaction between humans and robots in more general settings. We look forward to applications of these ideas beyond autonomous driving, to mobile robots, UAVs, and in general to human–robot interactive scenarios where robot actions can influence human actions.

Acknowledgements This work was partially supported by Berkeley DeepDrive, NSF VeHICaL 1545126, NSF Grants CCF-1139138 and CCF-1116993, ONR N00014-09-1-0230, NSF CAREER 1652083, and an NDSEG Fellowship.

References

Abbeel, P., & Ng, A. Y. (2005). Exploration and apprenticeship learning in reinforcement learning. In *Proceedings of the 22nd international conference on Machine learning* (pp. 1–8). ACM.

Agha-Mohammadi, A.-A., Chakravorty, S., & Amato, N. M. (2014). FIRM: Sampling-based feedback motion-planning under motion

uncertainty and imperfect measurements. *The International Journal of Robotics Research*, *33*(2), 268–304.
Andrew, G.,& Gao, J. (2007). Scalable training of L1-regularized log-linear models. In *Proceedings of the 24th international conference on Machine learning* (pp. 33–40). ACM.
Atanasov, N. A. (2015). Active information acquisition with mobile robots.
Atanasov, N., Ny Le J., Daniilidis, K. & Pappas, G. J. (2014). Information acquisition with sensing robots: Algorithms and error bounds. In *2014 IEEE international conference on robotics and automation (ICRA)* (pp. 6447–6454). IEEE.
Aumann, R. J., Maschler, M., & Stearns, R. E. (1995). *Repeated games with incomplete information*. Cambridge: MIT Press.
Bandyopadhyay, T., Won, K. S., Frazzoli, E., Hsu, D., Lee, W. S., & Rus, D.(2013). Intention-aware motion planning. In *Algorithmic foundations of robotics X* (pp. 475–491). Springer.
Bastien, F., Lamblin, P., Pascanu, R., Bergstra, J., Goodfellow, I. J., Bergeron, A., Bouchard, N., & Bengio, Y. (2012). Theano: new features and speed improvements. In *Deep learning and unsupervised feature learning NIPS 2012 workshop*.
Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., Turian, J., Warde-Farley, D., & Bengio, Y. (2010). Theano: a CPU and GPU math expression compiler. In *Proceedings of the python for scientific computing conference (SciPy)*, Oral Presentation.
Bernstein, D. S., Givan, R., Immerman, N., & Zilberstein, S. (2002). The complexity of decentralized control of Markov decision processes. *Mathematics of Operations Research*, *27*(4), 819–840.
Camacho, E. F., & Alba, C. B. (2013). *Model predictive control*. Berlin: Springer.
Chaudhari, P., Karaman, S., Hsu, D., & Frazzoli, E. (2013). Sampling-based algorithms for continuous-time POMDPs. In *American control conference (ACC), 2013* (pp. 4604–4610). IEEE.
Dissanayake, M., Newman, P., Clark, S., Durrant-Whyte, H. F., & Csorba, M. (2001). A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on Robotics and Automation*, *17*(3), 229–241.
Falcone, P., Borrelli, F., Asgari, J., Tseng, H. E., & Hrovat, D. (2007). Predictive active steering control for autonomous vehicle systems. *IEEE Transactions on Control Systems Technology*, *15*(3), 566–580.
Falcone, P., Borrelli, F., Tseng, H. E., Asgari, J., & Hrovat, D. (2007). Integrated braking and steering model predictive control approach in autonomous vehicles. *Advances in Automotive Control*, *5*, 273–278.
Falcone, P., Tseng, H. E., Borrelli, F., Asgari, J., & Hrovat, D. (2008). MPC-based yaw and lateral stabilisation via active front steering and braking. *Vehicle System Dynamics*, *46*(sup1), 611–628.
Fern, A., Natarajan, S., Judah, K., & Tadepalli, P. (2007). *A decision-theoretic model of assistance*. In *IJCAI*.
Fudenberg, D., & Tirole, J. (1991). *Game theory* (Vol. 393). Cambridge, Massachusetts.
Gray, A., Gao, Y., Hedrick, J. K. & Borrelli, F.(2013). Robust predictive control for semi-autonomous vehicles with an uncertain driver model. In *Intelligent vehicles symposium (IV), 2013 IEEE* (pp. 208–213). IEEE.
Hansen, E. A., Bernstein, D. S., & Zilberstein, S. (2004). Dynamic programming for partially observable stochastic games. *AAAI*, *4*, 709–715.
Hedden, T., & Zhang, J. (2002). What do you think i think you think?: Strategic reasoning in matrix games. *Cognition*, *85*(1), 1–36.
Hermes, C., Wohler, C., Schenk, K., & Kummert, F. (2009). Long-term vehicle motion prediction. In *2009 IEEE intelligent vehicles symposium* (pp. 652–657).
Javdani, S., Bagnell, J. A., & Srinivasa, S. (2015). Shared autonomy via hindsight optimization. arXiv preprint arXiv:1503.07619.
Javdani, S., Klingensmith, M., Bagnell, J. A., Pollard, N. S., & Srinivasa, S. S.(2013). Efficient touch based localization through submodularity. In *2013 IEEE international conference on robotics and automation (ICRA)* (pp. 1828–1835). IEEE.
Kuderer, M., Gulati, S., & Burgard, W. (2015). Learning driving styles for autonomous vehicles from demonstration. In *Proceedings of the IEEE international conference on robotics & automation (ICRA), Seattle, USA*, Vol. 134.
Lam, C.-P., Yang, A. Y., & Sastry, S. S.(2015). An efficient algorithm for discrete-time hidden mode stochastic hybrid systems. In *Control conference (ECC), 2015 European*. IEEE.
Leonard, J., How, J., Teller, S., Berger, M., Campbell, S., Fiore, G., et al. (2008). A perception-driven autonomous urban vehicle. *Journal of Field Robotics*, *25*(10), 727–774.
Levine, S, & Koltun, V.(2012). Continuous inverse optimal control with locally optimal examples. arXiv preprint arXiv:1206.4617.
Levinson, J., Askeland, J., Becker, J., Dolson, J., Held, D., Kammel, S., Kolter, J. Z., Langer, D., Pink, O., Pratt, V., et al. (2011). Towards fully autonomous driving: Systems and algorithms. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 163–168.
Luders, B., Kothari, M., & How, J. P.(2010). Chance constrained RRT for probabilistic robustness to environmental uncertainty. In *AIAA guidance, navigation, and control conference (GNC)*, Toronto, Canada.
Ng, A. Y., Russell, S. J., et al.(2000). Algorithms for inverse reinforcement learning. In *Proceedings of the 17th international conference on Machine learning*, pages 663–670.
Nikolaidis, S., Kuznetsov, A., Hsu, D., & Srinivasa, S. (2016). Formalizing human-robot mutual adaptation via a bounded memory based model. In *Human-robot interaction*.
Nikolaidis, S., Ramakrishnan, R., Gu, K., & Shah, J. (2015). Efficient model learning from joint-action demonstrations for human-robot collaborative tasks. In *Proceedings of the tenth annual ACM/IEEE international conference on human-robot interaction* (pp. 189–196). ACM.
Patil, S., Kahn, G., Laskey, M., Schulman, J., Goldberg, K., & Abbeel, P. (2015). Scaling up Gaussian belief space planning through covariance-free trajectory optimization and automatic differentiation. In *Algorithmic foundations of robotics XI* (pp. 515–533). Springer.
Prentice, S., & Roy, N. (2009). The belief roadmap: Efficient planning in belief space by factoring the covariance. *The International Journal of Robotics Research*, *28*, 1448–1465.
Raman, V., Donzé, A., Sadigh, D., Murray, R. M., & Seshia, S. A. (2015). Reactive synthesis from signal temporal logic specifications. In *Proceedings of the 18th international conference on hybrid systems: Computation and control* (pp. 239–248). ACM.
Sadigh, D., & Kapoor, A. (2015). Safe control under uncertainty. arXiv preprint arXiv:1510.07313.
Sadigh, D., Sastry, S. A., Seshia, S., & Dragan, A. D. (2016a). Planning for autonomous cars that leverages effects on human actions. In *Proceedings of the robotics: Science and systems conference (RSS)*.
Sadigh, D., Sastry, S. S., Seshia, S. A., & Dragan, A. (2016b). Information gathering actions over human internal state. In *IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 66–73). IEEE.
Sadigh, D., Sastry, S. S., Seshia, S. A., & Dragan, A. (2016c). Planning for autonomous cars that leverages effects on human actions. In *Proceedings of the robotics: Science and systems conference (RSS)*.
Seiler, K. M., Kurniawati, H., & Singh, S. P. (2015). An online and approximate solver for POMDPs with continuous action space. In *2015 IEEE international conference on robotics and automation (ICRA)* (pp. 2290–2297). IEEE.

Shimosaka, M., Kaneko, T., & Nishi, K. (2014). Modeling risk anticipation and defensive driving on residential roads with inverse reinforcement learning. In *2014 IEEE 17th international conference on intelligent transportation systems (ITSC)* (pp. 1694–1700). IEEE.

Trautman, P. (2013). *Robot navigation in dense crowds: Statistical models and experimental studies of human robot cooperation*. Pasadena: California Institute of Technology.

Trautman, P., & Krause, A. (2010). Unfreezing the robot: Navigation in dense, interacting crowds. In *2010 IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 797–803).

Trautman, P., Ma, J., Murray, R. M., & Krause, A. (2013). Robot navigation in dense human crowds: the case for cooperation. In *2013 IEEE international conference on robotics and automation (ICRA)* (pp. 2153–2160). IEEE.

Urmson, C., Anhalt, J., Bagnell, D., Baker, C., Bittner, R., Clark, M., et al. (2008). Autonomous driving in urban environments: Boss and the urban challenge. *Journal of Field Robotics*, *25*(8), 425–466.

Vanchinathan, H. P., Nikolic, I., Bona, F. De., & Krause, A. (2014). Explore-exploit in top-n recommender systems via Gaussian processes. In *Proceedings of the 8th ACM conference on recommender systems* (pp. 225–232). ACM.

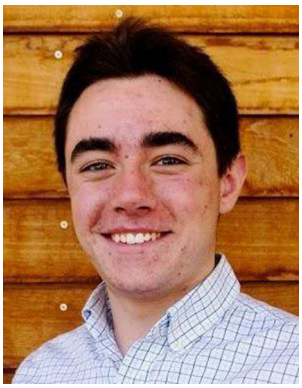
Vasudevan, R., Shia, V., Gao, Y., Cervera-Navarro, R., Bajcsy, R., & Borrelli, F. (2012). Safe semi-autonomous control with enhanced driver modeling. In *American control conference (ACC)* (pp. 2896–2903). IEEE.

Vitus, M. P. & Tomlin, C. J. (2013). A probabilistic approach to planning and control in autonomous urban driving. In *2013 IEEE 52nd annual conference on decision and control (CDC)* (pp. 2459–2464).

Ziebart, B. D. (2010). Modeling purposeful adaptive behavior with the principle of maximum causal entropy.

Ziebart, B. D., Maas, A. L., Bagnell, J. A., & Dey, A. K. (2008) Maximum entropy inverse reinforcement learning. In *AAAI* (pp. 1433–1438).

Dorsa Sadigh is an assistant professor in Computer Science and Electrical Engineering at Stanford University. Her research interests lie in the intersection of robotics, control theory, formal methods, and human-robot interaction. Specifically, she works on developing efficient algorithms for safe and interactive human-robot systems such as semiautonomous driving. She has received her doctoral degree in Electrical Engineering and Computer Sciences (EECS) at UC Berkeley in 2017, and has received her bachelor's degree in EECS at UC Berkeley in 2012. She is awarded the NSF and NDSEG graduate research fellowships as well as Leon O. Chua departmental award, Arthur M. Hopkin departmental award, and the Google Anita Borg Scholarship.



Nick Landolfi is an undergraduate studying Electrical Engineering & Computer Science and Statistics at UC Berkeley. He works with Prof. Anca Dragan in the InterACT Lab, where his research focuses on interactive autonomy in robotics. He is interested in modeling and planning in multi-agent scenarios, incorporating aspects of machine learning, stochastic control and optimization. He was awarded the Regents' and Chancellor's Scholarship and Cal Leadership Award in 2014.

Shankar S. Sastry received his Ph.D. degree in 1981 from the University of California, Berkeley. He was on the faculty of MIT as Assistant Professor from 1980 to 82 and Harvard University as a chaired Gordon Mc Kay professor in 1994. He is currently the Dean of Engineering at University of California, Berkeley. His areas of personal research are embedded control especially for wireless systems, cybersecurity for embedded systems, critical infrastructure protection, autonomous software for unmanned systems (especially aerial vehicles), computer vision, nonlinear and adaptive control, control of hybrid and embedded systems, and network embedded systems and software. He has supervised over 60 doctoral students and over 50 M.S. students to completion. His students now occupy leadership roles in several places and on the faculties of many major universities. He has coauthored over 450 technical papers and 9 books. Dr. Sastry served on the editorial board of numerous journals, and is currently an Associate Editor of the IEEE Proceedings.

Sanjit A. Seshia received the B.Tech. degree in Computer Science and Engineering from the Indian Institute of Technology, Bombay in 1998, and the M.S. and Ph.D. degrees in Computer Science from Carnegie Mellon University in 2000 and 2005 respectively. He is currently a Professor in the Department of Electrical Engineering and Computer Sciences at the University of California, Berkeley. His research interests are in dependable computing and computational logic, with a current focus on applying automated formal methods to embedded and cyber-physical systems, electronic design automation, computer security, and synthetic biology. His awards and honors include a Presidential Early Career Award for Scientists and Engineers (PECASE) from the White House, an Alfred P. Sloan Research Fellowship, the Prof. R. Narasimhan Lecture Award, and the School of Computer Sci-

ence Distinguished Dissertation Award at Carnegie Mellon University. He is a Fellow of the IEEE.

Anca D. Dragan is an Assistant Professor in UC Berkeley's EECS Department. She completed her Ph.D. in Robotics at Carnegie Mellon. She was born in Romania and received her B.Sc. in Computer Science from Jacobs University in Germany in 2009. Her research lies at the intersection of robotics, machine learning, and human–computer interaction: she works on algorithms that enable robots to seamlessly work with, around, and in support of people. Her's research and her outreach activities with children have been recognized by the Intel Fellowship and by scholarships from Siebel, the Dan David Prize, and Google Anita Borg. She has been honored by the Sloan Fellowship, MIT TR35, the Okawa award, and an NSF CAREER award.