Predictive Reachability for Safe Interaction in Social Environments

Luke Shimanuki

Magna Electronics [†], Boston MA 02210, USA
luke@shimanuki.cc

Abstract. Safe control in the presence of other human agents is a challenging but critical requirement for robots in many domains. Reachability-based control schemes provide formally verifiable safety certificates, but force overly-conservative robot behavior in order to avoid all possible behaviors of the other agents. On the other hand, approaches that attempt to predict the other agents' behaviors then plan around them grant the robot greater flexibility, but the safety of such policies is difficult to quantifiably measure outside of real-world testing. We present a framework integrating these two approaches: we leverage reachability analysis to guarantee safety, but when computing the reachable sets we restrict the behaviors of the other agents to a predicted set of policies. These policy sets encode the idea that people's interactions and negotiations are governed by an implicit set of social rules around how regions of space are claimed or yielded. Because the resulting controller is safe as long as the other agents' behaviors are represented in the predictions, its safety can be evaluated (or optimized in a learning setting) *offline* on a dataset of observed trajectories. We implement and evaluate a simple prediction model under this framework in an example driving domain.

Keywords: human-robot interaction, planning under uncertainty, decision and game theory, collision avoidance

1 Introduction

The goal of safe planning and control is generally to reach some desired state without colliding with any obstacles. Unfortunately, this gets complicated when the obstacles can move, especially if such movements are based on partially observable processes (e.g. the internal state of the mental processes of a human the robot is interacting with).

Reachability analysis has been demonstrated to provide policies that are provably safe regardless of the behavior of the other agents [2, 26, 13, 6, 24]. However, these policies must be overly conservative, and oftentimes the problem is unsatisfiable (thereby losing any safety guarantees), since in crowded environments another agent can nearly always force a collision. In order to behave in a natural way, the robot must be able to reason about what the other agents are likely to actually do, rather than everything they could possibly do.

[†] Work performed while at MIT. The claims in this document represent the opinions of the author only and are not endorsed by Magna Electronics.

The standard approach is to model how the other agents move, predict where they're going to go, and then use a time-dependent motion planner to avoid where they are expected to be. Unfortunately, there has been no demonstrated planning algorithm operating on these predictions that accounts for the distributional differences between the predictions and the real-world to provide sufficient safety guarantees for these methods without extensive real-world testing. This is because offline evaluation based on recorded behaviors does not capture how other agents react to the robot's actions, and simulators do not necessarily accurately reflect real-world human behavior.

We propose a different paradigm which can be evaluated offline and still provide safety guarantees. There are two problems that make predicting trajectories ill-formed: unobservability and interactivity. To handle interactivity, we will predict *policies* rather than fixed trajectories. To handle unobservability, we will predict *sets* of policies rather than single policies — these sets will implicitly represent a generalization of ϵ-shadows [4, 20] (also known as sigma hulls [22] or risk contours [3]) to obstacle policies instead of stationary obstacles. We know that the correct joint policy over all agents will avoid collisions, so one evaluation metric would be the number of situations in an observed dataset where a collision would be possible given that every agent follows a policy from the predicted set. And since we need the predictions to actually match reality, the other evaluation metric would be the number of situations in the dataset where an agent's actually observed behavior deviates from the policy set.

To illustrate with some trivial examples, consider the policy set where every agent is stationary. Then the collision rate would be zero, since stationary agents do not collide. But the deviation would be high, because nearly all the time, there will be a non-stationary agent in the data. In the other extreme, consider the policy set where every agent is allowed any possible movement. Then the collision rate would be high, because a collision would always be achievable by some assignment of policies. But the deviation would be low, because anything that was actually observed fits within the policy set. An ideal policy set is then one that is narrow enough to prevent collisions but broad enough to contain every actual observation. In Section 3 we prove that a robot following a policy from such a set will collide at most as frequently as the policy set either allows a collision or fails to capture an actual human behavior.

To motivate this approach, we construct a simple policy set in a manner similar to reachability analysis that can be proven to never allow collisions. A controller can then ensure safety by running a conventional controller optimizing some objective, checking whether the chosen action deviates from the policy set, and if it does deviate, falling back to a safe default policy. The default policy is guaranteed to be safe because the policy set is constructed to ensure that any action can be followed by the default policy without colliding (except for at initialization, when instead the controller will need to exhaustively enumerate or sample candidate actions to ensure that a safe action is found if one exists). The controller will only need to resort to the fallback default policy as often as the policy set fails to contain the desired non-conservative action, so a low empirical deviation from observed behaviors would demonstrate that this policy set allows for non-conservative robot behavior. We emphasize that the probability of deviance for such a policy set is easily evaluated offline, and translates to online safety guarantees as long as the recorded dataset is sampled from the same distribution.

We implement this method for an autonomous driving domain and construct a simple example policy set for which we demonstrate low deviance over the inD dataset [8]. The inD dataset [8] was chosen due to its availability, size, complexity of traffic (large unsignalized intersections), and precision of ground truth labels.

2 Background

2.1 Motion Forecasting

A key prerequisite for safe planning is having some notion of the other agents' future behavior. By far the most common form this takes is predicting trajectories or distributions over trajectories for each agent, which are then evaluated by measuring how closely the predicted trajectories match those observed in real-world recordings using some distance metric. There is a wide range of research in this area, but most recent state-of-the-art results employ large neural networks [27, 19], many explicitly modeling intermediate variables such as intent [11, 41] and interactions [41, 12]. These are trained via standard learning algorithms from recorded trajectories. However, as the prediction horizon increases, the error increases dramatically. This is unavoidable, as it reflects the actual variance in the real world, since the effect of multimodal outcomes becomes more prominent at longer horizons. Hence, these tools often only provide sufficient safety guarantees at low horizons (e.g. often evaluated up to only 2-3 seconds), before the multimodal complexities take effect.

Our work gives up on attempting to predict specific trajectories and instead predicts sets of policies, which can capture this variance, as well as the interaction between different agents. This allows for results that scale well to much longer horizons.

2.2 Planning under Uncertainty

Approaches for planning in the presence of other agents typically falls into one of three camps. The first characterizes the domain as a single agent avoiding collision with other obstacles with motion governed by some partially observable process. A common formalism for describing such a domain is a partially observable Markov decision process (POMDP). There is a wide body of research on solving these problems to varying degrees of generality [7, 9, 30, 10, 29, 16, 39]. Unfortunately, these approaches tend to scale poorly with dimension or complexity of the interactions between agents, as the problem is in general PSPACE-complete.

The second treats the problem as a multi-agent game theory problem, taking the assumption that each agent is maximizing some known utility function and then either estimating the optimal or equilibrium joint policy [32, 15, 5, 23] or ordering the agents using heuristics and then allowing each to select its optimal move given the moves of the prior agents [33, 25].

The third and final category is learning from real-world recordings. Typically these approaches do not explicitly model the unobservable structure of the problem, instead relying on the learning process to infer these kinds of reasoning from the data. The different methods for this kind of approach are highly diverse and take direction from many different subfields of machine learning, but some common patterns include inverse reinforcement learning [18, 1, 42] and end-to-end planning or imitation learning methods [38, 37, 40, 31].

Our work draws from all three of these paradigms, allowing each agent to make its decisions by using heuristics on a highly structured POMDP, inferring common motion patterns from real-world data, and then providing safety guarantees based on a game theoretic analysis.

2.3 Formally Verified Controllers

Analytical proofs of safety following synthesis [28, 21] or formal verification [35, 14] techniques have met with some success, but unfortunately they require known models of the world, and so do not scale up to more complex systems involving interactions between agents. In the absence of knowing exactly how the world will evolve, we are left with considering all possible ways it could potentially evolve. Reachability-based approaches, including open-loop planning around possible future states of other agents [2, 26] and closed-loop control that always ensures that there exists a policy that will avoid collision regardless of the behavior of other agents [13, 6, 24] do provide safety guarantees in these domains, so long as a plan is found. However, these result in overly-conservative behavior in crowded settings, since the other agents can nearly always force a collision regardless of evasive robot behavior, and in particular, they can provide no guarantees that a plan will be found or that the robot can act safely otherwise.

Our work follows a similar path as the closed-loop control approaches, but it instead restricts the behavior it expects from the other agents to allow for less conservative robot behavior. This is safe under the assumption that there exists some set of social rules limiting human behavior, and the validity of such a set of rules can be empirically verified offline. However, this does come at the cost of not guaranteeing safety in the rare instance of another agent intentionally trying to force a collision.

2.4 Evaluating Interactive Systems

It remains an open question how best to evaluate an interactive system to provide formal safety guarantees, as it has proven to be an extremely difficult objective to measure in many cases [34].

The alternative to formal safety certificates is empirically evaluating safety on a large number of potential situations sampled from the real-world distribution. There are three common methods of generating the motions of other agents for such an evaluation. The first, and most ideal, is real-world testing [36]. With sufficient time in the wild, this provides a good picture of how safe the planning system is. However, it is slow and costly, in terms of equipment, manpower, and risk of something going wrong. Evaluating via testing in the wild is not an option for most labs and even many companies researching domains of this kind, and it is especially difficult for any learning system to directly optimize over this metric in high-risk domains. If we can't test in the real world, another option is simulation [7, 16, 39]. However, this is inherently circular because the simulated motions of other agents must be derived from some computational process that does not necessarily reflect reality. A planner that achieves safety in this simulated domain has only demonstrated that it is able to predict and respond to the process governing the simulation, rather than real-world behaviors, and so we must then verify the accuracy of the simulator itself, which brings us back to the original problem.

The last option is to record real human behavior, but then allow the planner to change what the robot does offline [38, 17]. While this seems to be the best option available to the average researcher, it is limited by its inability to capture how other agents will respond to different robot actions. For autonomous driving, which is an example of an interactive domain, Houston et al. [17] find that 85% of situations requiring intervention (e.g. collision) under this evaluation paradigm are at least in-part due to the non-reactivity of the other agent motions. It is impossible to compute in how many of those a collision would have actually occured without being able to predict the agent motion in the counterfactual case, that is, what the other agent would have done had the robot done something else. Hence, up until now to the best of our knowledge, real-world testing has been the only way to provide statistical bounds for the performance of an interactive system, with the offline options often used in its place for cost reasons, but failing to provide the same guarantees.

Our work provides an analytical proof of safety for a particular model of the world which can then be compared against real-world data offline to provide quantitative safety guarantees for the domain in which the data was generated.

3 Problem Definition

3.1 Overview

As with most decision making problems, this class of domains can be described as a partially observable markov decision process (POMDP). For the purpose of this paper, we assume that all truly random processes (e.g. quantum effects) are insignificant, leaving us with a deterministic POMDP, i.e. one where the transition model is deterministic and hence all perceived randomness stems from the uncertainty over the unobservable state. Let O denote the observable state space for a single agent, including the pose, momentum, shape, and anything else that defines how it can move and interact with other agents (in particular, this assumes that the perception system can perfectly estimate these attributes of all other agents), and let Π denote the unobservable state space including information such as where it wants to go or how it plans to get there. Let n be the number of agents ($N = \{1..n\}$ being the set of agents) and τ be the number of observed timesteps. Suppose that each agent's behavior is fully described by a fixed policy $\pi : H \to A$, where H is the joint observable state space history $O^{n\tau}$, and A is the action space for a single agent, typically actuator settings for the duration of a timestep. Then the unobservable state space of an agent captures some space of such policies, that is $\Pi \subseteq H \to A$. In order to retain the Markov property, we restrict τ to be upper bounded by some constant (the examples in this paper require just 3), and so the full state space of the POMDP is $O^{n\tau} \times \Pi^{n-1}$ (the ego policy is not part of the state). Let us further suppose that the transition model for a single agent $M : O \times A \to O$ is fully known and independent of the state and actions of other agents, and that there is a unique action that leads from any given state o_1 to o_2. For situations where it is not unambiguous, we specify $h^{(i)}$ as the history h but with agent i treated as the ego-agent, that is, the one that is being controlled.

Then the problem we would like to solve can be informally described as

Problem 1 (Interaction Problem). Given the current history $h \in H$ and a goal state $g \in O$ (describing just the robot), find a policy $\pi \in \Pi$ minimizing the probability that the goal is not reached or the robot collides with another agent.

Under this problem definition, the evolution of the state is fully determined by the initial observable state, the policy chosen for the ego-agent, and the unobservable policies of all of the other agents. Then if the policies of all the other agents were known, it is trivial to determine whether a candidate policy would succeed (i.e. reach the goal without colliding with another agent), and hence this is equivalent to a traditional motion planning problem, albeit one with an exceedingly large state space since each action the ego-agent takes affects the trajectories of the other agents. Given that the policies of the other agents is generally not known though, then if we had even just a distribution over them that we could sample from, then we could still estimate the probability that a candidate policy reaches the goal or collides with another agent by sampling policies for the other agents and counting how frequently the candidate policy is successful. However, the problem that we would like to address in this paper is the one where not even the distribution over policies is known. Since the policies of the other agents are usually generated by a complex and opaque process (i.e. their brain) that is just as unobservable to us as the engineers as it is to the ego-agent we are trying to program, we are not able to observe even samples from this distribution. What we can do is select a policy for the ego-agent, put it out in the real world (thereby sampling a state from the distribution, even if we can't observe it), and then observing the *actions* each other agent's policy chooses in that specific state. Unfortunately, doing so is rather slow, expensive, and likely to lead to human injury or property damage unless the selected policy is actually safe, and so online methods (for either evaluation or direct learning) are not viable under most engineering teams' resource constraints. This conundrum has led towards attempts to learn approximate models of this distribution as best we can using imitation learning or similar offline methods on recorded trajectories [27, 19, 11, 41, 12], and then the learned distribution could be used in traditional planning or POMDP algorithms. However, in many domains, it has proved difficult to learn a sufficiently close approximation of the distribution, and even moreso to quantify how close the learned distribution is to the true distribution, which is required in order to provide any sort of probabilistic guarantees. We are optimistic that the field will continue to progress along this line of research, but in the meantime we propose an orthogonal direction that to the best of our knowledge has not yet been explored as thoroughly. That is, rather than learning the state distribution then checking whether a given policy is safe with respect to it, instead construct a constraint on the ego and other agent policies that guarantees collision avoidance (in the vein of multi-agent cooperative systems), and then check the frequency that an agent violates that constraint (and in the future, optimize the constraint to minimize the frequency of violation while still disallowing collision).

3.2 Policy Sets

In uncertain but unreactive motion planning domains where exact obstacle distributions are either unknown or difficult to reason about, one approach that has been proposed is to instead use ϵ-shadows [4, 20], which are task-space volumes that contain the obstacle with at least a certain probability. We extend that concept to the kind of interactive domains we've been discussing here, resulting in the following definition.

Definition 1 (ϵ-policy-shadow). *An ϵ-policy-shadow for agent i is a mapping from an observed history $h^{(i)} \in H$ to a region in task space at the next timestep that fully contains agent i with probability at least $1 - \epsilon$. It is equivalent to the union of volumes agent i could occupy at the next timestep if it were to follow a policy within a policy set $P \subset \Pi$ where the probability that agent i is following a policy not in P is at most ϵ.*

Using this idea, we can reframe our objective using the following definitions.

Definition 2 (deviant). *A set of policies $P \subset \Pi$ is deviant from a history $h \in H$ over a given time period $[\tau_1, \tau_2]$ if for any timestep $\tau \in [\tau_1, \tau_2]$, any agent i took an action $a \in A$ during that time period for which there does not exist a policy $\pi \in P$ such that $\pi(h_\tau^{(i)}) = a$, where $h_\tau^{(i)}$ denotes the truncated history up to timestep τ.*

Definition 3 (collidable). *A set of policies $P \subset \Pi$ is collidable over a history $h^{(i)} \in H$ and a given time period $[\tau_1, \tau_2]$ if there exists an assignment of policies from P to each agent such that simulating forward starting at $h_{\tau_1}^{(i)}$ for $\tau_2 - \tau_1$ timesteps results in agent i colliding with some other agent.*

Problem 2 (Offline Interaction Problem). Given a dataset of histories sampled from the distribution of interest, find a policy set $P \subset \Pi$ minimizing the number of histories over which P is deviant or collidable. Then select a policy $\pi \in P$ such that a particular goal state $g \in O$ is reached starting from a particular history $h \in H$.

This may seem like a strange objective, but it turns out that solving Problem 2 also solves Problem 1. Consider the 2 metrics in Problem 1. First, we observe that if the goal is something an agent in the dataset would have had as a goal, then that agent would likely be recorded as reaching the goal, and so either P contains a policy that leads to the goal or it is deviant. Second, we upper bound the probability that a collision is possible by following a policy in P.

Theorem 1. *The total risk of collision R of following a policy in $P \subset \Pi$ is upper bounded by $C + D$, where C is the probability that P is collidable, and D is the probability that P is deviant.*

Proof. Consider a situation for which following a policy in P leads to collision. If every other agent is also following a policy in P, then by definition this is a situation where P is collidable. If some agent is not following a policy in P, then by definition this is a situation where P is deviant. Then the probability of collision is upper bounded by the sum of the probability that P is collidable and the probability that P is deviant, hence $R \leq C + D$.

Short aside: we can think of the probability that P is not deviant as the recall (since it measures how many actual behaviors are represented), and the probability that P is not collidable as the precision (since they are measuring how well the model excludes bad outputs). Therefore we could consider measuring a variant of the F_1 score. However, the harmonic mean in this case doesn't mean much intuitively, whereas the sum upper bounds the risk of collision, so the sum turns out to be a nicer metric to optimize for.

Representing sets of policies is fairly difficult, and moreso checking whether any contained policy allows collision, especially since the set is usually infinitely large. While under certain assumptions, this can be approximated by sampling a large number of policies and verifying that none of them allow collision, this is not foolproof, and it is trivial to construct policy sets for which policies that lead to collision always exist but are rarely sampled (e.g. suppose there are 10 agents and each is allowed to teleport up to 1000 meters — this policy set is always collidable, but the odds that a pose sampled for one agent would be in collision with that for another agent is small because of the large sample space). Instead, we will construct a policy set that by definition is never collidable. Then, this policy set will be evaluated solely based on its deviance.

Problem 3 (Non-collidable Interaction Problem). Given a dataset of histories sampled from the distribution of interest, find a policy set $P \subset \mathit{\Pi}$ that is never collidable and minimizing the number of histories over which P is deviant. Then select a policy $\pi \in P$ such that a particular goal state $g \in O$ is reached starting from a particular history $h \in H$.

4 Collision-free Policy Set

We now construct a set, first in very general terms that can be applied to any interaction domain with any set of priors, then a specific example for a 2D navigation domain.

4.1 A Parameterized Policy Set

We define a space P_{ϕ,π_0} of policy sets parameterized by 2 parameters. The first parameter is a claiming map $\phi : H \times N \times T \times \mathbb{R}^d \to \mathbb{R}$ indicating the degree to which a given agent claims a given point in task space $\mathbb{R}^d$ for a particular timestep (T is the set of timesteps). If $\phi(h, i, \tau, x) > \phi(h, j, \tau, x)$ for each other agent j, then agent i is considered to claim x at time τ (and hence no other agent should occupy that space-time). From this scoring function we can derive a partitioning of $T \times \mathbb{R}^d$ into disjoint sets, denoted $\psi(h^{(i)})$ for each agent i. An example of a partitioning resulting from a specific claiming map is illustrated in Figures 1 and 2, although we note that the results in this section equally apply to any claiming map. The second parameter is a default policy $\pi_0 \in \mathit{\Pi}$. Each agent will ensure that it is always able to safely begin following π_0 and that each other agent can safely begin following π_0 at any timestep. Furtheremore, every agent following π_0 must lead to a steady state within a known time horizon, after which this steady state must be known to be safe for eternity (e.g. if the steady state is stationary, every other agents' default trajectory must not collide with it). The π_0 must also not depend on the state of any other agent, and hence can be evaluated to generate a trajectory prior to knowing the actions of the other agents.

The resulting policy set is then simple to construct. $\pi \in P_{\phi,\pi_0}$ iff the following conditions hold, starting from some history h_τ:

- Simulating forward the robot agent i following π starting at state $h_\tau^{(i)}$ for a single timestep does not collide with any point $x \notin \psi(h_{\tau-2}^{(i)})$.
- Simulating forward the robot agent i following π starting at state $h_\tau^{(i)}$ for a single timestep and then executing π_0 for $\eta - 1$ timesteps does not collide with any point $x \notin \psi(h_{\tau-1}^{(i)})$.
- Simulating forward another agent j following π starting at state $h_\tau^{(j)}$ for a single timestep does not collide with any point $x \in \psi(h_{\tau-2}^{(i)})$.
- Simulating forward another agent j following π starting at state $h_\tau^{(j)}$ for a single timestep and then executing π_0 for $\eta - 1$ timesteps does not collide with any point $x \in \psi(h_{\tau-1}^{(i)})$.

Here we use η to denote the time horizon, which must be at least as long as it takes for the default policy to reach steady state. Informally, this is basically saying that the robot is allowed to move anywhere in space-time that it claimed both 2 timesteps ago and 1 timestep ago (the intersection of these regions), as long as it ensures that it can execute the default policy afterwards without leaving the region it claimed 1 timestep ago, and that every other agent is allowed to do the same but with the complement of the robot's claimed region. The reason why the policy set is assymetric is because we are measuring only how often the robot in question is in collision. If our objective was to minimize all collisions regardless of which agents were involved, then we could restrict every agent to their specific claimed region. We restrict the policies to act based on information at least one timestep old because neither humans nor computers can actually act on new information immediately. The timestep would then be set to the combined input and processing lag of the decision-making system. Examples of policies that are contained and not contained in this set for a particular initial state and claiming map for a one-dimensional space are shown in Figure 3.

Now we show that this policy set does not allow for collisions.

Theorem 2. *P_{ϕ,π_0} is not collidable for any history $h^{(i)}$ and time period $[\tau, \tau+\eta]$.*

Proof. Suppose the robot agent i can execute a policy in P_{ϕ,π_0} for a single timestep starting at τ_1, where $\tau \leq \tau_1 \leq \tau + \eta$. Then each other agent is either also executing such a policy or the default policy; this is guaranteed to not be in collision because agent i avoids each other agent j's $\psi(h_{\tau_1-1}^{(j)})$, including any possible default policy. Suppose agent i cannot execute such a policy. Then it can execute the default policy, which is guaranteed to not collide with such a policy for any other agent because each other agent avoids $\psi(h_{\tau_1-1}^{(i)})$, which contains agent i's default policy. Agent i's default policy starting at τ_1 cannot collide with another agent j's default policy also starting at τ_1 because at time $\tau_1 - 1$ both agents selected trajectories such that an ensuing default policy is contained in $\psi(h_{\tau_1-2}^{(i)})$ and its complement, respectively, which are disjoint. If agent i initiates a default policy at τ_1 and some agent j initiates a default policy before τ_1, then at $\tau_1 - 1$ agent i chose a trajectory that can be followed by a default policy

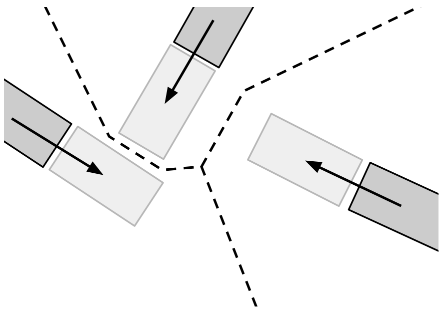

Fig. 1. An example partitioning in a 2D space at a particular time. Each agent's state at time $\tau+\delta$ given the default policy (depicted as a trajectory following the dotted line), along with the corresponding partitioning. Each agent's claimed region is the region of space that is closer to it's footprint at time $\tau+\delta$ than that of any other agent. The states at time τ are in dark gray, the states at time $\tau+\delta$ are in light gray, and the dotted lines divide the space at time $\tau+\delta$ into disjoint regions.

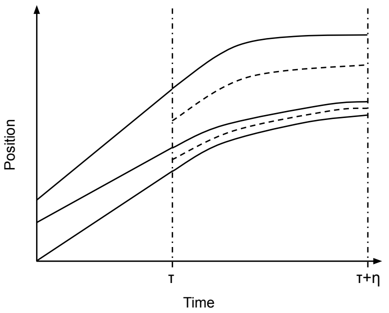

Fig. 2. An example partitioning in a 1D space over time, where each agent is a point robot. The solid curves are the trajectories resulting from the default policy (depicted here as a stopping policy) for each agent. The dashed curves divide the space-time into the regions claimed by each agent.

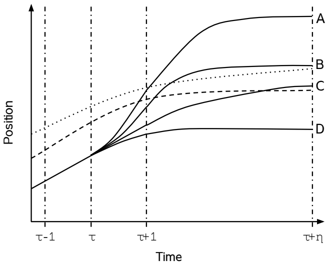

Fig. 3. We are currently trying to decide controller settings for timestep τ based on observations up to time $\tau-1$ in a one-dimensional domain. Suppose the dotted line represents the upper border of $\psi(h^{(i)}_{\tau-1})$ and the dashed line represents the upper border of $\psi(h^{(i)}_{\tau-2})$. Up to time τ is drawn the historical trajectory in addition to the expected trajectory for timestep $\tau-1$, since the action for that timestep was already chosen at the previous timestep. Then for timestep τ there are the trajectories produced by four example policies. Each of these is followed by the default policy (depicted here as a stopping policy) from time $\tau+1$ to $\tau+\eta$. Policy A is not contained in the policy set because it exceeds $\psi(h^{(i)}_{\tau-2})$ during timestep τ. Policy B does not exceed $\psi(h^{(i)}_{\tau-1})$ or $\psi(h^{(i)}_{\tau-2})$ during timestep τ, but the default policy in the following η timesteps does exceed $\psi(h^{(i)}_{\tau-1})$, so Policy B is also not contained in the set. Policy C does not exceed $\psi(h^{(i)}_{\tau-1})$ or $\psi(h^{(i)}_{\tau-2})$ during timestep τ, nor does the subsequent default policy exceed $\psi(h^{(i)}_{\tau-1})$, so Policy C is contained in the set, even though the subsequent default policy does exceed $\psi(h^{(i)}_{\tau-2})$. Policy D does not exceed $\psi(h^{(i)}_{\tau-1})$ or $\psi(h^{(i)}_{\tau-2})$ during timestep τ or in the subsequent default policy, so it is also contained in the set.

at time τ while staying within $\psi(h^{(i)}_{\tau_1-2})$, which cannot intersect with any other agent's default policy starting at $\tau_1 - 1$. If no such trajectory could be found, then agent i would have executed the default policy at $\tau_1 - 1$, and by induction we can see that the only situation where agent i's default policy can collide with any other agent's default policy is if those were the initial conditions, in which case the set of feasible actions will be empty, thereby still not allowing a collision, but instead being deviant in this instance.

Therefore, the overall risk of P_{ϕ,π_0} is its rate of deviation.

4.2 Policy Selection

Now that we have a policy set P_{ϕ,π_0} that is safe whenever it is not deviant, the remaining task is to select a policy $\pi \in P_{\phi,\pi_0}$ for a particular initial state within a scene. As defined above, P_{ϕ,π_0} allows the ego-agent i to take any action that keeps its footprint at the next timestep within $\psi(h^{(i)}_{\tau-2})$ and such that following π_0 until steady state is reached keeps its footprint within $\psi(h^{(i)}_{\tau-1})$. For a given candidate action, it is straightforward to evaluate whether it violates either of these constraints. Then a simple policy could initially enumerate or sample potential actions, ordered or weighted by how well they approach the goal, and then selecting the first action that does not violate the above constraints. Then, for future timesteps, it can generate candidate actions from any other (potentially unsafe) controller or set of controllers, which presumably optimizes for reaching the goal, then pick a candidate action that does not violate the constraints, or the default policy (which is guaranteed to satisfy the constraints if the initial timestep was successful) if none exist.

We note that the probability that P_{ϕ,π_0} is deviant upper bounds the probability that no such policy exists for goals that the agents in the recorded scenes are following. Hence, as long as P_{ϕ,π_0} can be shown to have deviance at most ϵ over the policy distribution of interest, such a policy will be found with probability at least $1 - \epsilon$, and that policy will be safe with probability at least $1 - \epsilon$ for scenes and goals sampled from the same distribution the dataset was from.

5 Example Implementation & Evaluation

We will now provide details for a specific domain that we will demonstrate this approach on.

5.1 Domain

We consider the autonomous navigation problem, in which each agent, including the ego-agent, is a single-link convex rigid body in two or three dimensions, and the goal is for the ego-agent to reach a particular region by a particular time. Each agent's motion is potentially non-holonomic and constrained by standard Newtonian mechanics, with limits on the magnitude of acceleration they can apply.

5.2 Parameter Details

Suppose simulating an agent i forward following π_0 starting at state $h_\tau^{(i)}$ for j timesteps results in agent i being centered at point $y_0 \in \mathbb{R}^d$. Then we define $\phi(h, i, \tau + j, x)$ as the negative distance from x to the footprint of agent i at pose y_0. Example partitionings resulting from this scoring function are depicted in Figures 1 and 2. Informally, this scoring function allows an agent to move anywhere near the trajectory followed by its default policy, so long as that for another agent would not be even closer.

For the default policy π_0, the simplest option is a stopping policy. In particular, we define π_k to be the policy in which the agent will start moving along its current kinematic curve (constant curvature) and slow down with a fixed deceleration until it stops. This policy is illustrated in Figure 4. We note that the behaviors permitted by P_{ϕ,π_k} are not limited to those described by π_k itself. π_k merely provides reference trajectories from which we can infer the degree to which a given agent has a claim on a particular region of space-time. Hence, the resulting policy sets can generalize to a broader set of policies so long as no other agent has a stronger claim on the regions of space-time that would be occupied.

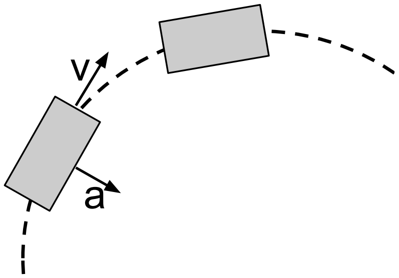

Fig. 4. The default policy π_k. The trajectory maintains a constant longitudinal deceleration and angular velocity, following a circular path based on the centripetal acceleration at the current state.

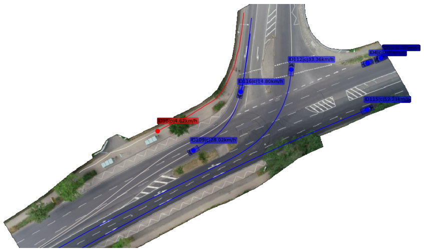

Fig. 5. An example intersection in the inD dataset [8] with recorded agent trajectories. Rendered by the inD Dataset Python Tools.

Note that the results in Section 4.1 hold regardless of our choice of π_0 and ϕ. The end goal would be to choose these parameters to minimize deviation. We expect that explicitly reasoning about dynamics models, environment geometry and semantics (e.g. in traffic domains, lanes would be a significant factor), and interactions between agents would lead to better performance, as would learning the parameters by directly optimizing over a training dataset. Nevertheless, even with our overly simplistic model, we are still able to achieve relatively strong safety guarantees.

5.3 Empirical Results

We evaluate our method on the inD dataset [8], which contains recorded trajectories for traffic participants at various intersections. We compute the deviance over every vehicle (cars and trucks, ignoring bicycles and pedestrians) that moves at least 5 meters (mostly ignoring parked vehicles), and we remove segments within 5 seconds of collisions in the ground truth, assuming that those were due to slight labeling error, as well as at the end of trajectories so we are not predicting beyond what has been recorded. The horizon η was set to 10 seconds, leaving sufficient time for any of the agents to come to a stop, and the timestep was set to 80ms, a reasonable amount of time for a computer to react (2 frames of the dataset, which is recorded at 25Hz). An example intersection present in the dataset is depicted in Figure 5.

Starting from a uniform prior, we found the posterior belief for the deviance of P_{ϕ,π_k} over 10-second long episodes to be $D_{\phi,\pi_k} \sim \text{Beta}(128, 10462)$, with a MAP of 1.20%, and 99.7% confidence interval $D_{\phi,\pi_k} \leq 1.52\%$, implying with high confidence that this controller will fail or act unsafely on average at most every 657.9 seconds (~ 11 minutes) when sampling scenarios from the distribution that generated the dataset.

6 Conclusion and Future Work

We have presented a compromise between the absolute but limited safety provided by reachability approaches and the non-conservative behavior allowed by systems relying on predictions.

Specifically in the context of our example autonomous driving domain, we have shown that even a rudimentary model with few priors around motion models and environmental semantics can achieve a fairly low risk of collision or failure. While the performance is still far worse than human-level and not remotely deployable in production environments, we believe that it will not be difficult to design more nuanced models in the future that are able to achieve very low deviance with high confidence. These extensions could include, but are not limited to, replacing our model's heuristics with parameterized functions that can then be learned to minimize deviance. For example, π_k could naturally be replaced with any trajectory-generating model (e.g. those in [18, 1, 42, 38, 37, 40, 31]) and then fine-tuned by optimizing over the resulting deviance. Alternatively, or in addition, ϕ could be replaced with a cost-map-generating model (e.g. that listed in [38]) and similarly fine-tuned to minimize the resulting deviance. Doing so would leverage the expressivity and performance of these existing approaches while still producing probabilistically-sound safety guarantees. Unfortunately, the deviance calculation is not differentiable, and so it may be more prudent to select simpler models with lower dimensionality unless an alternative efficient learning algorithm is found.

More broadly, our core contribution is a domain-agnostic approach for interacting with social agents, and in particular, one whose safety can be evaluated and improved upon offline. It also opens the door for various new directions for future work along this paradigm. We have presented only one particular (albeit parameterized) non-collidable policy set, and so we expect that an alternative non-collidable policy set based on a different framing of the negotiation problem might lead to better results for certain classes of domains. As for the specific one presented here, there is room for further development of the parameters ϕ and π_0, although we expect that selecting these will tend to be domain-dependent, such as with the extensions proposed above. Because the deviance can be evaluated offline, the parameters can be learned and evaluated based on domain-specific data. On the controller side, since the policy set itself acts more as a constraint rather than as an actual controller, designing controllers that explicitly optimize some objective subject to such a constraint could lead to better performance in risky situations, rather than always falling back to the default policy. Finally, our approach could be extended to integrate perception or dynamics uncertainty to better handle real-world environments.

References

[1] Pieter Abbeel, Dmitri Dolgov, Andrew Ng, and Sebastian Thrun. Apprenticeship learning for motion planning with application to parking lot navigation. pages 1083–1090, 09 2008. doi: 10.1109/IROS.2008.4651222.

[2] Matthias Althoff and John Dolan. Online verification of automated road vehicles using reachability analysis. *Robotics, IEEE Transactions on*, 30:903–918, 08 2014. doi: 10.1109/TRO.2014.2312453.

[3] Brian Williams Ashkan Jasour. Risk contours map for risk bounded motion planning under perception uncertainties. In *Robotics: Science and Systems*, 2019.

[4] Brian Axelrod, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. Provably safe robot navigation with obstacle uncertainty. *The International Journal of Robotics Research*, 2018.

[5] Mohammad Bahram, Andreas Lawitzky, Jasper Friedrichs, Michael Aeberhard, and Dirk Wollherr. A game-theoretic approach to replanning-aware interactive scene prediction and planning. *IEEE Transactions on Vehicular Technology*, 65: 1–1, 01 2015. doi: 10.1109/TVT.2015.2508009.

[6] Andrea Bajcsy, Somil Bansal, Eli Bronstein, Varun Tolani, and Claire Tomlin. An efficient reachability-based framework for provably safe autonomous navigation in unknown environments. pages 1758–1765, 12 2019.

[7] Tirthankar Bandyopadhyay, Kok Sung Won, Emilio Frazzoli, David Hsu, Wee Sun Lee, and Daniela Rus. Intention-aware motion planning. In *Workshop on the Algorithmic Foundations of Robotics*, 2012.

[8] Julian Bock, Robert Krajewski, Tobias Moers, Steffen Runde, Lennart Vater, and Lutz Eckstein. The ind dataset: A drone dataset of naturalistic road user trajectories at german intersections. 2019.

[9] Sebastian Brechtel, Tobias Gindele, and Rüdiger Dillmann. Probabilistic decision-making under uncertainty for autonomous driving using continuous pomdps. 10 2014. doi: 10.1109/ITSC.2014.6957722.

[10] Adam Bry and Nicholas Roy. Rapidly-exploring random belief trees for motion planning under uncertainty. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 723–730. IEEE, 2011.

[11] Sergio Casas, Wenjie Luo, and Raquel Urtasun. IntentNet: Learning to Predict Intention from Raw Sensor Data. In Aude Billard, Anca Dragan, Jan Peters, and

Jun Morimoto, editors, *Proceedings of The 2nd Conference on Robot Learning*, volume 87 of *Proceedings of Machine Learning Research*, pages 947–956. PMLR, 29–31 Oct 2018.

[12] Sergio Casas, Cole Gulino, Simon Suo, Katie Luo, Renjie Liao, and Raquel Urtasun. Implicit latent variable model for scene-consistent motion forecasting, 2020.

[13] Mo Chen and Claire Tomlin. Hamilton–jacobi reachability: Some recent theoretical advances and applications in unmanned airspace management. *Annual Review of Control, Robotics, and Autonomous Systems*, 1:333–358, 05 2018.

[14] Jonathan DeCastro, Javier Alonso-Mora, Lucas Liebenwein, Wilko Schwarting, Cristian-Ioan Vasile, Sertac Karaman, and Daniela Rus. Compositional and contract-based verification for autonomous driving on road networks. 12 2017.

[15] Michael During and Patrick Pascheka. Cooperative decentralized decision making for conflict resolution among autonomous agents. pages 154–161, 06 2014.

[16] Jason Hardy and Mark Campbell. Contingency planning over probabilistic hybrid obstacle predictions for autonomous road vehicles. In *IEEE International Conference on Intelligent Robots and Systems*, 2010.

[17] John Houston, Guido Zuidhof, Luca Bergamini, Yawei Ye, Long Chen, Ashesh Jain, Sammy Omari, Vladimir Iglovikov, and Peter Ondruska. One thousand and one hours: Self-driving motion prediction dataset. In *Conference on Robot Learning*, 2020.

[18] Sandy Huang, David Held, Pieter Abbeel, and Anca Dragan. Enabling Robots to Communicate Their Objectives. *Robotics: Science and Systems XIII*, Jul 2017.

[19] Michael Janner, Igor Mordatch, and Sergey Levine. γ-models: Generative temporal difference learning for infinite-horizon prediction. In *Advances in Neural Information Processing Systems*, 2020.

[20] Leslie Pack Kaelbling and Tomás Lozano-Pérez. Integrated task and motion planning in belief space. *The International Journal of Robotics Research*, 2013.

[21] Eric Kim, Murat Arcak, and Sanjit Seshia. Compositional controller synthesis for vehicular traffic networks. pages 6165–6171, 12 2015.

[22] Alex Lee, Yan Duan, Sachin Patil, John Schulman, Zoe McCarthy, Jur van den Berg, Ken Goldberg, and Pieter Abbeel. Sigma hulls for Gaussian belief space planning for imprecise articulated robots amid obstacles. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5660–5667, 2013.

[23] David Lenz, Tobias Kessler, and Alois Knoll. Tactical cooperative planning for autonomous highway driving using monte-carlo tree search. pages 447–453, 06 2016. doi: 10.1109/IVS.2016.7535424.

[24] Karen Leung, Edward Schmerling, Mengxuan Zhang, Mo Chen, John Talbot, J Gerdes, and Marco Pavone. On infusing reachability-based safety assurance within planning frameworks for human–robot vehicle interactions. *The International Journal of Robotics Research*, 39:027836492095079, 08 2020.

[25] N. Li, D. W. Oyler, M. Zhang, Y. Yildiz, I. Kolmanovsky, and A. R. Girard. Game theoretic modeling of driver and vehicle interactions for verification and validation of autonomous vehicle control systems. *IEEE Transactions on Control Systems Technology*, 26(5):1782–1797, 2018. doi: 10.1109/TCST.2017.2723574.

[26] Stefan Liu, Hendrik Roehm, Christian Heinzemann, Ingo Lütkebohle, Jens Oehlerking, and Matthias Althoff. Provably safe motion of mobile robots in human environments. 09 2017. doi: 10.1109/IROS.2017.8202313.

[27] W. Luo, B. Yang, and R. Urtasun. Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3569–3577, 2018. doi: 10.1109/CVPR.2018.00376.

[28] Petter Nilsson, Omar Hussien, Ayca Balkan, Yuxiao Chen, Aaron Ames, Jessy Grizzle, Necmiye Ozay, Huei Peng, and Paulo Tabuada. Correct-by-construction adaptive cruise control: Two approaches. *IEEE Transactions on Control Systems Technology*, 24:1–14, 12 2015. doi: 10.1109/TCST.2015.2501351.

[29] R. Platt, R. Tedrake, L. Kaelbling, and T. Lozano-Perez. Belief space planning assuming maximum likelihood observations. In *Proceedings of Robotics: Science and Systems*, Zaragoza, Spain, June 2010. doi: 10.15607/RSS.2010.VI.037.

[30] Sam Prentice and Nicholas Roy. The belief roadmap: Efficient planning in linear pomdps by factoring the covariance. In *Proceedings of the 13th International Symposium of Robotics Research (ISRR)*, Hiroshima, Japan, November 2007.

[31] Stephane Ross, Geoffrey Gordon, and J. Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. *Journal of Machine Learning Research - Proceedings Track*, 15, 11 2010.

[32] Dorsa Sadigh, Shankar Sastry, Sanjit Seshia, and Anca Dragan. Planning for autonomous cars that leverage effects on human actions. 06 2016. doi: 10.15607/RSS.2016.XII.029.

[33] Wilko Schwarting and Patrick Pascheka. Recursive conflict resolution for cooperative motion planning in dynamic highway traffic. pages 1039–1044, 10 2014.

[34] Wilko Schwarting, Javier Alonso-Mora, and Daniela Rus. Planning and Decision-Making for Autonomous Vehicles. *Annual Review of Control, Robotics, and Autonomous Systems*, 1(1):187–210, 2018. doi: 10.1146/annurev-control-060117-105157.

[35] Bastian Schürmann, Daniel Heß, Jan Eilbrecht, Olaf Stursberg, Frank Köster, and Matthias Althoff. Ensuring drivability of planned motions using formal methods. 10 2018.

[36] Waymo. Waymo Safety Report. 2020.

[37] H. Xu, Y. Gao, F. Yu, and T. Darrell. End-to-end learning of driving models from large-scale video datasets. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3530–3538, 2017.

[38] Wenyuan Zeng, Wenjie Luo, Simon Suo, Abbas Sadat, Bin Yang, Sergio Casas, and Raquel Urtasun. End-to-end interpretable neural motion planner. pages 8652–8661, 06 2019. doi: 10.1109/CVPR.2019.00886.

[39] Wei Zhan, Changliu Liu, Ching-Yao Chan, and Masayoshi Tomizuka. A non-conservatively defensive strategy for urban autonomous driving. In *IEEE 19th International Conference on Intelligent Transportation Systems*, 2016.

[40] Jiakai Zhang and Kyunghyun Cho. Query-efficient imitation learning for end-to-end autonomous driving. 05 2016.

[41] Hang Zhao, Jiyang Gao, Tian Lan, Chen Sun, Benjamin Sapp, Balakrishnan Varadarajan, Yue Shen, Yi Shen, Yuning Chai, Cordelia Schmid, Congcong Li, and Dragomir Anguelov. Tnt: Target-driven trajectory prediction, 2020.

[42] Brian D. Ziebart, Andrew Maas, J. Andrew Bagnell, and Anind K. Dey. Maximum entropy inverse reinforcement learning. In *Proc. AAAI*, pages 1433–1438, 2008.