Cautious Markov Games for Interaction Aware Robotics

Rohan Sinha[1] and Sanjay Lall[2]

***Abstract*— Autonomous vehicles (AVs) and other agents typically interact in structured environments with rules and conventions that all agents *should* follow, but do not always do, such as in traffic. Therefore, we study 1) how to incorporate these rules and conventions into multi-agent robotics problems and 2) what the implications are for interaction-aware methods for decision making. To do this, we express the rules as linear temporal logical constraints on the joint state trajectory and model the multi-agent interaction as a stochastic game. We learn the likelihood of other agents making decisions that can violate the rules as chance constraints to interpretably represent the tendency of agents to break the rules, rather than implicitly encode it in the agents' preference structure. We dub this framework the *cautious Markov game* (CMG), for which we efficiently construct policies using robust dynamic programming. We find that we can significantly reduce the conservatism of robust policies by exploiting the rule-based nature of the game on illustrative examples, thereby confirming our intuition that traffic rules significantly reduce the need for inter-agent negotiation.**

I. Introduction

To reliably deploy the next generation of autonomous robotic systems in unstructured open-world environments, they have to interact safely with other agents such as humans and other robots. In many of these settings, such as when an autonomous vehicle merges into traffic on a highway on-ramp, automated systems need to negotiate with other agents in their environment to achieve an objective that is in tension with the goals of others. Therefore, as surveyed in [1], a rapidly growing body of research in the controls and AI communities seeks to develop methods for autonomous decision-making that reason intelligently about the influence that an autonomous agents' decisions have on the behavior of others in its environment, often by employing a variety of learning algorithms on generalized problem formulations.

However, interactions between humans and robots are typically highly structured: Vehicles in traffic should obey traffic rules. Although rules could be incorporated into existing formulations in principle, recent works generally do not consider the influence that rules and conventions have on competitive multi-agent interactions. This is, in part, because rules often encode complex statements over the order of events along a trajectory –did agent 1 stop at the stop-sign before crossing? Did agent 1 yield for agent 2?– whose satisfaction depends on the decisions made by all the agents jointly. This makes rules challenging to represent mathematically and it can make it unclear who carries responsibility for breaking a rule, even though traffic rules generally make it unambiguous to humans which decisions are acceptable and which are not in a specific situation. Indeed, it is our view that the interaction-aware AV decision-making task would be significantly simplified if everyone obeyed traffic rules at all times, because many traffic rules (like driving on the right-hand side of the road) exist precisely to reduce the amount of negotiation between agents that is required to safely reach a destination.

Therefore, we 1) present a principled approach to incorporate traffic rules and conventions into multi-agent decision making and 2) study their influence on game theoretic formulations in this work. To do so, we model interaction rules and conventions as Linear Temporal Logical (LTL) constraints on the joint trajectories of all the agents in the scene, which we translate into chance constraints on an individual's decision making using reachability analysis. We then synthesize decision making-agents that explicitly and interpretably take the likelihood of others violating traffic rules into account. We dub the resulting chance-constrained stochastic game the *cautious Markov game* and provide a method to compute robust policies for N-player games and Nash equilibria in the zero-sum 2-player setting. We show on simulated examples that our assumptions accurately model common traffic scenarios and that we can significantly reduce the conservatism of adversarial approaches by taking traffic rules into account. Our results support our intuition that estimates of the likelihood with which other agents break rules present an actionable signal to incorporate in an AV decision-making stack and can resolve ambiguity inherent to game-theoretic approaches. As such, this work presents both a basic contribution towards intelligent, interaction-aware decision-making and an initial step towards formal synthesis methods for autonomous agents in multi-agent, rule-based environments, a problem that has received comparatively little attention in the literature [1]. Proofs of all our results and additional experimental details are available in the appendices of the extended version of this work, available at [2].

II. Related Work

Interaction-aware methods that move beyond myopically planning around trajectory forecasts have received increasing attention from robotics researchers in recent years, resulting in a diverse body of existing work ranging from ego-plan conditioned trajectory forecasting [3], [4], [5], [6] to learning the policies of other agents online [7]. In this work, we take a game theoretical approach to interaction aware-robotics, modeling opposing agents as rational with respect to some reward function to construct policies that anticipate how opponents will react, as advocated by a rapidly growing body of recent work that considers computing Nash equilibrium (NE) policies (e.g., see [8], [9], [10], [11], [12], [13], [14], [15], [16]).

However, computing Nash equilibria for general non-cooperative games is computationally intractable [17]. Only a handful of dynamic games, like the zero-sum (adversarial) stochastic game, have a unique solution that we can tractably compute [18], [19]. Therefore, recent game-theoretic algorithms generally focus on practical iterative or learning-based methods without strong guarantees, optimizing free-form trajectories in generic dynamic games. Many of these methods assume the objectives of other agents are known a priori (i.e., see [9], [10], [8], [20]). Others identify

[1]Rohan Sinha is with the department of Aeronautics and Astronautics at Stanford University. rhnsinha@stanford.edu

[2]Sanjay Lall is with the deparment of Electrical Engineering at Stanford University. lall@stanford.edu

parametric models, like neural networks or weighted basis functions, to capture the objectives of other road users, for example, through inverse reinforcement learning algorithms [21], [11], [22]. Research has tended towards such approaches because an adversarial (i.e., zero-sum) treatment is much too conservative for most applications. However, recent methods can both be fragile because they usually ignore uncertainties in learned quantities and lead to unverifiable decision making because of inherent ambiguity that can arise in non zero-sum games: Multiple distinct equilibria may exist to a game that encode behaviors that are qualitatively highly distinct, and practical NE solvers cannot reason about which equilibrium, if any, is found. For example, consider two people walking in opposite directions in a hallway. Will they cross each other to the left or to the right? Confusion about which equilibrium the other agent will select often results in an awkward shuffle where both agents commit to incompatible equilibria.

In safety critical robotics application, ambiguity arising from such *equilibrium selection* problems can lead to catastrophic accidents. One could seek to resolve ambiguity using approaches like online intent or reward inference [23], [24], some level of coordination or communication between agents [13], or through external randomization devices like the correlated equilibrium [25], [26]. Instead, our insight is that traffic rules resolve much of the need for negotiation and ambiguity inherent to game theoretic formulations: We drive on the right-hand side to avoid the "hallway shuffle" on the road. Therefore, we explicitly encode the structure in the control task by fixing the likelihood that other agents break traffic rules, leading to a novel chance-constrained stochastic game formulation.

Since traffic rules oftentimes specify the order in which events should occur, we model them as linear temporal logical (LTL) [27] constraints on the joint trajectories of the agents, in the spirit of recent work that transcribed traffic rules into temporal logic [28], [29]. Previously, formal methods have also been applied in the controls and robotics communities to guarantee that autonomous systems satisfy complex temporal behavior [30], [31], [32], [33]. These algorithms synthesize policies that satisfy temporal logical constraints for all time in various single-agent or cooperative multi-agent settings, using both exact solution methods and a variety of reinfocement learning approaches (e.g., see [27], [32], [30], [31], [34], [33], [35], [36]). These methods generally cannot handle the noncooperative multi-agent nature of rule-based open-world robotics tasks, as we cannot control whether the agents surrounding an autonomous vehicle follow traffic rules or not.

III. Preliminaries

In this section we introduce the basic notation of stochastic games and temporal logic, which we use to construct our formulation.

A stochastic game, or Markov game (MG), is a tuple $(\mathcal{S}, T, \{\mathcal{A}_i, R_i\}_{i=1}^N, \gamma)$ that extends the MDP formalism to the N-agent setting. Here, $\mathcal{S}$ is a *finite* set of shared states, $\mathcal{A}_i$ is the *finite* action set of agent $i \in \{1,\dots,N\}$. Furthermore, $T : \mathcal{S}\times\mathcal{A}\times\mathcal{S}\to[0,1]$ is the transition probability function, $R_i : \mathcal{S}\times\mathcal{A}\to\mathbb{R}$ is agent i's reward function, and $\gamma\in(0,1)$ is the discount factor. We use the shorthand notation $a \in \mathcal{A} := \prod_{i=1}^N \mathcal{A}_i$ to jointly refer to the actions played by the agents, and we use a negated subscript to refer to quantities belonging to all the agents except agent i. For example, $a_{-i} = (a_1,\dots,a_{i-1},a_{i+1},\dots,a_N)$. In addition, we use the notation $\Delta(\mathcal{X}) := \{\mathbf{x}\in\mathbb{R}^{|\mathcal{X}|} \;:\; \mathbf{x}\geq 0,\ \mathbf{x}^\intercal\mathbf{1}=1\}$ to refer to the set of categorical probability distributions over the elements of a *finite* set $\mathcal{X}$. In a stochastic game, agents act independently of each other according to stochastic policies $\pi_i : \mathcal{S}\to\Delta(\mathcal{A}_i)$ and we refer to their policies jointly using $\pi : \mathcal{S}\to\Delta := \prod_{i=1}^N \Delta(\mathcal{A}_i)$. Similar to an MDP, agents strive to maximize their expected discounted reward $V_i^\pi(s) = \mathbb{E}[\sum_{t=0}^\infty \gamma^t R_i(s^t,a^t) \mid s^0=s,\ a^t\sim\pi(s^t)]$ in an MG. However, the agents' objectives can be in tension with each other in stochastic games, so there generally is no single policy profile that maximizes the rewards of all the agents. Instead, we take the Nash equilibrium (NE) as the solution concept in stochastic games. A policy profile $\pi^\star$ constitutes a NE when

$$\pi_i^\star(s) \in \underset{\pi_i(s)\in\Delta(\mathcal{A}_i)}{\operatorname{argmax}} \; Q_i^{\pi^\star}(s,\pi_i(s),\pi_{-i}^\star(s)) \tag{1}$$

$$Q_i^{\pi^\star}(\cdot) := \mathbb{E}_{a\sim(\pi_i(s),\pi_{-i}^\star(s))}[R_i(s,a)+\gamma\mathbb{E}_{s'}[V^{\pi^\star}(s')|s,a]]$$

for all agents $i\in\{1,\dots,N\}$ and states $s\in\mathcal{S}$ [17]. NEs are an attractive solution concept because they result in stable interactions: No agent can unilaterally change its policy and obtain higher utility.

Linear temporal logic (LTL) **formulae** are built recursively from 1) a set of atomic propositions $\mathcal{P}$, functions of a time-varying input that either evaluate to True or False at each timestep, 2) composition with propositional logical connectives such as $\wedge$ (and), $\vee$ (or), $\neg$ (not), and $\to$ (implication), and 3) a set of temporal operators. The base temporal operators on an LTL formula ϕ are defined as

$$\mathbf{X}\phi \qquad \phi \text{ holds at the next timestep}$$
$$\phi_1\mathbf{U}\phi_2 \qquad \phi_1 \text{ is True until } \phi_2 \text{ becomes True},$$

from which other common shorthand operators like "always" or "never" are derived (see [27], [37] for examples). A trace, or input word w, is a *finite* sequence of truth values for each atomic proposition, i.e., for $b^i\in 2^\mathcal{P}$ (the input alphabet) we have that $w=(b^0,\dots,b^t)$. An input word w either satisfies a formula ϕ (written as $w\models\phi$), or it does not (written as $w\not\models\phi$). Therefore, LTL specifications allow us to formally verify statements over sequences of events, like traffic rules. To do so, we can use finite state machines (FSMs). Formally, we consider LTL formulas ϕ defined over atomic propositions that depend only on the system state $s\in\mathcal{S}$ (i.e., we have a map $L : \mathcal{S}\to 2^\mathcal{P}$). There then exists a FSM $(\mathcal{Q},\mathcal{S},q^0,\delta,F)$ so that when the FSM is given initial state $q_i^0\in\mathcal{Q}$ and follows the transition dynamics $\delta : \mathcal{Q}\times\mathcal{S}\to\mathcal{Q}$, it holds that ϕ is satisfied on a finite trajectory $h^t=(s^0,\dots,s^t)$ if and only if $q^{t+1}\in F\subseteq\mathcal{Q}$ [27], [37]. The subset F of the FSM state space $\mathcal{Q}$ is therefore called the set of accepting states, and we write $h^t\models\phi$ to indicate that the trajectory h^t satisfies the formula.

IV. The Cautious Stochastic Game

In this section we develop the multi-agent interaction formulation we dub the *Cautious Markov Game*. First, consider the multi-agent transition dynamics $(\mathcal{S},T,\{\mathcal{A}_i\}_{i=1}^N)$ with N agents that should each follow traffic rules and conventions, defined as LTL specifications over atomic propositions that are functions of the state $s\in\mathcal{S}$. Unfortunately, sometimes agents make decisions that end up breaking the rules, which we will simply model using fixed probabilities $p_i\in[0,1]$ for each agent. To do so, we first need to understand how rules constrain the behavior of individual agents, even though they encode statements over trajectories that result from the decisions made by all the agents combined.

A. Traffic Rule Desiderata

To develop a precise notion of what it means for an individual agent to break a rule, we first note that arbitrary LTL rules can result in ambiguity on which individual(s) is at fault for a rule violation: Consider two agents navigating an intersection, both of which can instantaneously stop their vehicle and should follow the "do not collide" traffic rule. If these agents collide, either might argue that the other was at fault, as they could both have taken unilateral action to avoid the accident. As this example illustrates, poorly chosen rules make it unclear who is at fault when the rules are violated from a causal modeling perspective [38]. Moreover, notions of joint or partial responsibility imply a need for coordination among the agents to satisfy the rules in general. However, traffic rules usually make it unambiguous how road-users should act. Therefore, we construct our approach around two basic observations on the properties that common traffic rules and conventions satisfy:

1. Rules are discriminative: We are guided by the intuition that acceptable interaction rules should be discriminative in nature. For example, a traffic rule should be satisfied by default if all the agents decide to stay at home. As such, agents only violate the rules when they engage in behavior that was explicitly disallowed, which should make it possible to pinpoint the instant at which a rule was violated as the timestep after which an LTL formula was longer satisfied (e.g., the instant at which a red light was run). Therefore, the discriminative nature of a good rule should not require some event to happen eventually (a common LTL specification), as this would imply that all the agents are breaking the rule until this event happens.

2. Rules eliminate coordination: We take the view that traffic rules generally exist to reduce the need for coordination among the agents. For example, traffic rules specify that we drive on the right-hand side of the road so that we do not have to coordinate with oncoming traffic. This perspective implies that an agent violating a rule is a result of their own decisions regardless of the behavior of others, unambiguously specifying acceptable behavior for each agent individually.

Therefore, we define the event of an agent breaking traffic rules as follows:

Definition 1. *Agent $i\in\{1,...,N\}$ breaks its associated* LTL *rule ϕ_i at time $t\in\mathbb{N}$ if for $k\in\{0,...,t-1\}$ it holds that the trajectory history $h^k\models\phi_i$ and $h^t\not\models\phi_i$.*

Under our desiderata, each agent carries unilateral responsibility for breaking its associated traffic rule; if agent i runs a red light or does not yield at a roundabout, we consider this a consequence of agent i's decisions and not the other agents'. Moreover, to decide when an agent breaks a rule, we need to assume that the agent can unilaterally make decisions to satisfy the rule for all time from a nonempty set of initial conditions; nobody should be blamed for something that was out of their control from the start. We will examine common traffic scenarios in §VI that support our hypothesis—that these desiderata model the vast majority of traffic rules.

B. Product Dynamics

To construct the Cautious Markov Game, we first take advantage of the defining property of the LTL formulae $\phi_1,...,\phi_N$: That we can easily verify whether the state history at some time $t\in\mathbb{N}$, defined as $h^t=(s^0,...,s^t)$ with $s^k\in\mathcal{S}$ as the state at time k, satisfies the rules using finite state machines (FSMs). We therefore augment the state $s\in\mathcal{S}$ using the FSM states associated with the rules, a common approach in the literature (e.g. see [30], [32], [27] for a detailed discussion).

Definition 2. *Let $(\mathcal{Q}_i,\mathcal{S},q_i^0,\delta_i,F_i)$ be the FSM associated with rule ϕ_i for $i=1,...,N$. The product dynamics associated with the multi-agent MDP $(\mathcal{S},T,\{\mathcal{A}_i\}_{i=1}^N)$ and a set of rules $\phi_1,...,\phi_N$ is the tuple $(\mathcal{S}_p,T_p,\{\mathcal{A}_i\}_{i=1}^N)$, where the product state $s_p\in\mathcal{S}_p=\mathcal{S}\times\mathcal{Q}_1,...,\mathcal{Q}_N$ and the product dynamics $T_p:\mathcal{S}_p\times\mathcal{S}_p\times\mathcal{A}\to[0,1]$ are given as*

$$T_p(s_p'|s_p,a)=\begin{cases}T(s'|s,a) & \text{if } \delta_i(q_i,s')=q_i' \ \ \forall\ i\in\{1,...,N\}\\ 0 & \text{else}\end{cases}. \tag{2}$$

The product dynamics are essentially equivalent to that of the original multi-agent MDP: They simply represent the joint evolution of the system state s, which the agents observe, and the FSM states $q_1,...,q_N$, which the agents track internally. However, we can now focus our attention on memoryless policies of the product state s_p, as a trajectory history h^t violates ϕ_i if and only if $s_p\in\mathcal{S}_p$ evolves into the *avoid set*

$$\mathcal{F}_i:=\{(s,q_1,...,q_N)\in\mathcal{S}_p:q_i\notin F_i\}, \tag{3}$$

where F_i is the set of accepting conditions of the FSM associated with ϕ_i.

C. The Cautious Markov Game

Leveraging the formalism of the product dynamics, we now construct the *cautious Markov game* in three steps. First, we note that recognizing the timestep at which a rule was broken is separate from the question of when violation of the rule became inevitable: running a red light may become inevitable if a car drives too fast to be able to stop in time, but the light will not be run until the car enters the intersection. Therefore, we identify the set $\mathcal{R}_i\subseteq\mathcal{S}_p$ from which agent i can unilaterally avoid reaching $\mathcal{F}_i$ for all future timesteps, defined as

$$\mathcal{R}_i:=\left\{s_p\in\mathcal{S}_p\ \middle|\ \begin{array}{c}\exists\pi_i\in\Pi_i \text{ s.t. } \mathrm{Prob}(s_p^t\in\mathcal{F}_i|s_p^0=s_p)=0,\\ \forall\pi_{-i}\in\Pi_{-i},\ t\geq 0\end{array}\right\}. \tag{4}$$

Here, Π_i is the set of all Markov policies in the product state space for agent i. $\mathcal{R}_i$ is the set of states from which agent i can guarantee ϕ_i is satisfied for all time and for all opponent policies. The set $\mathcal{R}_i$ therefore reflects the set of states from which agent i can unilaterally satisfy its associated traffic rule without coordinating with other agents, making precise the unilateral responsibility as noted in our desiderata.

Secondly, we use $\mathcal{R}_i$ to translate the violation of agent i's traffic rule ϕ_i, a statement over realized state trajectories $h^t\in\mathcal{S}^t$, to a statement over agent i's decision making. Specifically, we define the set of *prudent*, or *good*, actions for agent i at product state $s_p\in\mathcal{S}_p$ as

$$\mathcal{G}_i(s_p):=\left\{a_i\in\mathcal{A}_i\ \middle|\ \begin{array}{l}T_p(s_p'|s_p,a_i,a_{-i})=0,\\ \forall s_p'\in\mathcal{S}_p\setminus\mathcal{R}_i,\ a_{-i}\in\mathcal{A}_{-i}\end{array}\right\}, \tag{5}$$

and we take the set of *imprudent*, or *bad*, actions as its complement $\mathcal{B}_i(s_p):=\mathcal{A}_i\setminus\mathcal{G}_i(s_p)$. For a state $s_p\in\mathcal{R}_i$, the set $\mathcal{G}_i(s_p)$ collects all the actions for which agent i can guarantee the state remains in $\mathcal{R}_i$ almost surely, regardless of the decisions of the other agents. Conversely, if agent i takes an *imprudent action*, there is a nonzero probability that the state evolves

into $\mathcal{S}_p \setminus \mathcal{R}_i$, from which agent i can no longer guarantee that the rule will be satisfied for all time. We emphasize that taking an imprudent action does not immediately imply that agent i's rule will be broken. For example, an AV cannot violate a crosswalk yield rule if a pedestrian waiting at the corner does not cross, even if the AV decides to drive too fast to be able to yield in time (a result of imprudent decision). Still, under our desiderata, agents should not take imprudent actions, as they will be at fault if a rule violation occurs. To make this intuition precise, we prove in Appendix III of the extended version of this work [2] that agent i's rule is satisfied for all time and all opponent policies $\pi_{-i} \in \Pi_{-i}$ if and only if the initial condition $s_p^0 \in \mathcal{R}_i$ and $a_i^t \in \mathcal{G}_i(s_p^t)$ for all $t \geq 0$.

Therefore, if agent i breaks rule ϕ_i, this means agent i took at least one *imprudent action*, that is, an action that can result in a future rule violation. Therefore, we simply consider the probability that an agent takes an imprudent action as a fixed quantity, perhaps estimated from data, giving rise to the *Cautious Stochastic Game* that we consider in this work.

Definition 3. *A Cautious Markov Game (CMG) is the tuple* $(\mathcal{S}_p, T_p, \{\mathcal{A}_i, \mathcal{G}_i, \mathcal{B}_i, R_i, p_i\}_{i=1}^N, \gamma)$. *Here* $\mathcal{S}_p$ *is the shared set of product states,* $\mathcal{A}_i$ *is the action set for agent* i, $T_p : \mathcal{S}_p \times \mathcal{S}_p \times \mathcal{A} \to [0,1]$ *is the state transition probability,* $\gamma \in (0,1)$ *is the discount factor,* $R_i : \mathcal{S}_p \times \mathcal{A} \to \mathbb{R}$ *is agent* i*'s reward function, and*

- $\mathcal{G}_i(s_p) \subseteq \mathcal{A}_i$ *and* $\mathcal{B}_i(s_p) = \mathcal{A}_i \setminus \mathcal{G}_i(s_p)$ *are the sets of* prudent *and* imprudent *actions for agent* i *at a state* $s_p \in \mathcal{S}_p$,
- $p_i \in [0,1]$ *is the probability that agent* i *takes an* imprudent *action, i.e. an action in* $\mathcal{B}_i(s_p)$*, whenever* $\mathcal{B}_i(s_p) \neq \emptyset$ *and* $\mathcal{G}_i(s_p) \neq \emptyset$.

The difference between a cautious game and an unconstrained stochastic game is that the agents take imprudent actions with a fixed likelihood p_i. That is, at a state $s_p \in \mathcal{S}$, their policies are constrained to the set $\Delta_i^G(s_p) := \{\mathbf{z}_i \in \Delta(\mathcal{A}_i) : \text{Prob}(a_i \in \mathcal{G}_i(s_p)) = 1 - p_i\}$. Therefore, our formalism differs from the constraints on cumulative discounted expected costs considered in constrained MDPs [39]. Our formalism means that the agents need to take the likelihood of their opponents breaking the rules into account to maximize their expected cumulative reward and act *cautiously* towards their opponents, which is why we dubbed this formulation a cautious Markov game. It is a classic result for stochastic games that memoryless equilibrium policies exist [17]. Similarly, we consider the problem of identifying Markov perfect equilibria of the cautious Markov game.

V. Solving the Cautious Game

In this section we build up our approach towards the cautious Markov game that we defined in §IV. First, we show how to compute $\mathcal{R}_i$ for each agent $i \in \{1,...,N\}$, to define a Cautious Markov Game associated with an MG and rules $\phi_1,...,\phi_N$. Then, we construct algorithms to efficiently solve for Nash equilibria in the 2-player zero-sum case and compute robust (maxmin) policies for a general N-agent setting.

Identifying Rule-Breaking and Rule-Following Actions: To identify the rule-breaking and rule-following action sets $\mathcal{B}_i(s_p)$ and $\mathcal{G}_i(s_p)$ necessary to formulate the Cautious Markov Game, we need to compute the safe set of states $\mathcal{R}_i$ associated with each agent $i \in \{1,...,N\}$. Identifying the set $\mathcal{R}_i$ generally involves solving a *reach-avoid game*, where the ego's objective is to steer clear of $\mathcal{F}_i$ and the opponents' joint disturbance maximizes the likelihood of entering $\mathcal{F}_i$. For multi-agent MDPs, this can be done by taking advantage of classic results for zero-sum stochastic games [18]. Similarly, for continuous time systems, sets like $\mathcal{R}_i$ defined in (4) are often computed using Hamilton-Jacobi (HJ) reachability analysis [40]. We make a simple observation to reduce computational cost: Since there are a finite number of decisions that the other agents can make, we can perform any robust (i.e. for all opponent actions) reachability computation for agent i by *uniformly randomizing* over the decisions made by the opposing agents.

Theorem 1. *Define the MDP* $(\mathcal{S}_p, T_i, \mathcal{A}_i, R_i^{\text{reach}}, \gamma)$ *as a reachability problem for each agent* $i = 1, \ldots, N$*, with randomized transition dynamics and reward functions given as*

$$T_i(s_p' | s_p, a_i) := \frac{1}{|\mathcal{A}_{-i}|} \sum_{a_{-i} \in \mathcal{A}_{-i}} T_p(s_p' | s_p, a_i, a_{-i}), \quad (6)$$

$$R_i^{\text{reach}}(s_p, a_i) := \begin{cases} -1 & \text{if } s_p \in \mathcal{F}_i \\ 0 & \text{else} \end{cases}, \quad (7)$$

and $\gamma \in (0,1)$*. Let* π_i^{reach}, V_i^{reach} *be the optimal policy and value function associated with this MDP. Then, it holds that*

$$\mathcal{R}_i = \{s_p \in \mathcal{S}_p : V_i^{\text{reach}}(s_p) = 0\}. \quad (8)$$

Proof: For a proof of Theorem 1, see Appendix IV in the extended version [2]. We use Bellman's principle of optimality to show that if $V_i^{\text{reach}}(s_p^t) = 0$, then agent i can take an action to guarantee that $V_i^{\text{reach}}(s_p^{t+1}) = 0$. Then, we show that $s_p \in \mathcal{R}_i$ implies that $V_i^{\text{reach}}(s_p) = 0$.

Therefore, we can compute $\mathcal{R}_i$, and by extension $\mathcal{G}_i(s_p)$ and $\mathcal{B}_i(s_p)$, efficiently using single-agent value iteration. We consider developing algorithms to compute the prudent and imprudent action sets for continuous dynamical systems using methods to compute safe sets like HJ reachability [40] or control barrier functions (CBFs) [41] as interesting directions for future work.

Learning Imprudent Action Likelihoods: The rule-violation probabilities p_i for all non-ego agents are straightforward to estimate from a dataset of K trajectories $\mathcal{D} = \{\{s^j, a^j\}_{j=1}^{T_k}\}_{k=1}^K$, by counting the events when $a_i \in \mathcal{G}_i(s_p)$ in the product state space. Moreover, we emphasize that our results apply not only when p_i is fixed, but also when it is a function of the product state, i.e., when $p_i : \mathcal{S}_p \to [0,1]$. This means p_i can be a local property of the environment, which we can estimate using neural networks.

Robustly Solving Cautious Stochastic Games: Adversarially robust policies are an ideal candidate to verify how much negotiation is needed in an interaction: If a robust policy performs reasonably, then there is little need to reason about opponents' decision making. Moreover, robust policies are tractable to compute [32], [18]. In contrast, equation (1) shows that we need to solve a normal form game at each state (the stage games) to compute a NE of a stochastic game. As a result, computing NEs for MGs using a value-iteration-like algorithm is computationally hard. Moreover, it is well known that, even if we assume we could compute equilibria to the stage games, a value iteration-like algorithm need not converge to a NE [26], [42]. This makes solving stochastic games a challenging discipline fraught with subtleties.

The stage games that we need to solve in a cautious game are different from the normal form stage games in a regular MG because we place chance constraints on the agents'

policies: by fixing p_i, we can align the result of game theoretic analysis with partial observations on an opponent's behavior without modifying the incentive structures of the agents. We include an extended discussion on the stage games of the CMG in Appendix V of the extended version [2], where we show that the cautious stage games inherit the computational hardness of normal form games. This means that the hardness results of stochastic games extend to the CMG.

Therefore, given a set of rule-violation probabilities $p_1,...,p_N$, we compute policies for an agent of interest (the ego), using a *robust value-iteration* (VI) procedure instead. In a robust VI scheme, we assume the other agents coordinate together against the ego by solving worst-case adversarial stage games similar to [43], [32], [18]. We summarize this procedure in algorithm 1, included in Appendix VII of the extended version [2]. In comparison with the classical Value Iteration algorithm, in robust VI, we solve an LP at each state for each iteration. The robust value function $V_i^\star$ for agent i and its associated implicit policy $\pi_i^\star$ returned by the robust VI algorithm produce a certificate of robustness for agent i, that is, that agent i's expected utility will be at least $V_i^\star$ under the implicit robust policy.

Theorem 2. *Let $V_i^\star$ be the output of algorithm 1 for agent i, with implicit maxmin policy $\pi_i^\star$ such that $\pi_i^\star(s_p) \in \Delta_i^G(s_p)$ at each $s_p \in \mathcal{S}_p$. Let V_i be the value function for agent i under $\pi_i^\star$ and any opponent policies π_{-i} such that $\pi_{-i}(s_p) \in \Delta_{-i}^G(s_p)$ at each $s_p \in \mathcal{S}_p$. Then it holds for any $s_p \in \mathcal{S}_p$ that $V_i^\star(s_p) \leq V_i(s_p)$.*

Proof: For a proof, see Appendix VIII in [2]. We first show that robust value iteration converges to a unique fixed point using contraction theory, analogous to [18], [32], after which we apply Bellman's principle of optimality to yield the theorem.

Moreover, since we can compute the subproblems of robust VI using LP, it follows that we can efficiently compute solutions in polynomial time. In addition, we show in Appendix VIII of the extended version [2] that in the 2-player zero-sum case, algorithm 1 identifies the unique NE of the CMG. Therefore, We can view algorithm 1 as a generalization of the classic algorithm proposed for 2-player zero-sum Markov Games in [18]. We note that for general Markov games without rules, the robust policy computed for an ego robot is often much too conservative to find practical use. Especially in autonomous driving, a collision penalty for the ego agent will then result in a robust policy that assumes other road users will try to collide with the ego agent. However, by specifying the game's rules, the CMG constrains the other agents only to take actions that violate the rules with a fixed probability. Hence, if the rules are well-designed, much of the ambiguity of the other agents' behavior is eliminated a priori, thereby reducing the conservatism of the robust policy without requiring inverse RL approaches and searching for general-sum Nash equilibria.

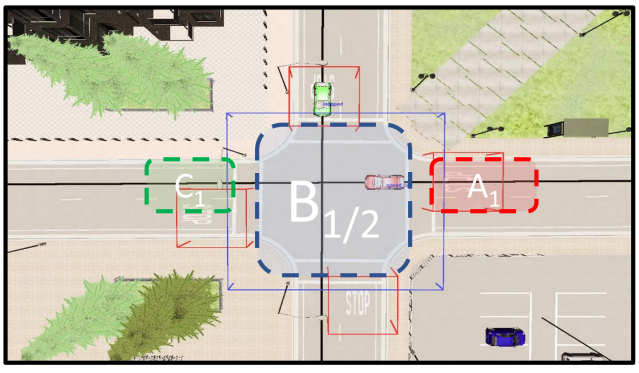

Fig. 1. Snapshot of the behavior of the cautious agent (red car) and the opponent (green car) at a 4-way intersection in CARLA, annotated to illustrate the atomic propositions associated with (9). The highlighted bounding boxes indicate the atomic propositions associated with A_1 (red), B_i (blue), and C_1 (green) respectively. The reference trajectories are in black.

VI. Simulations

In this section, we 1) show that we can model common scenarios in AV decision-making using the CMG framework and 2) illustrate traffic rules' influence on the problem formulations in interaction-aware robotics on concrete examples using the CARLA autonomous driving simulator [44]. For simplicity, our examples contain two agents that navigate traffic intersections following pre-specified lane reference trajectories, an autonomous ego vehicle (agent 1) and an opponent (agent 2). The state of each of the agents consists of their respective position and velocity $s_i = (x_i, v_i)$. We focus on high-level decision making and take the action set of each agent as $\mathcal{A}_i = \{\text{slow down}, \text{ maintain speed}, \text{ speed up}\}$, which modifies the reference commands fed to low-level PIDs by a pre-specified amount. Velocities are limited to 10km/h, beyond which the cars will not speed up more. In these examples, we learn the likelihood $p_2(\mathcal{S}_p)$ that the opponent takes imprudent actions from data collected with a robust policy computed on the original MG, thereby conservatively ignoring traffic rules. For more experiment details, see Appendix IX in [2] and our accompanying video.

Comparisons: We compare our *cautious* approach that estimates the opponent's imprudent action likelihood p_2 with 1) a naive, *optimistic* approach which simply assumes the opponent never takes imprudent actions (i.e., it optimistically assumes that $p_2 = 0$) and 2) a conservative, *pessimistic* approach that conservatively assumes the opponent always takes imprudent actions when possible (i.e., it pessimistically assumes $p_2 = 1$). We chose these baselines because existing methods do not account for traffic rules and therefore cannot explicitly reason about the likelihood of other agents taking imprudent actions and breaking the rules. Moreover, computing or learning solutions to general-sum games using commonplace but unverifiable algorithms can lead to sub-optimalities induced by hyperparameter choices, equilibrium selection problems, and nonstationary learning dynamics, which can confound our results. Instead, we compute adversarially robust strategies under the assumed value of p_2 to ensure that we can solely attribute our results to traffic

	$p_2 = 0$	$p_2 = \frac{1}{5}$	$p_2 = \frac{4}{5}$	$p_2 = 1$
Cautious	**5.3**	**3.2**	**.16**	**0.0**
Optimist	**5.3**	**3.2**	-.95	-2.0
Pessimist	1.2	0.8	.11	**0.0**

TABLE I

Utilities for the 4-way intersection simulation. The table contains the realised ego utility $V_1(s_p^0)$ against an opponent that takes imprudent actions with true probability p_2. The cautious agent computed its policy using the estimated value of p_2.

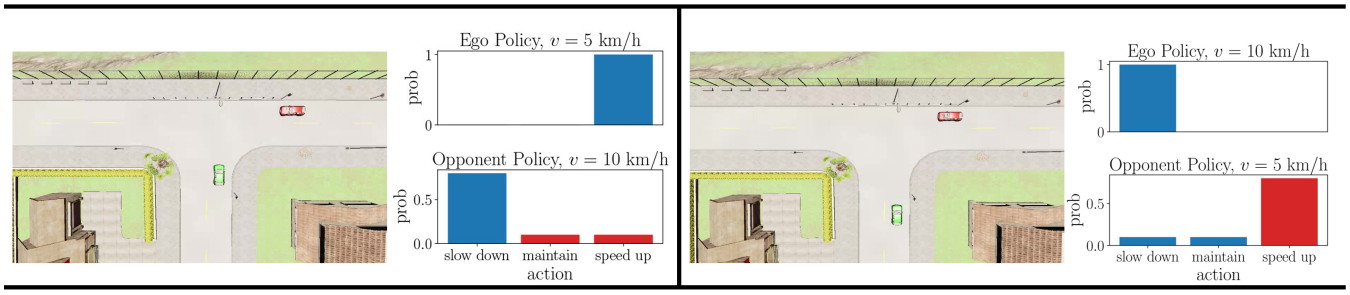

Fig. 2. Behavior of the cautious agent (red car) and the opponent (green car) at a T-junction when $p_2 = .2$ (left) and when $p_2 = .8$ (right). The histograms indicate the action distributions $\pi_i(s_p^t)$ of the agents at a timestep t and the images show the associated world state s^t. The histograms show the prudent actions in blue, the imprudent actions in red.

rules' influence on a problem formulation rather than how well any particular algorithm performs on a given problem.

Stop-signed Intersection: Figure 1 shows the two agents navigating a 4-way stop-signed intersection. The agents receive a positive reward for reaching their destination before their opponent and are penalized for collisions. Both agents are expected to respect the first-in-first-out (FIFO) traffic principle, necessitating an (ϕ Before ϕ') operator, which we construct from the base LTL operators as $\mathbf{SB}$ (read as *strictly before*) in Appendix II of [2]. To construct the rule, we take the atomic propositions A_i, B_i, C_i as indicators on whether agents 1,2 have arrived at the intersection, are occupying the intersection, or have crossed the intersection respectively. We evaluate these atomic propositions by checking whether the vehicles are inside bounding boxes, as illustrated in Fig. 1. We can then write the FIFO rules as

$$\phi_i = (A_j \ \mathbf{SB} \ A_i) \rightarrow (C_j \ \mathbf{SB} \ B_i), \tag{9}$$

for $(i,j) \in \{(1,2),(2,1)\}$, which reads as *if agent j arrives at the intersection before agent i, then agent j should cross before agent i enters the intersection.* Our ego agent should never break the rules, so we set the imprudent action likelihood $p_1 = 0$. We vary the true imprudent action likelihood p_2 of the opponent and compare the performance of the cautious ego agent against the optimistic and pessimistic ego agents. In this scenario, the ego arrives at the stop sign first. Table I shows that the cautious agent 1) is less conservative than the pessimist and 2) performs better than the optimist. Qualitatively, this is because the ego decides to cross the road if p_2 is low, but lets the opponent pass when they are likely to cross out of turn. In contrast, the optimistic agent makes the unsafe decision to cross first even when the opponents' imprudent action likelihood is high, whereas the pessimistic agent is overly conservative: it always waits for the opponent to cross first.

Merging at T-Junction: To underscore the qualitative behavior of the *cautious* policies, consider the scenario where the opponent needs to merge into the lane occupied by the ego at a T-junction, as shown in Fig. 2. In this example, the agents receive rewards for following the speed limit and are penalized for collisions. The ego agent drives on a road with priority, so it does not have a traffic rule. The opponent should follow a yield rule, roughly given as *if the vehicles will get too close in the future extrapolating at the current speeds, then the ego should cross the intersection before the opponent* (see Appendix IX of [2] for details and the exact LTL statement.) In Fig. 2 we see how the rule and the violation likelihood of the opponent affects the agents' decision making: Even when the vehicles are close to each other, a low violation likelihood results in the ego confidently accelerating across the intersection, knowing that the opponent is most likely to yield. In the other scenario, the vehicles are further away from each other. However, since the violation likelihood is high, the ego agent slows down early for the opponent because it expects the opponent will drive too fast to slow down in time.

VII. Conclusion, Limitations, and Future Work

In this work, we developed a precise notion of what it means for an agent to break a traffic rule in a stochastic game. First, we used this definition to convert rule-breaking, a statement over trajectories, to one over the agents' decisions. Then, given an estimate of the likelihood that agents take imprudent actions, we discussed how facts about normal-form games generally translate to the setting in which we know this prior over actions. Our simulations showed that accounting for the rule-based nature of the interaction through the prudent and imprudent actions can significantly reduce the conservatism of robust (max-min) policies on a simple example. This showcases that 1) traffic rules reduce the need for inter-agent negotiation 2) estimates of when opponents make imprudent decisions are an actionable signal in an AV stack.

Limitations and Future Work: We studied how to interpretably incorporate traffic rules in a multi-agent decision-making context, as well as its implications for interaction-aware methods for decision-making. Although our solution approach results in good performance on our simulations, it scales poorly in the state-space size, which often grows exponentially with the number of agents in a scene. Therefore, developing efficient algorithms to scale to complex settings with many agents, for example, by adapting online Monte-Carlo tree search [45] or applying methods from multi-agent reinforcement learning [7], [46], potentially by predicting the likelihood of imprudent actions or representing policies using recurrent neural networks, present exciting avenues for future research. In addition, we intend to incorporate traffic rules into practical state-of-the-art general-sum game solvers like [8], [10] using our framework to reduce conservatism further. Another limitation of our work is that we cannot automatically admit the addition of new agents; we need to recompute solutions in this case. Nevertheless, the conclusions of this work should extend to more complex settings, making it an actionable starting point for future research.

References

[1] 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.

[2] Rohan Sinha and Sanjay Lall. Cautious Markov Games for Interaction Aware Robotics (extended version). Available at https://rohansinha.nl/icra2023/, 2022.

[3] Boris Ivanovic and Marco Pavone. The trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[4] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *Computer Vision – ECCV 2020*, pages 683–700, Cham, 2020. Springer International Publishing.

[5] Christoforos Mavrogiannis, Jonathan A. DeCastro, and Siddhartha S. Srinivasa. Implicit multiagent coordination at unsignalized intersections via multimodal inference enabled by topological braids. 15th International Workshop on the Algorithmic Foundations of Robotics (WAFR), 2020.

[6] Junha Roh, Christoforos I. Mavrogiannis, Rishabh Madan, Dieter Fox, and Siddhartha S. Srinivasa. Multimodal trajectory prediction via topological invariance for navigation at uncontrolled intersections. In *4th Conference on Robot Learning, CoRL 2020, 16-18 November 2020, Virtual Event / Cambridge, MA, USA*, volume 155 of *Proceedings of Machine Learning Research*, pages 2216–2227. PMLR, 2020.

[7] Michael L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the Eleventh International Conference on International Conference on Machine Learning*, ICML'94, page 157–163, San Francisco, CA, USA, 1994. Morgan Kaufmann Publishers Inc.

[8] Simon Le Cleac'h, Mac Schwager, and Zachary Manchester. ALGAMES: A Fast Augmented Lagrangian Solver for Constrained Dynamic Games. *Autonomous Robots*, 2021.

[9] Riccardo Spica, Davide Falanga, Eric Cristofalo, Eduardo Montijano, Davide Scaramuzza, and Mac Schwager. A Real-Time Game Theoretic Planner for Autonomous Two-Player Drone Racing. *IEEE Transactions on Robotics*, 36(5), 2020.

[10] David Fridovich-Keil, Ellis Ratner, Lasse Peters, Anca D. Dragan, and Claire J. Tomlin. Efficient iterative linear-quadratic approximations for nonlinear multi-player general-sum differential games. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1475–1481, 2020.

[11] Dorsa Sadigh, Shankar Sastry, Sanjit A Seshia, and Anca D Dragan. Planning for Autonomous Cars that Leverage Effects on Human Actions. page 9, 2016.

[12] G. Ye, Q. Lin, T.-H. Juang, and H. Liu. Collision-free Navigation of Human-centered Robots via Markov Games. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11338–11344, May 2020. ISSN: 2577-087X.

[13] Alessandro Zanardi, Enrico Mion, Mattia Bruschetta, Saverio Bolognani, Andrea Censi, and Emilio Frazzoli. Urban Driving Games With Lexicographic Preferences and Socially Efficient Nash Equilibria. *IEEE Robotics and Automation Letters*, 6(3):4978–4985, July 2021. Conference Name: IEEE Robotics and Automation Letters.

[14] Mingyu Wang, Zijian Wang, John Talbot, J. Christian Gerdes, and Mac Schwager. Game Theoretic Planning for Self-Driving Cars in Competitive Scenarios. In *Robotics: Science and Systems XV*. Robotics: Science and Systems Foundation, June 2019.

[15] Andreas Britzelmeier and Axel Dreves. A decomposition algorithm for Nash equilibria in intersection management. *Optimization*, 0(0):1–38, June 2020. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/02331934.2020.1786088.

[16] Axel Dreves and Matthias Gerdts. A generalized Nash equilibrium approach for optimal control problems of autonomous cars. *Optimal Control Applications and Methods*, 39(1):326–342, 2018. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/oca.2348.

[17] Yoav Shoham and Kevin Leyton-Brown. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, 2008.

[18] L. S. Shapley. Stochastic Games. *Proceedings of the National Academy of Sciences*, 39(10):1095–1100, October 1953. Publisher: National Academy of Sciences Section: Mathematics.

[19] Tamer Başar and Geert Jan Olsder. *6. Nash and Saddle-Point Equilibria of Infinite Dynamic Games*, pages 265–363. 1998.

[20] Ran Tian, Nan Li, Ilya Kolmanovsky, Yildiray Yildiz, and Anouck R. Girard. Game-theoretic modeling of traffic in unsignalized intersection network for autonomous vehicle control verification and validation. *IEEE Transactions on Intelligent Transportation Systems*, 23(3):2211–2226, 2022.

[21] Andrew Y. Ng and Stuart Russell. Algorithms for Inverse Reinforcement Learning. In *in Proc. 17th International Conf. on Machine Learning*, pages 663–670. Morgan Kaufmann, 2000.

[22] Lasse Peters, David Fridovich-Keil, Vicenç Rubies-Royo, Claire Tomlin, and Cyrill Stachniss. Inferring objectives in continuous dynamic games from noise-corrupted partial state observations. 07 2021.

[23] Tirthankar Bandyopadhyay, Kok Sung Won, Emilio Frazzoli, David Hsu, Wee Sun Lee, and Daniela Rus. Intention-Aware Motion Planning. In *Algorithmic Foundations of Robotics X*, volume 86, pages 475–491. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. Series Title: Springer Tracts in Advanced Robotics.

[24] D. Sadigh, S. S. Sastry, S. A. Seshia, and A. Dragan. Information gathering actions over human internal state. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 66–73, October 2016. ISSN: 2153-0866.

[25] Tim Roughgarden. *Twenty Lectures on Algorithmic Game Theory*. Cambridge University Press, Cambridge, 2016.

[26] Amy Greenwald and Keith Hall. Correlated Q-Learning. page 8, 2003.

[27] Christel Baier and Joost-Pieter Katoen. *Principles of model checking*. The MIT Press, Cambridge, Mass, 2008. OCLC: ocn171152628.

[28] Albert Rizaldi and Matthias Althoff. Formalising Traffic Rules for Accountability of Autonomous Vehicles. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, pages 1658–1665, Gran Canaria, Spain, September 2015. IEEE.

[29] Klemens Esterle, Luis Gressenbuch, and Alois Knoll. Formalizing Traffic Rules for Machine Interpretability. *2020 IEEE 3rd Connected and Automated Vehicles Symposium (CAVS)*, pages 1–7, November 2020. arXiv: 2007.00330.

[30] D. Sadigh, E. S. Kim, S. Coogan, S. S. Sastry, and S. A. Seshia. A learning based approach to control synthesis of Markov decision processes for linear temporal logic specifications. In *53rd IEEE Conference on Decision and Control*, pages 1091–1096, December 2014. ISSN: 0191-2216.

[31] Xiao Li, Cristian-Ioan Vasile, and Calin Belta. Reinforcement learning with temporal logic rewards. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3834–3839, Vancouver, BC, September 2017. IEEE.

[32] E. M. Wolff, U. Topcu, and R. M. Murray. Robust control of uncertain Markov Decision Processes with temporal logic specifications. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pages 3372–3379, December 2012. ISSN: 0743-1546.

[33] Vasumathi Raman, Alexandre Donzé, Dorsa Sadigh, Richard M. Murray, and Sanjit A. Seshia. Reactive synthesis from signal temporal logic specifications. In *Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control*, HSCC '15, pages 239–248, New York, NY, USA, April 2015. Association for Computing Machinery.

[34] C. Sun, X. Li, and C. Belta. Automata Guided Semi-Decentralized Multi-Agent Reinforcement Learning. In *2020 American Control Conference (ACC)*, pages 3900–3905, July 2020. ISSN: 2378-5861.

[35] Karen Leung, Nikos Aréchiga, and Marco Pavone. Back-propagation through Signal Temporal Logic Specifications: Infusing Logical Structure into Gradient-Based Methods. *arXiv:2008.00097 [cs, eess]*, January 2021. arXiv: 2008.00097.

[36] Jonathan DeCastro, Karen Leung, Nikos Arechiga, and Marco Pavone. Interpretable Policies from Formally-Specified Temporal Properties. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–7, Rhodes, Greece, September 2020. IEEE.

[37] Giuseppe De Giacomo and Moshe Y. Vardi. Linear Temporal Logic and Linear Dynamic Logic on Finite Traces. In *IJCAI '13 Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 854–860. Association for Computing Machinery, 2013. Accepted: 2014-11-21T22:06:43Z.

[38] Judea Pearl. *Causality*. Cambridge University Press, 2 edition, 2009.

[39] Eitan Altman. *Constrained Markov Decision Processes*. 1999.

[40] Somil Bansal, Mo Chen, Sylvia Herbert, and Claire J. Tomlin. Hamilton-jacobi reachability: A brief overview and recent advances, 2017.

[41] Aaron D. Ames, Samuel Coogan, Magnus Egerstedt, Gennaro Notomista, Koushil Sreenath, and Paulo Tabuada. Control barrier functions: Theory and applications. In *2019 18th European Control Conference (ECC)*, pages 3420–3431, 2019.

[42] Martin Zinkevich, Amy Greenwald, and Michael Littman. Cyclic equilibria in markov games. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems*, volume 18. MIT Press, 2005.

[43] Arnab Nilim and Laurent El Ghaoui. Robust control of markov decision processes with uncertain transition matrices. *Operations Research*, 53(5):780–798, 2005.

[44] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017.

[45] Cameron B. Browne, Edward Powley, Daniel Whitehouse, Simon M. Lucas, Peter I. Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(1):1–43, 2012.

[46] Lucian Busoniu, Robert Babuska, and Bart De Schutter. A Comprehensive Survey of Multiagent Reinforcement Learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*,

38(2):156–172, March 2008. Conference Name: IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews).
[47] John F. Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences*, 36(1):48–49, 1950.
[48] Sanjoy Dasgupta, Christos Papadimitriou, and Umesh Vazirani. *Algorithms*. McGraw-Hill, 2008.

Symbol	Description
x	variables or elements of a set are lowercase
$\mathbf{x}$	vectors are boldfaced
$\mathcal{X}$	sets are caligraphic
x^t	time-varying quantities are indexed with superscript $t \in \mathbb{N}_{\geq 0}$
x_i	agent specific quantities are indexed with subscript $i \in \{1,...,N\}$
$\mathbf{U}$	temporal operators are uppercase and boldfaced
$\mathbf{R}$	matrices are also uppercase and boldfaced
N	number of agents
$\mathcal{S}$	finite set of states
$\mathcal{A}_i$	finite set of actions for agent i
R_i	reward function for agent i
T	Transition probability function
γ	discount factor
s	state, element of $\mathcal{S}$
a_i	action played by agent i, element of $\mathcal{A}_i$
$\mathcal{A}$	joint action set, $\mathcal{A}_1 \times \cdots \times \mathcal{A}_N$
a	joint action, element of $\mathcal{A}$
a_{-i}	actions of all the agents except agent i, $(a_1,a_2,...,a_{i-1},a_{i+1},...,a_N)$
$\Delta(\mathcal{X})$	set of probability distributions over a finite set $\mathcal{X}$
Δ	set of stochastic strategy profiles $\Delta(\mathcal{A}_1) \times \cdots \times \Delta(\mathcal{A}_N)$
π_i, π, π_{-i}	policies of agent i in $\Delta(\mathcal{A}_i)$, all the N agents in Δ, and all the agents except i in Δ_{-i} respectively
ϕ	temporal logical formula
$\mathcal{Q}$	set of finite state machine states
q	finite state machine state
δ	finite state machine transition function
F	set of accepting conditions of an FSM
$\mathcal{P}$	Set of atomic propositions
b^t	set of truth values for each atomic proposition, i.e., an element of $2^{\mathcal{P}}$
w	input word on which LTL statements are evaluated
L	Labeling function that maps $\mathcal{S}$ to $2^{\mathcal{P}}$
h^t	state history at time t, $(s^0,...,s^t)$
$\mathcal{S}_p$, T_p, s_p	state space, transition function, state in the product game defined in definition 2.
$\mathcal{F}_i$	Avoid set; set of product game states in which rule i has been broken
$\mathcal{R}_i$	set of states from which agent i can unilaterally avoid reaching $\mathcal{F}_i$
$\mathcal{G}_i(s_p)$	set of prudent actions for agent i at product state s_p
$\mathcal{B}_i(s_p)$	set of imprudent actions for agent i at product state s_p
p_i	likelihood of agent i taking an imprudent action
Q_i^π	action value function for agent i under policy profile π
V_i^π	value function for agent i under policy profile π
$\Delta_i^G(s_p)$	set of stochastic strategies for agent i for which agent i takes an imprudent action with probability p_i at state s_p
R_i^{reach}, π_i^{reach}, V_i^{reach}	reward function, optimal policy, optimal value function, for agent i associated with the reachability MDP in theorem 1
Δ_{-i}^R	set of coordinated adversarial strategies that obey the prior over actions against agent i
Δ_G	$\Delta_1^G \times \cdots \times \Delta_N^G$
$V_i^\star$, $\pi_i^\star$	robust value functions and policies

APPENDIX I
GLOSSARY

A glossary of all the conventions and symbols used in this paper is included above.

APPENDIX II
DEFINING THE "BEFORE" OPERATOR IN LTL.

Many traffic rules and conventions specify certain events should happen in the right order, if they are to occur at all. An ability to express a requirement that one event should occur *before* another is indispensable to properly encode real-world traffic rules like the FIFO principle at a stop-signed intersection. We show such an operator can directly be constructed from the base operators $\mathbf{U}$, $\mathbf{X}$ in LTL syntax (see [27] for their definition). However, some nuance exists in the definition of a "before" operator: For example, should an operator that requires "x before y" allow x and y to occur at the same instant or not? We define both a strict and a loose "before" operator to express either requirement.

Definition 4. *For two* LTL *formulae x and y, we define the* loose before *operator as $x\mathbf{LB}y := (\neg y)\mathbf{U}x$. If an input trace $h \models x\mathbf{LB}y$, we say x happens loosely before y.*

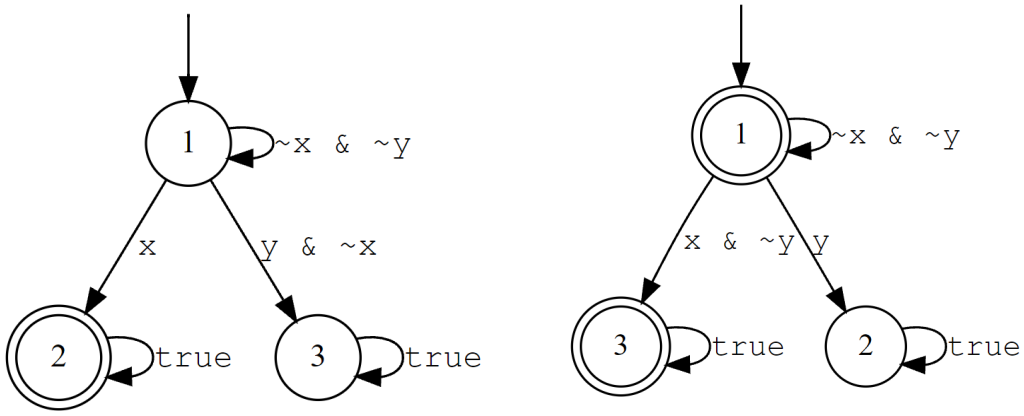

Fig. 3. Left: Finite state machine associated with the loose before operator. Right: FSM associated with the strict before operator. Initial state is indicated with the arrow from the top, accepting conditions are represented with doubly circled nodes.

Definition 5. *For two* LTL *formulae x and y, we define the* strict before *operator as $x\mathbf{SB}y := \neg(y\mathbf{LB}x) = \neg((\neg x)\mathbf{U}y)$. If an input trace $h \models x\mathbf{SB}y$, we say x happens strictly before y.*

We illustrate the behavior of the loose and strong before operators using their corresponding finite state machines, shown in Fig. 3. Besides the fact that both operators encode the basic idea that x should occur before y, it is apparent that the *loose before* operator

1) allows x and y to be satisfied for the first time simultaneously,
2) requires x to be satisfied at some point in time.

In contrast, the *strict before* operator

1) does not require x or y to be satisfied at some point in time,
2) requires x to be satisfied strictly before y is satisfied for the first time.

Since the *strict before* operator is accepting by default and disallows events happening simultaneously, it presents a useful tool in modelling real-world traffic rules that satisfy our desiderata.

Appendix III
Prudent and Imprudent actions

Lemma 1. *If $s_p \in \mathcal{R}_i$, then there exists an action $a_i \in \mathcal{A}_i$ so that $T_p(s'_p|s_p,a_i,a_{-i}) = 0$ for all $s'_p \in \mathcal{S}_p \setminus \mathcal{R}_i$ and $a_{-i} \in \mathcal{A}_{-i}$. In other words, there exists an action a_i so that $\mathrm{Prob}(s'_p \in \mathcal{R}_i) = 1$.*

Proof: Since s_p is in $\mathcal{R}_i$, there exists a policy π_i so that $\mathcal{F}_i$ is avoided for all time for all $\pi_{-i} \in \Pi_{-i}$ almost surely. Therefore, if agent i starts from s_p, takes the action $a_i = \pi_i(s_p)$, and transitions into s'_p, then $s'_p \notin \mathcal{F}_i$ and agent i can avoid $\mathcal{F}_i$ from s'_p by continuing to act according to π_i. Therefore, there exists a policy so that $\mathcal{F}_i$ is avoided from s'_p, implying that $s'_p \in \mathcal{R}_i$.

Theorem 3. *Define the set of* prudent*, or* good*, actions at each state $s_p \in \mathcal{S}_p$ for each agent $i = 1,...,N$ as the set*

$$\mathcal{G}_i(s) := \{a \in \mathcal{A}_i : T_p(s'_p|s_p,a_i,a_{-i}) = 0 \ \forall s'_p \in \mathcal{S}_p \setminus \mathcal{R}_i, \ a_{-i} \in \mathcal{A}_{-i}\}, \tag{10}$$

and let the set of imprudent*, or* bad *actions be $\mathcal{B}_i(s_p) := \mathcal{A}_i \setminus \mathcal{G}_i(s_p)$. Then, for a policy $\pi_i \in \Pi_i$, it holds that $\mathrm{Prob}(s_p^t \in \mathcal{F}_i | s_p^0 = s_p) = 0$ for all $t \geq 0$ and $\pi_{-i} \in \Pi_{-i}$ if and only if the initial condition $s_p^0 \in \mathcal{R}_i$ and $\pi_i(s_p^t) \in \mathcal{G}_i(s_p^t)$ for all $t \geq 0$.*

Proof: We first show the "only if" direction. If $s_p^t \in \mathcal{R}_i$, then lemma 1 implies that $\mathcal{G}_i(s_p^t)$ is not empty. By definition, if $s_p^t \in \mathcal{R}_i$, taking $\pi_i(s_p^t) \in \mathcal{G}_i(s_p^t)$ then implies that $s_p^{t+1} \in \mathcal{R}_i$. Moreover, $\mathcal{F}_i \cap \mathcal{R}_i = \emptyset$, so if $s_p^0 \in \mathcal{R}_i$, then the unilateral policy π_i for agent i guarantees $\mathcal{F}_i$ is avoided by induction.

Now we show the other direction. Assume that under a policy $\pi_i \in \Pi_i$, the state never enters $\mathcal{F}_i$ from some initial condition for any opponent policy $\pi_{-i} \in \Pi_{-i}$. We proceed by contradiction: Suppose that there exists some $t \geq 0$ and some realizable trajectory history $h_p^t \in \mathcal{S}_p^t$ under π_i from s_p^0 for which $a_i^t = \pi_i(s_p^t) \notin \mathcal{G}_i(s_p^t)$. Then, by the definition of $\mathcal{G}_i(s_p^t)$, it holds that $\mathrm{Prob}(s_p^{t+1} \notin \mathcal{R}_i) > 0$ for some $a_{-i} \in \mathcal{A}_{-i}$. By the definition of $\mathcal{R}_i$, there are no policies in Π_i that can guarantee that $\mathcal{F}_i$ is avoided for an initial condition in $\mathcal{S}_p \setminus \mathcal{R}_i$ for an arbitrary $\pi_{-i} \in \Pi_{-i}$. Therefore, there then exists some $k \geq 0$ and $\pi_{-i} \in \Pi_{-i}$ so that $\mathrm{Prob}(s_p^{t+k+1} \in \mathcal{F}_i | s_p^0) > 0$ under the policy π_i. This contradicts our premise, proving the theorem.

Appendix IV
Proof of Theorem 1

Theorem 4. *Define the MDP $(\mathcal{S}_p, T_i, \mathcal{A}_i, R_i^{\mathrm{reach}}, \gamma)$ as a reachability problem for each agent $i = 1,...,N$, with*

$$R_i^{\mathrm{reach}}(s_p,a_i) = \begin{cases} -1 & \text{if } s_p \in \mathcal{F}_i \\ 0 & \text{else} \end{cases} \tag{11}$$

and $\gamma \in (0,1)$. *Let* π_i^{reach}, V_i^{reach} *be the optimal policy and value function associated with this MDP. Then, it holds that*

$$\mathcal{R}_i = \{s_p \in \mathcal{S}_p : V_i^{\text{reach}}(s_p) = 0\}. \tag{12}$$

Proof: First, notice that for $(s_p', s_p, a_i) \in \mathcal{S}_p \times \mathcal{S}_p \times \mathcal{A}_i$ it holds that

$$T_i(s_p'|s_p, a_i) = 0 \iff T(s_p'|s_p, a_i, a_{-i}) = 0 \quad \forall a_{-i} \in \mathcal{A}_{-i}. \tag{13}$$

Now, we prove the "only if" direction. Let $\hat{\mathcal{R}}_i = \{s_p \in \mathcal{S}_p : V_i^{\text{reach}}(s_p) = 0\}$. Since

$$V_i^{\text{reach}}(s_p) = R_i^{\text{reach}}(s_p, \pi_i^{\text{reach}}(s_p)) + \gamma \sum_{s_p' \in \mathcal{S}_p} T_i(s_p'|s_p, \pi_i^{\text{reach}}(s_p)) V_i^{\text{reach}}(s_p')$$

and $R_i^{\text{reach}}(s_p, a_i) \leq 0$, it follows that $V_i^{\text{reach}}(s_p) = 0$ if and only if $R_i^{\text{reach}}(s_p, \pi_i^{\text{reach}}(s_p)) = 0$ and $V_i^{\text{reach}}(s_p') = 0$ for all $s_p' \in \mathcal{S}_p$ for which $T_i(s_p'|s_p, \pi_i^{\text{reach}}(s_p)) > 0$. Therefore, for a state $s_p \in \hat{\mathcal{R}}_i$, it holds that $T_i(s_p'|s_p, \pi_i^{\text{reach}}(s_p)) = 0$ for all $s_p' \in \mathcal{S}_p \setminus \hat{\mathcal{R}}_i$. Using (13), it follows that for $s_p \in \hat{\mathcal{R}}_i$, it holds that $T_p(s_p'|s_p, \pi^{\text{reach}}(s_p), a_{-i}) = 0$ for all $s_p' \in \mathcal{S}_p \setminus \hat{\mathcal{R}}_i$ and $a_{-i} \in \mathcal{A}_{-i}$. Since $\hat{\mathcal{R}}_i \cap \mathcal{F}_i = \emptyset$, we therefore have that if $s_p \in \hat{\mathcal{R}}_i$, there exists a policy such that $\text{Prob}_{T_p}(s_p^t \in \mathcal{F}_i | s_p^0 = s_p) = 0$ for all $t \geq 0$ and $\pi_{-i} \in \Pi_{-i}$ by induction. Therefore $\hat{\mathcal{R}}_i \subseteq \mathcal{R}_i$.

For the other direction, assume that $s_p \in \mathcal{R}_i$. Therefore, there exists a policy π_i so that $\text{Prob}_{T_p}(s_p^t \in \mathcal{F}_i | s_p^0 = s_p) = 0$ for all $t \geq 0$ and $\pi_{-i} \in \Pi_{-i}$. By theorem 3, this implies that for $s_p \in \mathcal{R}_i$, it holds that $\pi_i(s_p) \in \mathcal{G}_i(s_p)$. Therefore, we have that $T_p(s_p'|s_p, \pi_i(s_p), a_{-i}) = 0$ for all $s_p' \in \mathcal{S}_p \setminus \mathcal{R}_i$ and $a_{-i} \in \mathcal{A}_{-i}$. Equation (13) then implies that $T_i(s_p'|s_p, \pi_i(s_p)) = 0$ for all $s_p' \in \mathcal{S}_p \setminus \mathcal{R}_i$. Therefore, if the initial condition $s_p \in \mathcal{R}_i$, the policy π_i guarantees that $s_p^t \in \mathcal{R}_i$ for all $t \geq 0$ under the randomized dynamics $T_i(s_p'|s_p, \pi_i(s_p))$ by induction. This implies that the value function under π_i and the randomized dynamics T_i satisfies $V_i^{\pi}(s_p) = 0$. Since we assumed $V_i^{\text{reach}}(s_p) \in [\frac{-1}{1-\gamma}, 0]$ is the optimal value function of the reachability problem, it holds that $0 \geq V_i^{\text{reach}}(s_p) \geq V_i^{\pi}(s_p) = 0$. Therefore, we have that $\mathcal{R}_i \subseteq \hat{\mathcal{R}}_i$, proving the result.

Appendix V
Extended Discussion on Cautious Games Without Dynamics

To build out the machinery to handle stochastic games with rule-breaking likelihoods, we discuss how the Nash equilibrium concept extends to single-shot cautious games without dynamics (i.e, the stage games of the Markov Game). For such static games we drop the state dependence of reward functions R_i. In our setting, we know the likelihood of agent i taking an action in $\mathcal{B}_i$ is p_i a priori, so we say that such static games have a *prior over actions*.

Clearly, the set of mixed profiles that satisfy the prior over actions, defined as $\Delta_i^G := \{\mathbf{z}_i \in \Delta(\mathcal{A}_i) : \text{Prob}(a_i \in \mathcal{G}_i(s_p)) = 1 - p_i\}$ and, by extension, $\Delta_G := \prod_{i=1}^N \Delta_i^G$ are convex. The prior over actions indicates some partial knowledge of the behavior of other agents, such as an estimate of the likelihood that a road user will act to violate a traffic rule. We emphasize that the equilibria of the game with prior over actions are different from the normal-form stochastic equilibria of a game: There need not exist any mixed Nash equilibria to the game formed by action sets $\{\mathcal{A}_i\}_{i=1}^N$ and reward functions $\{R_i\}_{i=1}^N$ that happen to satisfy the prior over actions, whereas it is trivial to show that an equilibrium for the constrained policies $\pi \in \Delta_G$ has to exist for a game with priors over actions by repeating Nash's classic fixed-point proof (see [47]), since Δ_G is a convex subset of Δ. Therefore, specifying games with priors over actions allows us to align the result of game theoretic analysis with partial observations on an opponent's behavior without modifying the incentive structures of the agents. Moreover, by selecting $p_i = 0$ and $\mathcal{G}_i = \mathcal{A}_i$, it should be readily apparent that games with priors over actions generalize normal-form games. Therefore, general games with priors over actions inherit at least the computational hardness of general-sum normal-form games (specifically, PPAD-completeness [25]), so we consider efficiently solving cautious games in a general setting intractable.

Because identifying equilibria for general-sum N-player games is a computationally intractable problem, we instead focus on computing worst-case, or robust, strategies and payoffs for each agent. To compute robust payoffs for agent i, we assume that all the other agents centrally coordinate their decisions to frustrate agent i, while still satisfying the prior over actions.

Definition 6. *For a game with priors over actions, let* $\Delta_{-i}^R = \{\mathbf{z}_{-i} \in \Delta(\mathcal{A}_{-i}) : \text{Prob}(a_j \in \mathcal{B}_j) = p_j \quad \forall j \neq i\}$. *Then, the robust payoff* $R_i^\star$ *and robust strategy* $\mathbf{z}_i^\star \in \Delta_i^G$ *for agent* $i \in \{1, \ldots, N\}$ *are the solution and optimizer of the (maxmin) problem*

$$R_i^\star = \max_{\mathbf{z}_i \in \Delta_i^G} \min_{\mathbf{z}_{-i} \in \Delta_{-i}^R} R_i(\mathbf{z}_i, \mathbf{z}_{-i}). \tag{14}$$

In Appendix VII we show that we can solve (14) using linear programming (LP) in two separate ways: one uses using LP duality, and one takes advantage of an interpretation involving p_i-biased coin tosses. Because Δ_{-i}^G parametrizes only strategy profiles where the opponents behave independently of each other and Δ_{-i}^R does not, we get a lower-bound certificate of performance when agent i plays strategy $\mathbf{z}_i^\star$. That is, $R_i^\star \leq R_i(\mathbf{z}_i^\star, \mathbf{z}_{-i})$ for all $\mathbf{z}_{-i} \in \Delta_{-i}^G \subseteq \Delta_{-i}^R$.

It is a classic result in game theory that robust strategies correspond to the unique Nash equilibrium for 2-player zero-sum games [48], [17]. This classic result also applies to games with priors over actions, we include a proof in Appendix VI.

Example: Rock-Paper-Scissors (RPS): We illustrate the utility of a prior over actions with a simple static example. Consider a game of rock-paper-scissors between two agents, the ego and the opponent. Each agent receives a payoff of 1 if it wins, 0 if the outcome is a stalemate, and -1 if it loses, a classic zero-sum game. As is shown in Fig. 4, the *unique* NE for rock-paper-scissors is for both players to uniformly randomize their strategies, yielding an expected payoff of 0

for each agent [48]. Suppose now that we play many RPS games in sequence and only observe our reward and whether the human opponent played scissors or not. We might find that they do not randomize their play exactly according to the zero-sum NE. For example, as shown in Fig. 4, we might observe over time that $\text{Prob}(a_2 = \text{scissors}) = \frac{1}{10}$.

Rather than identifying a payoff function that *explains* our opponent's behavior, resulting in a non zero-sum game that is in general computationally intractable to solve, we can *align* the structure of the game with observations of our opponent's behavior by specifying a game with a prior over actions. As shown in Fig. 4, specifying the prior allows us to quickly exploit partial knowledge of our actual opponent's behavior, as we quickly learn not to play rock, resulting in an expected payoff of .23 and a 41% chance of winning. In Fig. 4 we also compare our approach, which estimates the opponent's prior over actions and solves (14) to compute the policy online, with the multiplicative weights (MW) algorithm [25], a no-regret learning algorithm that updates a policy distribution online and is guaranteed to converge to the NE payoff in a zero-sum game. We note that the MW algorithm requires observing a_2 at each timestep, not just whether agent 2 played scissors or not. Fig. 4 shows that our approach allows us to learn optimal behavior faster than using the MW algorithm.

APPENDIX VI
2-PLAYER ZERO-SUM GAMES

Theorem 5. *Suppose we have a 2-player zero-sum game with priors over actions. That is, suppose $R_1(a) = -R_2(a) =: R(a)$ for all $a \in \mathcal{A}$. Then, there exists a unique Nash equilibrium $R^\star = R_1^\star = -R_2^\star$ that is attained when both players play max-min strategies $\mathbf{z}_1^\star$ and $\mathbf{z}_2^\star$, defined as*

$$\mathbf{z}_i^\star \in \underset{\mathbf{z}_i \in \Delta_i^G}{\text{argmax}} \min_{\mathbf{z}_j \in \Delta_j^G} R_i(\mathbf{z}_i, \mathbf{z}_j), \tag{15}$$

for $(i,j) \in \{(1,2),(2,1)\}$.

Proof: Suppose $\hat{\mathbf{z}}_1 \in \Delta_1^G$ and $\hat{\mathbf{z}}_2 \in \Delta_2^G$ form a Nash equilibrium. Since the game is zero sum, $R_1(\hat{\mathbf{z}}) = -R_2(\hat{\mathbf{z}}) =: \hat{R}$. Applying the definition of Nash equilibrium to agent 1, it holds that

$$\hat{R} = \max_{\mathbf{z}_1 \in \Delta_1^G} R(\mathbf{z}_1, \hat{\mathbf{z}}_2) \geq \min_{\mathbf{z}_2 \in \Delta_2^G} \max_{\mathbf{z}_1 \in \Delta_1^G} R(\mathbf{z}_1, \mathbf{z}_2). \tag{16}$$

Similarly, using the NE definition and the fact that $R_2(a) = -R_1(a) = -R(a)$, we have for agent 2 that

$$\hat{R} = \min_{\mathbf{z}_2 \in \Delta_2^G} R(\hat{\mathbf{z}}_1, \mathbf{z}_2) \leq \max_{\mathbf{z}_1 \in \Delta_1^G} \min_{\mathbf{z}_2 \in \Delta_2^G} R(\mathbf{z}_1, \mathbf{z}_2). \tag{17}$$

If we define the matrix $\mathbf{R} \in \mathbb{R}^{|\mathcal{A}_1| \times |\mathcal{A}_2|}$, with entries $[\mathbf{R}]_{ij} = R(a_1^i, a_2^j)$ for $a_1^i \in \mathcal{A}_1$ and $a_2^j \in \mathcal{A}_2$, then it holds that $R(\mathbf{z}_1, \mathbf{z}_2) = \mathbf{z}_1^\intercal \mathbf{R} \mathbf{z}_2$ for $(\mathbf{z}_1, \mathbf{z}_2) \in \Delta_G$. Therefore, by von Neumman's minimax theorem and convexity of the compact sets Δ_1^G and Δ_2^G, defining $R^\star$ as

$$R^\star := \min_{\mathbf{z}_2 \in \Delta_2^G} \max_{\mathbf{z}_1 \in \Delta_1^G} \mathbf{z}_1^\intercal \mathbf{R} \mathbf{z}_2 = \max_{\mathbf{z}_1 \in \Delta_2^G} \min_{\mathbf{z}_2 \in \Delta_1^G} \mathbf{z}_1^\intercal \mathbf{R} \mathbf{z}_2, \tag{18}$$

implies that $R^\star \leq \hat{R} \leq R^\star$. Hence, the game has a unique equilibrium attained when the agents play maxmin policies.

APPENDIX VII
ADVERSARIALLY ROBUST STRATEGIES FOR N-PLAYER GAMES

In this section, we show how to compute the robust value and policy associated with a game with priors over actions defined in definition 6 in two ways. Both methods show that the maxmin problem is a Linear Program.

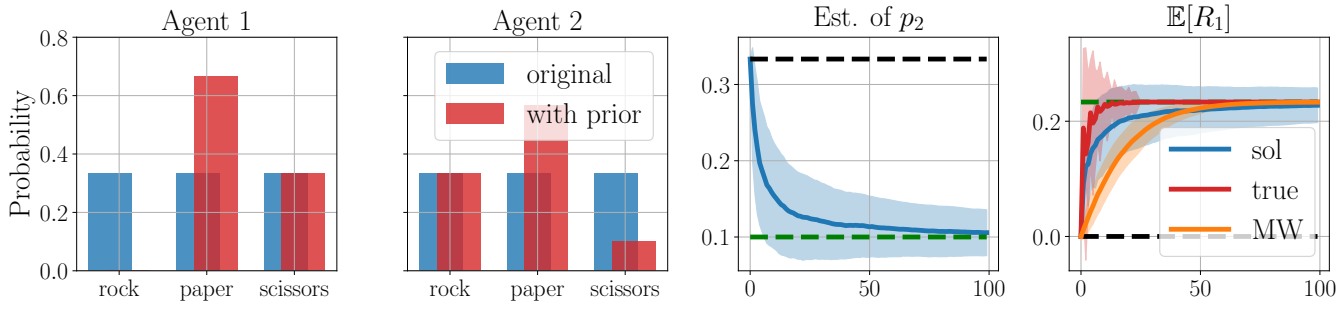

Fig. 4. Left two plots: NE policies for RPS for the original zero-sum game, and with a prior over actions on agent 2. Center right: Estimate of $\text{Prob}(a_2 = \text{scissors})$ vs. the no. of games played. Right: Expected reward of agent 1: sol indicates $R^\star$ from (14) under the current estimate, true compares the current policy to π_2, and MW compares a policy computed using the MW algorithm to π_2, all with 2σ bars.

1) Base Interpretation: From the definition of Δ^R_{-i}, we can write the robust problem (14) as

$$\begin{aligned}\max_{\mathbf{z}_i\in\Delta_i^G}\min_{\mathbf{z}_{-i}}&\mathbf{z}_i^\intercal\mathbf{R}_i\mathbf{z}_{-i}\\ \text{s.t.}\quad&\mathbf{z}_{-i}\in\Delta(\mathcal{A}_{-i}),\\ &\text{Prob}_{\mathbf{z}_{-i}}(a_j\in\mathcal{B}_j)=p_j\quad\forall j\neq i,\end{aligned}$$

where the matrix $\mathbf{R}_i\in\mathbb{R}^{|\mathcal{A}_i|\times|\mathcal{A}_{-i}|}$ has entries $[\mathbf{R}_i]_{kj}=R_i(a_i^k,a_{-i}^j)$ for $a_i^k\in\mathcal{A}_i$ and $a_{-i}^j\in\mathcal{A}_{-i}$. We will write the inner $\min$ as a $\max$ by taking the dual. First, define $\mathbf{b}^\intercal=[1,p_1,...,p_{i-1},p_{i+1},...,p_N]$. Also, let the matrix

$$\mathbf{F}_i=[\mathbf{f}_0\quad\mathbf{f}_1\quad...\quad\mathbf{f}_{i-1}\quad\mathbf{f}_{i+1}\quad...\quad\mathbf{f}_N]\in\mathbb{R}^{|\mathcal{A}_{-i}|\times N},$$

where $\mathbf{f}_0=\mathbb{1}_{|\mathcal{A}_{-i}|}$ is the one vector and the k'th entry of the vector $[\mathbf{f}_j]_k=1$ if $a_{-i}^k=(a_1,...,a_{i-1},a_{i+1},...,a_N)$ has $a_j\in\mathcal{B}_j$. Then we can rewrite the inner min as

$$\begin{aligned}\min_{\mathbf{z}_{-i}}&\underbrace{\mathbf{z}_i^\intercal\mathbf{R}_i}_{=:c^\intercal}\mathbf{z}_{-i}\\ \text{s.t.}\quad&\mathbf{z}_{-i}\geq 0\\ &\mathbf{F}_i^\intercal\mathbf{z}_{-i}=\mathbf{b}.\end{aligned}$$

The lagrangian of this LP is

$$\begin{aligned}\mathcal{L}(\mathbf{z}_{-i},\lambda,\nu)&=c^\intercal\mathbf{z}_{-i}+\lambda^\intercal(-\mathbf{z}_{-i})+\nu^\intercal(\mathbf{F}_i^\intercal\mathbf{z}_{-i}-\mathbf{b})\\ &=(c+\mathbf{F}_i\nu-\lambda)^\intercal\mathbf{z}_{-i}-\nu^\intercal\mathbf{b},\end{aligned}$$

from which it follows that the Lagrangian dual function is

$$g(\lambda,\nu):=\inf_{\mathbf{z}_{-i}}\mathcal{L}(\mathbf{z}_{-i},\lambda,\nu)=\begin{cases}-\nu^\intercal\mathbf{b} & \text{if } (c+\mathbf{F}_i\nu-\lambda)=0\\ -\infty & \text{else}\end{cases}.$$

Hence, the dual program $\max_{\lambda\geq 0,\ \nu}g(\lambda,\nu)$ can be rewritten as the LP

$$\begin{aligned}\max_{\lambda,\ \nu}&-\nu^\intercal\mathbf{b}\\ \text{s.t.}\quad&\lambda\geq 0\\ &c+\mathbf{F}_i\nu=\lambda,\end{aligned}$$

which is equivalent to

$$\begin{aligned}\max_{\nu}&-\nu^\intercal\mathbf{b}\\ \text{s.t.}\quad&\mathbf{R}_i^\intercal\mathbf{z}_i+\mathbf{F}_i\nu\geq 0.\end{aligned}$$

Substituting $v=-\nu$, we get

$$\begin{aligned}\max_{v}&\ v^\intercal\mathbf{b}\\ \text{s.t.}\quad&\mathbf{R}_i^\intercal\mathbf{z}_i-\mathbf{F}_iv\geq 0.\end{aligned}$$

returning to the original problem, it follows that

$$\begin{aligned}R_i^\star=\max_{\mathbf{z}_i\in\Delta_i^G}\max_{v}&\ v^\intercal\mathbf{b}\\ \text{s.t.}\quad&\mathbf{R}_i^\intercal\mathbf{z}_i-\mathbf{F}_iv\geq 0\end{aligned}\tag{19}$$

which is a basic LP with decision variables $\mathbf{z}_i$ and ν. In addition, note that the optimal profile $\mathbf{z}_i^\star$ that maximizes (19) is also a maximimzer for the robust problem (14).

2) Random Agent Interpretation: The second method relies on the interpretation that a game-with priors over actions corresponds to a game where each agent first flips a coin to decide whether to play a prudent or imprudent action. To compute the robust value $R_i^\star$, we consider a coordinated agent for every outcome scenario of the coin tosses. To explicitly write this out, let the set $\mathcal{X}_j = \{(1-p_j,\mathcal{G}_j),(p_j,\mathcal{B}_j)\}$ collect the tuples representing the prudent and imprudent scenarios for each agent. In addition, let $\mathcal{X}_{-i} = \prod_{j\neq i}\mathcal{X}_j$ collect every opponent scenario for agent i. Then, for every scenario $x_{-i}\in\mathcal{X}_{-i}$, define the associated opponent action set as $\mathcal{U}_{-i}(x_{-i}) = \prod_{(w_j,\mathcal{U}_j)\in x_j \forall j\neq i}\mathcal{U}_j$. This action set occurs with likelihood $w_{x_{-i}} = \prod_{(w_j,\mathcal{U}_j)\in x_j\forall j\neq i} w_j$. By representing the coordinated opponent strategy using distributions conditioned over each scenario, it follows that

$$\Delta_{-i}^R = \left\{ \{w_{x_{-i}}\mathbf{z}_{x_{-i}}\}_{x_{-i}\in\mathcal{X}_{-i}} : \mathbf{z}_{x_{-i}}\in\Delta(\mathcal{U}_{x_{-i}}) \quad \forall x_{-i}\in\mathcal{X}_{-i} \right\}.$$

It then follows that the robust problem

$$\max_{\mathbf{z}_i\in\Delta_i^G}\min_{\mathbf{z}_{-i}\in\Delta_{-i}^R} R_i(\mathbf{z}_i,\mathbf{z}_{-i})$$

is equivalent to

$$\begin{aligned}\max_{\mathbf{z}_i}\min_{\mathbf{z}_{x_{-i}}} & \sum_{x_{-i}\in\mathcal{X}_{-i}} w_{x_{-i}} R_i(\mathbf{z}_i,\mathbf{z}_{x_{-i}}) \\ & \text{s.t.} \quad \mathbf{z}_i\in\Delta_i^G \\ & \quad \mathbf{z}_{x_{-i}}\in\Delta(\mathcal{U}_{-i}(x_{-i})) \quad \forall x_{-i}\in\mathcal{X}_{-i}.\end{aligned}$$

Now, we define the matrices $\mathbf{R}_i^{g,x_{-i}}\in\mathbb{R}^{|\mathcal{G}_i|\times|\mathcal{U}_{-i}(x_{-i})|}$ and $\mathbf{R}_i^{b,x_{-i}}\in\mathbb{R}^{|\mathcal{B}_i|\times|\mathcal{U}_{-i}(x_{-i})|}$ with k,l'th entries as $R_i(a_i^k,a_{-i}^l)$ for $a_i^k\in\mathcal{G}_i$, $a_{-i}^l\in\mathcal{U}_i(x_{-i})$ and $a_i^k\in\mathcal{B}_i$, $a_{-i}^l\in\mathcal{U}_i(x_{-i})$ respectively. Then, applying the epigraph trick, it follows that we can write the robust problem as

$$\begin{aligned}\max_{\mathbf{x}_i,\mathbf{y}_i,t_{x_{-i}}} & \sum_{x_{-i}\in\mathcal{X}_{-i}} w_{x_{-i}} t_{x_{-i}} \\ \text{s.t.} \quad & \mathbf{x}_i\in\Delta(\mathcal{G}_i), \quad \mathbf{y}_i\in\Delta(\mathcal{B}_i), \quad t_{x_{-i}}\in\mathbb{R} \quad \forall x_{-i}\in\mathcal{X}_{-i}, \\ & t_{x_{-i}} \leq (1-p_i)\mathbf{R}_i^{g,x_{-i}\intercal}\mathbf{x}_i + p_i\mathbf{R}_i^{b,x_{-i}\intercal}\mathbf{y}_i \quad \forall x_{-i}\in\mathcal{X}_{-i}.\end{aligned} \tag{20}$$

APPENDIX VIII
ROBUST VALUE ITERATION FOR CAUTIOUS GAMES

Here, we present the robust value iteration algorithm for cautious stochastic games. The algorithm returns the robust value function $V_i^\star$ for a particular agent i (the ego), from which the robust action-value function $Q_i^\star$ and consequently, the robust implicit policy $\pi_i^\star$ can be constructed. In robust value iteration, we iteratively update a Q_i-table for all states and actions using the value function computed at the previous iteration. Then, we update the value function for each agent by solving a worst-case solution to the stage game as defined in Def. 6. This robust stage game considers the worst the opponents can do while still satisfying the imprudent action likelihoods $p_1,...,p_{i-1},p_{i+1},...,p_N$. We define the algorithm below, and prove its properties in this section.

Algorithm 1: Cautious Markov Game Robust Value Iteration

Given: CMG $(\mathcal{S}_p,\gamma,T_p,\{R_i,\mathcal{G}_i,\mathcal{B}_i,p_i\}_{i=1}^N)$, ego agent i
$V_i^0(s_p)\leftarrow 0 \quad \forall s_p\in\mathcal{S}_p$ and $k\leftarrow 1$
while not converged **do**
  for $s_p\in\mathcal{S}_p, a\in\mathcal{A}$ **do**
    $\mathcal{Q}_i^k(s_p,a)\leftarrow R_i(s_p,a)+\gamma\sum_{s_p'\in\mathcal{S}_p} T_p(s_p'|s_p,a)V_i^{k-1}(s_p')$
  for $s_p\in\mathcal{S}_p$ **do**
    $V_i^k(s_p)\leftarrow$ solution to (14) using $\mathcal{Q}_i^k(s_p,\cdot)$
  $k\leftarrow k+1$
return $\mathcal{Q}_i^\star(s_p,a)$, $V_i^\star(s_p)$

Lemma 2. *Algorithm 1 converges to a unique fixed point $V_i^\star$ for each agent $i=1,...,N$.*

Proof: The proof presented is essentially an analogous result to that of the original 2-player zero-sum game due to [18]. Moreover, it can also be shown using robust dynamic programming theory as presented in [32]. For any function $V:\mathcal{S}_p\to\mathbb{R}$, define $\|V\|_\infty = \max_{s_p\in\mathcal{S}_p}|V(s_p)|$ and let $\mathrm{CB}[V_i](s_p)$ denote the result at state s_p of applying one iteration of

the value iteration algorithm 1 to V_i for agent $i \in \{1,...,N\}$. Then, for any two $V_i, \hat{V}_i$ and $s_p \in \mathcal{S}_p$, it holds that

$$\begin{aligned}
\mathrm{CB}[V_i](s_p) &= \max_{\pi_i(s_p)\in\Delta_i^G(s_p)} \min_{\pi_{-i}(s_p)\in\Delta_{-i}^R(s_p)} Q_i(s_p,\pi_i(s_p),\pi_{-i}(s_p))) \\
&= \max_{\pi_i(s_p)\in\Delta_i^G(s_p)} \min_{\pi_{-i}(s_p)\in\Delta_{-i}^R(s_p)} \mathbb{E}_{a\sim(\pi_i(s_p),\pi_{-i}(s_p))}\big[R_i(s_p,a)+\gamma \sum_{s_p\in\mathcal{S}_p} T_p(s_p'|s_p,a)V_i(s_p')\big] \\
&= \max_{\pi_i(s_p)\in\Delta_i^G(s_p)} \min_{\pi_{-i}(s_p)\in\Delta_{-i}^R(s_p)} \mathbb{E}_{a\sim(\pi_i(s_p),\pi_{-i}(s_p))}\big[R_i(s_p,a)+\gamma\mathbb{E}_{s_p'}[V_i(s_p')|a]\big] \\
&= \max_{\pi_i(s_p)\in\Delta_i^G(s_p)} \min_{\pi_{-i}(s_p)\in\Delta_{-i}^R(s_p)} \mathbb{E}_{a\sim(\pi_i(s_p),\pi_{-i}(s_p))}\big[R_i(s_p,a)+\gamma\mathbb{E}_{s_p'}[\hat{V}(s_p')|a]+\gamma\mathbb{E}_{s_p'}[V_i(s_p')-\hat{V}_i(s_p')|a]\big] \\
&\leq \mathrm{CB}[\hat{V}_i](s_p)+ \max_{\pi_i(s_p)\in\Delta_i^G(s_p)\pi_{-i}(s_p)\in\Delta_{-i}^R(s_p)} \gamma\mathbb{E}_{s_p'}[V_i(s_p')-\hat{V}_i(s_p')] \\
&\leq \mathrm{CB}[\hat{V}_i](s_p)+\gamma\|V_i-\hat{V}_i\|_\infty.
\end{aligned}$$

By applying a symmetric argument to $\mathrm{CB}[\hat{V}_i](s_p)$, we get that $|\mathrm{CB}[V_i](s_p)-\mathrm{CB}[\hat{V}_i](s_p)| \leq \gamma\|V_i-\hat{V}_i\|_\infty$. Therefore, we have that $\mathrm{CB}[\cdot]$ is a contraction operator; i.e., that

$$\|\mathrm{CB}[V_i]-\mathrm{CB}[\hat{V}_i]\|_\infty \leq \gamma\|V_i-\hat{V}_i\|_\infty,$$

since $\gamma \in (0,1)$. By the contractive fixed point theorem [32], contractivity of $\mathrm{CB}[\cdot]$ implies that algorithm 1 converges to a unique fixed point $V_i^\star$ for which

$$V_i^\star(s_p) = \max_{\mathbf{z}_i\in\Delta_i^G(s_p)\mathbf{z}_{-i}\in\Delta_{-i}^R(s_p)} \min Q_i^\star(s_p,\mathbf{z}_i,\mathbf{z}_{-i}), \quad \forall s_p \in \mathcal{S}_p.$$

Lemma 3. *For a 2-player zero-sum Cautious Stochastic game, a policy profile $\pi^\star$ such that $\pi^\star(s_p) \in \Delta_G(s_p)$ for all $s_p \in \mathcal{S}_p$, associated with value functions $V_1^\star(s_p)$ and $V_2^\star(s_p)$, constitutes a Nash equilibrium if and only if both players play robust (maxmin) strategies. That is, if and only if*

$$\pi_i^\star(s_p) \in \operatorname*{argmax}_{\pi_i(s_p)\in\Delta_i^G(s_p)\pi_j(s_p)\in\Delta_j^G(s_p)} \min Q_i^\star(s_p,\pi_i(s_p),\pi_j(s_p)) \quad \forall s_p \in \mathcal{S}_p,$$

for $(i,j) \in \{(1,2),(2,1)\}$. Moreover, for each state $s_p \in \mathcal{S}_p$, it holds that $V_1^\star(s_p) = -V_2^\star(s_p)$.

Proof: Since the game is zero sum, it follows that for any policy profile π with $\pi(s_p) \in \Delta_G(s_p)$ for all $s_p \in \mathcal{S}_p$, it holds that $V_1^\pi(s_p) = -V_2^\pi(s_p)$ for every $s_p \in \mathcal{S}_p$. Therefore, it follows that the Q-functions for each agent associated with any such policy π satisfy $Q_1^\pi(s_p,a) = -Q_2^\pi(s_p,a)$ for all $s_p \in \mathcal{S}_p$ and $a \in \mathcal{A}$. Note that a policy profile $\pi^\star$ is a Nash equilibrium if and only if $\pi^\star(s_p)$ constitutes a Nash equilibrium for the subgame at each state $s_p \in \mathcal{S}_p$ with payoffs $Q_i^\star(s_p,a)$ for each agent $i = 1,2$ [17]. Since the game is zero-sum, we have that $Q_1^\star(s,a) = -Q_2^\star(s,a)$, so these subgames are zero-sum 2-player games with priors over actions. Theorem 5 therefore gives us that a Markov policy profile for the 2-player zero-sum CMG constitutes a Nash equilibrium if and only if both agents act according to maxmin strategies, and that $V_1^\star(s_p) = -V_2^\star(s_p)$ for each $s_p \in \mathcal{S}_p$.

Theorem 6. *For a 2-player Cautious Stochastic Game, Algorithm 1 converges to the* unique *Nash equilibrium value function $V^\star(s_p) = V_1^\star(s_p) = -V_2^\star(s_p)$ of the game, attained when both agents play their robust (maxmin) implicit policies. That is, when*

$$\pi_i^\star(s_p) \in \operatorname*{argmax}_{\pi_i(s_p)\in\Delta_i^G(s_p)\pi_j(s_p)\in\Delta_j^G(s_p)} \min Q_i^\star(s_p,\pi_i(s_p),\pi_j(s_p)) \quad \forall s_p \in \mathcal{S}_p,$$

for $(i,j) \in \{(1,2),(2,1)\}$.

Proof: Since the game has 2 players, by Lemma 2, we have that Algorithm 1 converges to a unique fixed point $V_i^\star$ satisfying

$$V_i^\star(s_p) = \max_{\mathbf{z}_i\in\Delta_i^G(s_p)z_j\in\Delta_j^G(s_p)} \min Q_i^\star(s_p,\mathbf{z}_i,\mathbf{z}_{-i}), \quad \forall s_p \in \mathcal{S}_p, \tag{21}$$

for $(i,j) \in \{(1,2),(2,1)\}$. Moreover, since the game is zero-sum, Lemma 3 then yields the result.

Theorem 7. *Let $V_i^\star$ be the output of algorithm 1 for agent i, with implicit maxmin policy $\pi_i^\star$, i.e., a policy satisfying*

$$\pi_i^\star(s_p) \in \operatorname*{argmax}_{\pi_i(s_p)\in\Delta_i^G(s_p)\pi_{-i}(s_p)\in\Delta_{-i}^R(s_p)} \min Q_i^\star(s_p,\pi_i(s_p),\pi_{-i}(s_p)) \quad \forall s_p \in \mathcal{S}_p.$$

Let V_i be the value function for agent i under $\pi_i^\star$ and any opponent policies π_{-i} such that $\pi_{-i}(s_p) \in \Delta_{-i}^G(s_p)$ at each $s_p \in \mathcal{S}_p$. Then it holds for any $s_p \in \mathcal{S}_p$ that $V_i^\star(s_p) \leq V_i(s_p)$.

Proof: Since $\pi_i^\star$ is the implicit maxmin policy associated with $V_i^\star$, lemma 2 gives us that

$$V_i^\star(s_p) = \min_{\pi_{-i}(s_p)\in\Delta_{-i}^R(s_p)} Q_i^\star(s_p,\pi_i^\star(s_p),\pi_{-i}(s_p)) \quad \forall s_p \in \mathcal{S}_p. \tag{22}$$

Next, let π_{-i} be any coordinated opponent policy so that $\pi_{-i}(s_p) \in \Delta^R_{-i}(s_p)$ for each $s_p \in \mathcal{S}_p$, and let V_i denote the cumulative expected discounted reward for agent i under the policies $\pi_i^\star$ and π_{-i}. By Bellman's principle of optimality, equation (22) then gives us that $V_i^\star(s_p) \leq V_i(s_p)$ for all $s_p \in \mathcal{S}_p$. The result then follows from the fact that $\Delta^G_{-i}(s_p) \subseteq \Delta^R_{-i}(s_p)$ for all $s_p \in \mathcal{S}_p$.

APPENDIX IX
EXPERIMENTAL DETAILS

In this section, we present implementation details related to our simulated experiments using the CARLA simulator [44]. We first discuss how we construct a basic autonomy stack using the CARLA Python API that resembles the manner in which autonomous vehicles operate today: using low-level tracking controllers to track reference commands from high-level decision making. Next, we discuss how we discretized the simulation into an abstract MDP representation for high-level decision making which we used for learning and planning. Finally, we discuss implementation details for the stop-signed intersection and the T-junction experiments in §VI. **We emphasize that videos that illustrate our experiments are included in the supplementary material.**

Base Autonomy Stack in Carla: To control the vehicles in Carla we make use of a basic autonomy stack that assumes perfect state knowledge. We use the Python waypoint API to extract a map representation from the OpenDRIVE specifications of the Carla environments. We build a routing graph from the intersections in the map. Then, we construct simulations by setting start and goal locations on the map for each agent. We use a shortest-path algorithm on the routing graph to construct a reference trajectory of waypoints along the centerline of the lane for each agent. For low-level control, we use a PID controller to track the centerline of the lane (lateral control). The high-level decision making (the focus of this work) feeds a reference velocity command to another low-level PID controller to control the longitudinal speed of the vehicles. This base autonomy stack is identical for all the agents.

High-Level Decision-Making: We make use of the MDP formalism in this work, which requires discrete state and action sets. Therefore, to construct high-level decision making policies, we first discretize the action set of each agent into $\mathcal{A}_i = \{\text{slow down}, \text{maintain speed}, \text{speed up}\}$. These actions increment or decrement a fixed velocity reference command chosen from $\{0\text{km/h}, 5\text{km/h}, 10\text{km/h}\}$, which is held constant for a fixed-time duration and tracked by the PIDs. The velocity references are bounded, i.e., selecting $\text{speed up}\}$ as an action when the reference is 10km/h maintains the reference at 10km/h. For the states $s_i = (x_i, v_i)$ of each individual agent, we take x_i as the distance from the starting point along the reference trajectory of the vehicle and v_i as its velocity. The state space then consists of the product set of the state space of each agent. For the purpose of decision making, we take the continuous states used in the simulation and discretize them as follows: we first divide the reference trajectory into a fixed number of grid cells to represent x_i as a finite number of positions. We take $v_i \in \{0\text{km/h}, 5\text{km/h}, 10\text{km/h}\}$. Then, we adjust the timestep between which the high-level decision making stack updates its reference commands so that an integer number of grid cells are traversed for the discretized v_i. To convert a continuous state into a discrete one, we compute which grid cell the agent occupies, and which discrete velocity is closest to its actual velocity. The discretized dynamics are simple to represent as single integrators ($x_i' = x_i + v_i dt$), but we add randomness to the position transitions by randomly incrementing or decrementing the deterministic grid position transition with probability $1/3$ when we compute policies to account for innacuracies of the discretization. The low-level decision making PIDs are updated every $.05$ seconds, the high-level reference commands are updated at the slower timestep based on the number of grid cells used.

Data Collection and Learning: In our experiments, we evaluate the decision making capacity of an autonomous vehicle that uses robust solutions to the cautious Markov game to make decisions against an arbitrary opponent. This requires us to estimate the likelihood with which the opponents take imprudent actions. To collect data to learn the likelihood at which the opponents take imprudent actions, we use a conservative policy that does not take the rules into account: we compute an adversarially robust policy for the ego on the base MDP $(\mathcal{S}, \{\mathcal{A}_i\}_{i=1}^N, T)$ that only captures the position-velocity dynamics of the agents. This worst-case policy always stops to let the opponent pass. We count the number of imprudent actions the opponents take at each observed state and estimate $p_2(\mathcal{S}_p)$ as a product-state dependant quantity from 20 trajectories for each experiment, and interpolate p_2 on states where we have no data. All our simulations run for 30 seconds (simulated time).

Stop Signed Intersection: We illustrate how to construct traffic rules using the stop-signed intersection experiment. To construct the rules, we define three atomic propositions, which we evaluate by checking whether the vehicles are inside a set of bounding boxes in the simulations:

$$A_i(s) = x_i \text{ has arrived at the stop sign}$$
$$B_i(s) = x_i \text{ occupies the intersection}$$
$$C_i(s) = x_i \text{ has crossed the intersection.}$$

The agents should satisfy the FIFO traffic convention, which states that whichever agent arrived first, should cross the street first. We write this as

$$\phi_i = (A_j \ \mathbf{SB} \ A_i) \rightarrow (C_j \ \mathbf{SB} \ B_i),$$

for $(i,j) \in \{(1,2),(2,1)\}$. Which states that "if agent j arrives at the intersection before agent i, then agent j should have crossed the intersection before agent i enters the intersection". To benchmark our algorithms, we start the simulation from an initial condition where the ego agent is closer to the intersection, such that we benchmark when the ego anticipates the opponent to yield or not. We set the discount factor $\gamma = 0.8$ and use a reward of 5 when the agent is at its goal state, and a penalty of 5 for a collision.

Merging at a T-Junction: In the T-junction experiment, we initialize the simulations so that both agents would collide with each other if they continued at their initial speed of 5 km/h. The ego agent does not have a traffic rule in this experiment. The opponent needs to follow a yielding rule to merge into the ego vehicle's lane. To decide when the opponent needs to yield, we take the current positions and velocities of the agents and extrapolate the positions of the agents into the future assuming they maintain their current speed. If they would collide along their current trajectory (defined as reaching within 2 car-lengths distance of each other), then the opponent must yield for the ego. This is defined using the rule

$$\phi_2 = \mathrm{Col} \Longrightarrow C_1 \ \mathbf{SB} \ B_2,$$

where C_1 indicates whether the ego has crossed the intersection and B_2 is an indicator on whether the opponent is occupying the intersection. We use a reward of 2 when the agent is at 10 km/h and a penalty of 10 for a collision.