Encoding Defensive Driving as a Dynamic Nash Game

Chih-Yuan Chiu,* David Fridovich-Keil,* and Claire J. Tomlin

Abstract—Robots deployed in real-world environments should operate safely in a robust manner. In scenarios where an “ego” agent navigates in an environment with multiple other “non-ego” agents, two modes of safety are commonly proposed—adversarial robustness and probabilistic constraint satisfaction. However, while the former is generally computationally-intractable and leads to overconservative solutions, the latter typically relies on strong distributional assumptions and ignores strategic coupling between agents.

To avoid these drawbacks, we present a novel formulation of robustness within the framework of general sum dynamic game theory, modeled on defensive driving. More precisely, we inject the ego’s cost function with an adversarial phase, a time interval during which other agents are assumed to be temporarily distracted, to robustify the ego agent’s trajectory against other agents’ potentially dangerous behavior during this time. We demonstrate the effectiveness of our new formulation in encoding safety via multiple traffic scenarios.

I. INTRODUCTION

Decision-making modules in autonomous systems must meet safety and robustness criteria before they are deployed in real-world settings with uncertain or unknown environments. Scenarios such as autonomous driving, in which an “ego” agent must interact with other, possibly non-cooperative “non-ego” agents, are of particular interest. These scenarios are naturally modeled using dynamic game theory, which describes the evolution of each agent’s state according to their minimization of a cost function. Each player’s cost function depends on the control strategies of that player, as well as the shared state of all the players.

To ensure safe and efficient operation of the ego agent in such multi-agent settings, existing methods formulate safety in the following two ways. Adversarial robustness methods, such as Hamilton-Jacobi-Isaacs (HJI) equation-based reachability theory, aim to generate trajectories that would ensure the safety of the ego agent despite worst-case behavior of all other agents [1–3]. Another commonly proposed methodology involves probabilistic constraint satisfaction [4, 5]. Here, algorithms attempt to bound the probability that the ego agent’s trajectory becomes unsafe. Unfortunately, each of these approaches carries significant drawbacks. HJI methods so-called “curse of dimensionality,” with computational cost increasing exponentially in the dimension of the state. Meanwhile, probabilistic constraint satisfaction encodes safety via distributional assumptions, but does not allow the ego player to anticipate more specific patterns of adversarial interactions with non-ego agents.

This paper addresses these issues by presenting a third, novel formulation of robustness in multi-agent games, illustrated in Fig. 1. Here, the ego agent imagines that all other agents behave adversarially during an initial time frame before resuming “normal” cooperative behavior. In this example, an ego vehicle assumes that the non-ego, oncoming vehicle behaves adversarially for an initial period of time, which results in its swerving into the ego’s lane. After the adversarial time interval expires, the non-ego agent is assumed to return to its own lane, and resume cooperative behavior for the remainder of the time horizon.

Fig. 1: (Top) To encode robustness into its safety guarantees, the ego agent imagines that all other agents behave adversarially during an initial time frame before resuming “normal” cooperative behavior. (Bottom) In this example, an ego vehicle assumes that the non-ego, oncoming vehicle behaves adversarially for an initial period of time, which results in its swerving into the ego’s lane. After the adversarial time interval expires, the non-ego agent is assumed to return to its own lane, and resume cooperative behavior for the remainder of the time horizon.
expected to resume “normal” cooperative behavior. This is modeled on the concept of defensive driving, wherein a driver on a busy road guards themselves against other drivers who, while momentarily distracted, may temporarily behave dangerously.

The rest of our paper is structured as follows. Sec. II presents related work on the use of HJI reachability theory and chance-constrained optimal control to encode adversarial and probabilistic robustness, respectively, as well as recent literature on iterative algorithms for solving dynamic games. Sec. III presents the mathematical formulation for the dynamic game that models the multi-agent interactions studied in our work. Sec. IV presents the main methodology in this paper, and formally introduces the concept of an adversarial time horizon. Sec. V demonstrates the spectrum of robustness which can be expressed in our formulation, compared with a purely cooperative game-theoretic approach, using multiple traffic scenarios. Sec. VI summarizes our contributions and discusses directions for future research.

II. RELATED WORK

This section reviews prior literature on several of the prevailing formulations of safety used in the design of multi-agent and uncertain systems—adversarial reachability, multi-agent forward reachability, and probabilistic constraint satisfaction. We compare and contrast these against the novel methodology in our work. We conclude this section with a brief summary of modern algorithms for iteratively solving dynamic games.

A. Adversarial Reachability

Adversarial reachability methods [1, 2, 6, 7] involve the construction of a zero-sum differential game between two agents, the ego agent and an adversary. The Nash equilibrium of this game satisfies a Hamilton-Jacobi-Isaacs (HJI) partial differential equation, which can be numerically solved via state space discretization. In this zero-sum framework, the ego agent assumes that the adversarial agent is constantly attempting to compromise the ego’s safety, and will thus compute and execute a control strategy that steers the ego’s trajectory away from any feasible trajectory of the non-ego agent.

Adversarial reachability appropriately describes many intricate dynamic interactions, such as capture-the-flag, reach-avoid, and pursuit-evasion-defense games [4, 9]. However, it suffers from several significant limitations. First, this zero-sum formulation can only model dynamic interactions between two agents, or two groups of colluding agents, with opposing goals. This is inadequate for many motion-planning tasks, such as those involving traffic scenarios, which must account for the presence of an arbitrary number of agents, with possibly an arbitrary number of goals. Second, the adversarial nature of the zero-sum game leads to the construction of extremely conservative ego trajectories, since the ego must imagine the worst-case non-ego behavior that can possibly transpire Fig. 1. Our approach, on the other hand, avoids the first issue by considering a general-sum game applicable to N-player scenarios. Moreover, we avoid the second issue by modeling antagonistic non-ego behavior using the novel notion of an adversarial-to-cooperative time horizon, as shown in Fig. 1, than a worst-case bounded disturbance. By modeling non-ego agents as first adversarial, then cooperative, we avoid overly conservative ego strategies corresponding to purely adversarial non-ego trajectories that are unlikely to materialize.

B. Multi-Agent Forward Reachability

Reachability-based formulations can also be used for safety-critical path planning in a non-game theoretic manner. For instance, forward reachable sets (FRS) of the ego agent can be computed offline by numerically solving the Hamilton-Jacobi-Bellman equation, then used to aid online motion planning modules in the generation of obstacle-avoiding trajectories. This is the approach taken by the Reachability-based Trajectory Design for Dynamical environments (RTD-D) and RTD-Interval (RTD-I) algorithms presented in [10, 11], in which not-at-fault ego trajectories are generated by leveraging the offline-computed FRS of the ego agent and online obstacle motion predictions from an external module. Although the resulting trajectories avoid at-fault collisions, this framework does not allow the ego agent to account for the dynamic reactions of non-ego agents to its behavior. Our work explicitly models the dynamic obstacles as non-ego agents, within the framework of a dynamic feedback game.

Geometric prediction modules form another framework for using reachability-based methods in a non-game-theoretic setting. For instance, [12, 13] ensure constraint satisfaction by computing fail-safe ego trajectories which avoid an overapproximation of all dynamically feasible non-ego trajectories. This is posed as an optimal control problem (rather than a dynamic game), with the set of all feasible non-ego trajectories serving as a state constraint. Although these approaches ensure that the ego will not collide with the non-ego agent, they do so at the cost of generating overly conservative maneuvers, particularly in situations when the non-ego agent may not be purely adversarial throughout the entire time horizon. By contrast, our formulation generates less conservative trajectories by assuming that non-ego agents display hostile behavior only during a fixed subset of the overall time horizon.

C. Probabilistic Constraint Satisfaction

Probabilistic constraint satisfaction is another commonly used method for establishing safety guarantees in motion planning. These approaches bound the probability that an ego agent, operating in an unpredictable environment with stochastic disturbance, becomes unsafe [5]. In particular, risk-sensitive algorithms guard the ego agent from low-probability events that may result in highly dangerous outcomes. For example, [14] generates risk-sensitive trajectories by optimizing an exponential-quadratic cost term that amplifies the cost of low-probability, yet highly dangerous outcomes. Meanwhile, [4] associates individual constraint
we are interested in ensuring the safety of one particular dynamics map \( f \).

Concatenating the dynamical quantities of interest of each input of player

Here, we restricted to open-loop games \([23, 24]\). Although other game solvers, such as the Augmented Lagrangian GAME-theoretic Solver (ALGAMES) of players. ILQGames iteratively solves linear-quadratic gamesally inefficient overall.

Optimal control problems reduces computation time at each iteration; however, IGR algorithms can still be computationally inefficient overall.

Our work uses ILQGames \([21]\), a recently developed iterative linear-quadratic algorithm, as our primary game solver. ILQGames iteratively solves linear-quadratic games and incurs computational complexity cubic in the number of players. Although other game solvers, such as the Augmented Lagrangian GAME-theoretic Solver (ALGAMES) \([22]\), exist, all known implementations of these methods are restricted to open-loop games \([23, 24]\).

### III. Preliminaries

Consider the \( N \)-player finite horizon general-sum differential game with nonlinear system dynamics:

\[
\dot{x} = f(t, x, u_{1:N}).
\]

Here, \( x \in \mathbb{R}^n \) is the state of the system, obtained by concatenating the dynamical quantities of interest of each player. \( t \in \mathbb{R} \) denotes time, \( u_i \in \mathbb{R}^{m_i} \) is the control input of player \( i \), for each \( i \in \{1, \ldots, N\} := [N] \), and \( u_{1:N} := (u_1, \ldots, u_N) \in \mathbb{R}^m \), where \( m := \sum_{i=1}^{N} m_i \). The dynamics map \( f : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n \) is assumed to be continuous in \( x \) and continuously differentiable in \( x(t) \) and \( u_i(t) \), for each \( i = 1, \ldots, N \) and each \( t \in [0, T] \). Since we are interested in ensuring the safety of one particular player amidst their interactions with all other players, we refer to Player 1 as the ego agent, and the other players as non-ego agents, defined as the integral of a running cost \( g_i : [0, T] \times \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R} \) over the time horizon \([0, T]\):

\[
J_i(u_{1:N}(\cdot)) = \int_0^T g_i(t, x(t), u_{1:N}(t)) \, dt,
\]

for each \( i \in \{1, \ldots, N\} \). The running costs \( g_i \) encode implicit dependence on the state trajectory \( x(\cdot) : [0, T] \rightarrow \mathbb{R}^n \) and explicit dependence on the control signals \( u_{1:N}(\cdot) : [0, T] \rightarrow \mathbb{R}^m \).

To minimize its cost, each player selects a control strategy to employ over the time horizon \([0, T]\), as described below. We assume that, at each time \( t \in [0, T] \), each player \( i \) observes the state \( x(t) \), but not the other control’s inputs \( \{u_j(t) \mid j \neq i\} \), and uses this information to design their control, i.e.

\[
u(t) := \gamma_i(t, x(t)),
\]

where \( \gamma_i : [0, T] \times \mathbb{R}^n \rightarrow \mathbb{R}^{m_i} \), defined as Player \( i \)’s strategy, is assumed to be measurable. We define the strategy space of Player \( i \), denoted \( \Gamma_i \), as the collection of all of Player \( i \)’s possible strategies, and denote, with a slight abuse of notation, the overall cost \( J_i \) of each Player \( i \) by:

\[
J_i(\gamma_1, \ldots, \gamma_N) \equiv J_i(\gamma_1(\cdot, x(\cdot)), \ldots, \gamma_N(\cdot, x(\cdot))).
\]

We now define the Nash equilibrium of the game described above.

**Definition 1:** (Nash equilibrium, \([25]\) Chapter 6) The strategy set \( (\gamma_1^*, \ldots, \gamma_N^*) \) is said to be a Nash equilibrium if no player is unilaterally incentivized to deviate from his or her strategy. Precisely, the following inequality must hold for each player \( i \):

\[
J_i(\gamma_1^*, \ldots, \gamma_{i-1}^*, \gamma_i, \gamma_{i+1}^*, \ldots, \gamma_N^*) \leq J_i(\gamma_1^*, \ldots, \gamma_{i-1}^*, \gamma_i^*, \gamma_{i+1}^*, \ldots, \gamma_N^*), \forall \gamma_i \in \Gamma_i.
\]

The search for a global Nash equilibrium is computationally intractable for dynamic games with general dynamics and cost functions. As such, in this work, we concern ourselves with finding a local Nash equilibrium, which is defined similarly to \((\cdot)\), but with the inequalities only constrained to hold within a neighborhood of the strategy set \( (\gamma_1^*, \ldots, \gamma_N^*) \). Moreover, in the dynamic games considered in this paper, additional constraints will be imposed on the dynamical quantities of each player, to model appropriate behavior between autonomous agents in traffic scenarios. These constraints will translate into a set of state constraints, and will significantly affect the set of Nash equilibria of the game. As such, in this work, we search for a (similarly defined) generalized local Nash equilibrium.

### IV. Methods

Our main contribution is a novel formulation of safety, best understood through the lens of defensive driving. In Sec. IV-A, we describe how, in the ego agent’s mind, the concept of defensive driving can be encoded into the running cost of each non-ego agent, i.e. \( g_i(x, u_{1:N}) \), for
each \( i \in \{2, \cdots, N\} \). To demonstrate this defensive driving framework in practice, we simulate realistic traffic scenarios; Sec. [V-B] details the dynamics, costs, and constraints imposed on the various agents in these simulations. Finally, in Sec. [V-C] we summarize the ILQGames algorithm as the main feedback game solver used in this work.

### A. Encoding Defensive Driving as a Running Cost

In our framework, the ego agent (Player 1) operates under the assumption that all other agents are momentarily distracted. To encode this “imaged” scenario, the ego agent imagines the overall time horizon \([0, T]\) as divided into two sub-intervals, the adversarial interval \([0, T_{adv}]\) and cooperative interval \([T_{adv}, T]\), with \(0 < T_{adv} < T\). During the adversarial interval, the ego agent imagines the other agents to be “momentarily distracted”, and desires to act defensively. This phenomenon is modeled using an adversarial running cost \( g_{adv,i} : [0, T_{adv}] \rightarrow \mathbb{R} \) for each \( i \in \{2, \cdots, N\} \). On the other hand, during the cooperative interval, the ego agent imagines the other agents to be “sufficiently little. The main feedback game solver used in this work.

![Image](image.png)

\[ J_i \text{ costs on } \]

\[ \text{Note that we have dropped the dependence of these costs on } t \text{ for clarity.} \]

\[ g_i(t, u_{1:N}) = \begin{cases} 
 g_{adv,i}(x, u_{1:N}), & t \in [0, T_{adv}], \\
 g_{coop,i}(x, u_{1:N}), & t \in [T_{adv}, T]. 
\end{cases} \]

In this scenario, the net integrated cost \( J_i \), first defined in (2) can be written as follows:

\[ \hat{J}_i := \int_0^{T_{adv}} g_{adv,i}(x, u_{1:N})dt + \int_{T_{adv}}^T g_{coop,i}(x, u_{1:N})dt. \]

With increasing \( T_{adv} \), the ego agent imagines an increasingly adversarial encounter and acts more and more defensively as a result. In practice, the user or system designer would select a suitable \( T_{adv} \) before operation, e.g., by choosing the largest \( T_{adv} \) such that the solution deviates from a nominal solution with \( T_{adv} = 0 \) sufficiently little.

### B. Simulation Setup

To test this construction, we simulate two traffic encounters that involve significant interaction (see Sec. [V]), in which a responsible human driver would likely drive defensively. Our method attempts to capture the spectrum of this “defensive” behavior as \( T_{adv} \), the duration of the adversarial time horizon, is varied. In each setting, each agent (in this case, each car) has augmented bicycle dynamics, i.e.:

\[ p_{x,i} = v_i \sin \theta_i, \quad \dot{p}_{x,i} = a_i \]
\[ p_{y,i} = v_i \cos \theta_i, \quad \dot{p}_{y,i} = \omega_i \]
\[ \dot{\theta}_i = (v_i/L_i) \tan \phi_i, \quad \dot{\phi}_i = j_i, \]

where \( x = (p_{x,i}, p_{y,i}, \theta_i, v_i, \omega_i, a_i)_{i=1}^N \) represents each vehicle’s position, heading, speed, front wheel angle, and acceleration, and \( u_i = (\omega_i, j_i) \) represents each vehicle’s front wheel rate and tangent jerk. \( L_i \) is each player’s inter-axle distance.

We define \( g_{adv,i} \) and \( g_{coop,i} \) as weighted combinations of the following functions, with different behavior encouraged through the use of different weighting coefficients. We use the shorthand \( p_i = (p_{x,i}, p_{y,i}) \) for the position of each agent, \( d\_adv(p_i) = \|p_i - p_{ref,i}\| \) for the distance between an agent and the corresponding lane centerline \( x_{ref,i} \) in the \((p_{x,i}, p_{y,i})\)-plane, and \( d_{prox} \) for a constant desired minimum proximity between agents:

\[ \text{input: } u_i^T R_{i} u_i \]
\[ \text{lane center: } d\_adv(p_i)^2 \]
\[ \text{ideal speed: } (v_i - v_{ref,i})^2 \]
\[ \text{cooperative: } 1 \{ \|p_i - p_j\| < d_{prox} \}(d_{prox} - \|p_i - p_j\|)^2 \]
\[ \text{adversarial: } ||p_i - p_j||^2. \]

Recall that, for non-ego agents, the “adversarial” cost is only present during the adversarial horizon \([0, T_{adv}]\) and the “cooperative” cost is present thereafter during the cooperative horizon \([T_{adv}, T]\). We also enforce the following inequality constraints, where \( d_{lane} \) is the lane half-width, and \( v_i \) and \( \tau_i \) denote speed limits:

\[ \text{proximity: } ||p_i - p_j|| > d_{prox} \]
\[ \text{lane: } |d\_adv(p_i)| < d_{lane} \]
\[ \text{speed range: } v_i < v_i < \tau_i. \]

Here, the “proximity” constraint is enforced only for the ego agent, to force the ego to bear responsibility for satisfying joint state constraints which encode his her own safety (e.g. non-collision). In addition, all agents must satisfy individual constraints that encode reasonable conduct in traffic (e.g., staying within a range of speeds). All constraints are enforced over the entire time horizon \([0, T]\). For all tests, we use a time horizon \( T = 15\text{s} \) and discretize time (following [21] and [25]) at 0.1\text{s} intervals.

### C. Implementation Details

The traffic simulations in this work are solved approximately to local feedback Nash equilibria in real time using ILQGames, a recently developed, open-source C++ based game solver algorithm introduced in [21]. ILQGames iteratively solves linear-quadratic games, obtained by linearizing dynamics and quadraticizing costs, and inculs computational complexity that is cubic in the number of players [21]. While we note that the original paper does not address constrained Nash games, we are currently preparing a manuscript concerning this issue. The modified solver accounts for both equality and inequality constraints on \( x(t) \) and \( u_i(t) \). We note that other game solvers, such as ALGAMES [22] and Iterative Best Response algorithms [20], can likewise handle constraints; however, their applications are restricted to open-loop games. A thorough treatment of constraints
Fig. 2: Oncoming example. The ego vehicle (right lane, heading upwards) and the non-ego vehicle (left lane, heading downwards) perform increasingly extreme maneuvers as $T_{adv}$ increases, in this “oncoming” scenario. Dark blue, turquoise, and light green are used to represent the agents’ location at $T_{adv} = 0, 0.5, 1.0$ s, respectively. When $T_{adv} = 0$ s, the ego vehicle does not deviate significantly from its lane because it anticipates that the non-ego vehicle will behave cooperatively throughout the entire time horizon by swerving to avoid a collision. However, when $T_{adv} = 5$ s, the ego vehicle will actively swerve outward to avoid the non-ego agent, because it anticipates that instead, the non-ego agent will behave adversarially during the first 5 seconds of the time horizon.

in games can be found in [26]. A preliminary version of the constrained-ILQGames solver was used in the examples presented here. We note, however, that in practice these constraints can be replaced with “soft costs” of appropriate magnitude.

V. RESULTS

We present simulation results for various traffic scenarios in which a responsible traffic participant would likely drive defensively. First, we consider a simple situation involving oncoming vehicles on a straight road, as a proof of concept. Then, we analyze a more complicated intersection example with a crosswalk. In both cases, the ILQGames algorithm solves our defensive game in real-time.

A. Oncoming Example

In this example, the ego car is traveling North on a straight road when it encounters another car traveling South. Since the road has a lane in each direction, “ideally” the ego vehicle would not deviate too far from its lane or speed. However, to drive more defensively, the ego vehicle should plan as though the oncoming Southbound car were to act noncooperatively. Our method encodes precisely this type of defensive planning. Fig. 2 shows the planned trajectories that emerge for increasing $T_{adv}$. As shown, the ego vehicle (red) imagines more aggressive maneuvers for itself and the oncoming car as $T_{adv}$ increases. Note, however, that these are merely imagined trajectories and that (a) the ego vehicle can always choose to follow this trajectory only for an initial period of time, and recompute its trajectory thereafter, and (b) the oncoming vehicle will make its own decisions and will not generally follow this “partially adversarial” trajectory. We solve each of these problems (with fixed $T_{adv}$) in under 0.5 s.

B. Three-Player Intersection Example

This simulated traffic scenario is designed to model the behavior of two vehicles and a pedestrian at an intersection. As shown in Fig. 3, the ego vehicle is present in the intersection alongside a non-ego vehicle heading in the opposite direction, who wishes to make a left turn, and a pedestrian, who wishes to cross the road. To reach their goal locations, these three agents must cross paths in the intersection. When $T_{adv} = 0$ s, the ego vehicle continues straight along its lane because it anticipates that the non-ego vehicle will behave cooperatively throughout the entire time horizon. In particular, it anticipates that the non-ego vehicle will continue along its curved path at nominal speed, resulting in a collision-free trajectory. However, as with the oncoming example, the ego vehicle’s trajectory becomes increasingly more conservative as the adversarial time horizon increases in length. When $T_{adv} = 1$ s, the ego vehicle actively swerves rightwards to avoid the non-ego vehicle. This is because the ego vehicle assumes, in this scenario, that the non-ego vehicle’s initial speed is more gradual, and will thus approach the intersection at the same time as the ego vehicle. As before, each problem is solved in well under 0.75 s in single-threaded operation on a standard laptop, via the ILQGames algorithm [21]. This performance indicates real-time capabilities which will be explored in future work on
Fig. 3: Three Player Intersection example. The ego vehicle (right lane, heading upwards) navigates through an intersection while avoiding collision with a non-ego vehicle (left lane, heading downwards initially before making a left turn) and a pedestrian (horizontal path at the intersection, left to right). Dark blue, turquoise, and light green are used to represent the agents’ location at $T_{adv} = 0.0, 2.5, 5.0$ s, respectively. As with the oncoming example, these three agents perform increasingly extreme maneuvers as $T_{adv}$ increases. In particular, when $T_{adv} = 0$ s, the ego vehicle anticipates that the non-ego will continue along its curved path at its nominal speed, allowing it to approach the intersection before the ego vehicle does. Thus, the ego vehicle swerves leftwards, to avoid the non-ego agent as it makes its left turn and continues rightwards in the figure, resulting in a collision-free trajectory. (The pedestrian’s trajectories for $T_{adv} = 0$ s and $T_{adv} = 0.5$ s coincide.) However, when $T_{adv} = 1$ s, the ego vehicle actively swerves rightwards to avoid the non-ego vehicle, assumed in this scenario to be driven at a more gradual speed, approaching the intersection at the same time as the ego vehicle. The pedestrian’s speed also noticeably decreases, as it likewise becomes more motivated to approach the intersection at the same time as the ego vehicle.

VI. DISCUSSION

We have presented a novel formulation of robustness in motion planning for multi-agent problems. Modeled on defensive driving, our method explicitly models other agents as adversarial in only a limited, initial portion of the overall time interval. Like adversarial methods in Hamilton-Jacobi-Isaacs (HJI) optimal control, our approach draws upon earlier work in differential game theory. However, instead of forcing the ego to respond to the unpredictable behavior of non-ego agents by avoiding all feasible non-ego trajectories, we use a piecewise definition of game cost to endow the ego with the perspective that other agents are temporarily distracted. As such, our approach generates far less conservative behavior than purely adversarial (e.g., HJI) methods. Simulation results illustrate that this novel formulation of safety can be used to solve these “defensive” problems in real-time. We are eager to implement this method in hardware and test its performance in a receding time horizon with other (human) drivers.

VII. FUTURE WORK

Future work will examine a more diverse collection of methods for encoding defensive behavior in games. For example, it may be desirable to allow the ego agent to select the adversarial time horizon more flexibly. Of particular interest is the case where the ego agent may choose to vary $T_{adv}$ from one non-ego agent to another. For example, the ego may observe that some non-ego agents are behaving more adversarially than others, and respond accordingly by associating such players with higher values of $T_{adv}$. In addition, the ego may wish to allocate different parts of the overall time horizon to be adversarial, rather than simply the first $T_{adv}$ seconds. For example, choosing the adversarial time horizon to be the final $T_{adv}$ seconds of the overall time horizon, rather than the first $T_{adv}$ seconds, would transform the game from adversarial-to-cooperative type to cooperative-to-adversarial type. This would be useful in situations where the ego agent predicts that the surrounding non-ego agents are currently focused on cooperation, but may enter a state of momentary distraction in the near future. For example, such a formulation may cause the ego agent to gradually approach an intersection at which other agents might run a red light.

Alternatively, constraint satisfaction on the part of the ego can also be directly encoded into the information structure of the dynamic game itself, e.g. by delegating the ego as the follower in a feedback Stackelberg game. This is a topic of the authors’ ongoing research.

Future work will also examine the physical implementation of this framework on hardware, and test its performance in a receding horizon fashion in real-life traffic scenarios with other human drivers. We also anticipate the use of this novel notion of safety in game-theoretic applications outside of motion planning, such as in economics or public health applications, where it may be prudent to model certain agents as adversarial in certain time intervals.

REFERENCES

[1] Somil Bansal et al. “Hamilton-Jacobi reachability: A brief overview and recent advances”. In: 2017 IEEE
