Hierarchical Game-Theoretic Planning for Autonomous Vehicles

Jaime F. Fisac*[1] Eli Bronstein*[1] Elis Stefansson[2] Dorsa Sadigh[3] S. Shankar Sastry[1] Anca D. Dragan[1]

***Abstract*— The actions of an autonomous vehicle on the road affect and are affected by those of other drivers, whether overtaking, negotiating a merge, or avoiding an accident. This mutual dependence, best captured by dynamic game theory, creates a strong coupling between the vehicle's planning and its predictions of other drivers' behavior, and constitutes an open problem with direct implications on the safety and viability of autonomous driving technology. Unfortunately, dynamic games are too computationally demanding to meet the real-time constraints of autonomous driving in its continuous state and action space. In this paper, we introduce a novel game-theoretic trajectory planning algorithm for autonomous driving, that enables real-time performance by hierarchically decomposing the underlying dynamic game into a long-horizon "strategic" game with simplified dynamics and full information structure, and a short-horizon "tactical" game with full dynamics and a simplified information structure. The value of the strategic game is used to guide the tactical planning, implicitly extending the planning horizon, pushing the local trajectory optimization closer to global solutions, and, most importantly, quantitatively accounting for the autonomous vehicle and the human driver's ability and incentives to influence each other. In addition, our approach admits non-deterministic models of human decision-making, rather than relying on perfectly rational predictions. Our results showcase richer, safer, and more effective autonomous behavior in comparison to existing techniques.**

I. Introduction

Imagine you are driving your car on the highway and, just as you are about to pass a large truck on the other lane, you spot another car quickly approaching in the wing mirror. Your driver's gut immediately gets the picture: the other driver is trying to squeeze past and cut in front of you at the very last second, barely missing the truck. Your mind races forward to produce an alarming conclusion: it is too tight—yet the other driver seems determined to attempt the risky maneuver anyway. If you brake immediately, you could give the other car enough room to complete the maneuver without risking an accident; if you accelerate, you might close the gap fast enough to dissuade the other driver altogether.

Driving is fundamentally a game-theoretic problem, in which road users' decisions continually couple with each other over time. Accurately planning through these road interactions is a central, safety-critical challenge in autonomous driving. Most approaches in the literature follow a "pipeline" approach that generates predictions of the trajectories of human-driven vehicles and then feeds them to the planning module as unalterable moving obstacles [1–4]. This can lead to both excessively conservative and in some cases unsafe behavior [5], a well-studied issue in the robotic navigation literature known as the "frozen robot" problem [6].

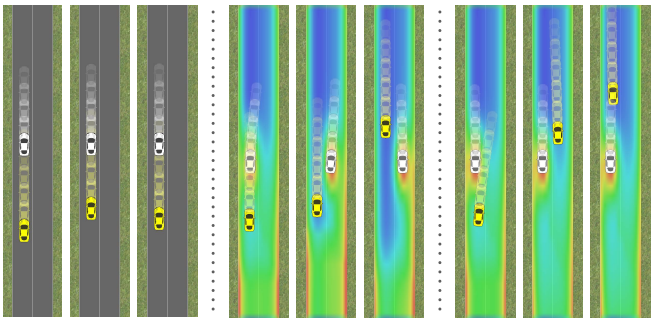


Fig. 1: Demonstration of our hierarchical game-theoretic planning framework on a simulated overtaking scenario. The heatmap displays the hierarchical planner's strategic value, ranging from red (low value) to blue (high value), which accounts for the outcome of possible interactions between the two vehicles. *Left:* Using a short-horizon trajectory planner, the autonomous vehicle slows down and is unable to overtake the human. *Center:* Using the hierarchical game-theoretic planner, the autonomous vehicle approaches the human from behind, incentivizing her to change lanes and let it pass (note the growth of a high-value region directly behind the human in the left lane). *Right:* If the human does not maneuver, the autonomous vehicle executes a lane change and overtakes, following the higher values in the right lane.

Recent work has addressed this by modeling human drivers as utility-driven agents who will plan their trajectory in response to the autonomous vehicle's internal plan. The autonomous vehicle can then select a plan that will elicit the best human trajectory in response [7, 8]. Unfortunately, this treats the human as a pure *follower* in the game-theoretic sense, effectively inverting the roles in previous approaches. That is, the human is assumed to take the autonomous vehicle's future trajectory as immutable and plan her own fully accommodating to it, rather than try to influence it. Further, the human driver must be able to observe, or exactly predict, the future trajectory planned by the autonomous vehicle, which is unrealistic beyond very short planning horizons.

In this work, we introduce a hierarchical game-theoretic framework to address the mutual influence between the human and the autonomous vehicle while maintaining computational tractability. In contrast with recent game-theoretic planning schemes that assume open-loop information structures [9–11], our framework hinges on the use of a fully coupled interaction model in order to plan for horizons of multiple seconds, during which drivers can affect each

[1]Department of Electrical Engineering and Computer Sciences. University of California, Berkeley, United States.
[2]Department of Mathematics, KTH Royal Institute of Technology, Sweden.
[3]Computer Science Department, Stanford University, United States. Email: {ebronstein, shankar_sastry, jfisac, anca}@berkeley.edu, elisst@kth.se dorsa@cs.stanford.edu
*The first two authors contributed equally to this work.
This work was partially supported by an NSF CAREER award and a Ford URP. We thank NVDIA for hardware support.
The authors would like to thank Karl H. Johansson for helpful discussions and valuable input on vehicle coordination.

other's behavior through their actions over time. We do this by computing the optimal value and strategies for a dynamic nonzero-sum game with a long horizon (typically a few seconds) and a full closed-loop feedback information structure [12, 13]. In order to maintain tractability, we propose solving this long-horizon game using simplified dynamics, which will approximately capture the vehicles' ability to execute different trajectories. The resulting long-term value, which captures the expected outcome of the strategic interaction from every state, can then be used as an informative terminal component in the objective function used in a receding-horizon planning and control scheme. This low-level planner can use a higher-fidelity representation of the dynamics, while only planning for a shorter time horizon (typically less than one second) during which simplifications in the interaction have a less critical effect [14–16].

Our framework therefore hierarchically combines:

- A *strategic* (high-level) planner that determines the outcome of long-term interactions using simplified dynamics and fully coupled interaction.
- A *tactical* (low-level) planner that computes short-term vehicle trajectories using high-fidelity dynamics and simplified interaction, informed by the long-term value computed by the strategic planner.

Thanks to the more accurate interaction model and the more tractable dynamical model, the hierarchical framework makes it possible to reason farther into the future than most receding-horizon trajectory planners. The high-level game value informs the trajectory optimization as a terminal cost, implicitly giving it an approximate insight into the longer time scale (in a similar spirit to a variety of planning schemes, e.g. [17]). In addition, since this strategic value is computed globally via dynamic programming, it can help mitigate the local nature of most trajectory optimization schemes, biasing them towards better solutions.

An important strength of our framework is that the strategic planner does not require using a deterministic model of the human, such as an ideal rational agent, but instead allows a variety of models including probabilistic models such as noisy rationality, commonly used in inverse optimal control (also inverse reinforcement learning) [18, 19]. In addition, the framework is agnostic to the concrete planner used at the tactical level: while we demonstrate our approach with a trajectory optimizer based on [7], this could be replaced with other methods, including deep closed-loop prediction models, such as [20], by introducing the strategic value as a terminal cost term in their objective function. Therefore, the method proposed here should not be seen as competing with such planning schemes, but rather as complementing them.

Importantly, solving the underlying dynamic game does not imply that the autonomous vehicle will be more selfish or aggressive—its driving behavior will ultimately depend on the optimization objective specified by the system designer, which may include terms encoding comfort and safety of other road users. With adequate objective design, our framework can enable safer and more efficient autonomous driving by planning with a more accurate model of interactions.

II. Dynamic Game Formulation

We consider a single[1] human driver H and a single autonomous system A in control of their respective vehicles. The dynamics of the joint state $x^t \in \mathcal{X} \subset \mathbb{R}^n$ of the vehicles in the world, which we assume to be fully observable, are

$$x^{t+1} = f(x^t, u_A^t, u_H^t) \ , \tag{1}$$

where $u_i^t \in \mathcal{U}_i \subset \mathbb{R}^{m_i}$ is the driving control action for each $i \in \{A, H\}$ at time step t; we assume $\mathcal{U}_i$ is compact.

The autonomous system is attempting to maximize an objective that depends on the evolution of the two vehicles over some finite time horizon, namely a cumulative return:

$$R_A(x^{0:N}, u_A^{0:N}, u_H^{0:N}) = \sum_{t=0}^{N} r_A(x^t, u_A^t, u_H^t) \ . \tag{2}$$

The reward function r_A captures the designer's specifications of the vehicle's behavior and may encode aspects like fuel consumption, passenger comfort, courteousness, time efficiency, and safety . Some of these aspects (crucially safety) may depend on the joint state of the two vehicles; the reward function may also explicitly depend on the human driver's actions (the designer may, for instance, decide to penalize it for causing other vehicles to maneuver abruptly). The autonomous vehicle therefore needs to reason about not only its own future actions, but also those of the human driver.

We assume that the autonomous vehicle has some predictive model of the human's actions as a function of the currently available information (the joint state, and possibly the autonomous vehicle's current action). The coupling in the planning problem is then explicit. If the system models the human as exactly or approximately attempting to maximize her own objective function, the coupling takes the form of a dynamic game, in which each player acts strategically as per her own objective function accounting for the other's possible actions. Since both players observe the current state at each time, this dynamic game has closed-loop feedback information structure, and optimal values and strategies can be computed using dynamic programming [12, 24].

Unfortunately, deriving these strategies can be computationally prohibitive due to the exponential scaling of computation with the dimensionality of the joint state space (which will be high for the dynamical models used in vehicle trajectory planning). However, we argue that successfully reasoning about traffic interactions over a horizon of a few seconds does not require a full-fidelity model of vehicle dynamics, and that highly informative insights can be tractably obtained through approximate models. We further argue that it is both useful and reasonable to model human drivers as similarly reasoning about vehicle interactions over the next few seconds without needing to account for fully detailed dynamics. This insight is at the core of our solution approach.

[1] While extension of our formulation and solution to N players is well-defined (and relatively straightforward) in *theory*, in practice the solution requires exponential computation in the number of interacting vehicles, which constitutes a fundamental open problem. We thus limit the scope of this work to pairwise interactions, and note that decomposition strategies [21, 22] and recent prediction approaches [23] may enable tractable extensions.

III. Hierarchical Game-Theoretic Planning

We propose a hierarchical decomposition of the interaction between the autonomous vehicle and the human driver. At the high level, we solve a dynamic game representing the long-horizon interaction between the two vehicles through approximate dynamics. At the low level, we use the players' computed value functions as an approximation of the best long-horizon outcome achievable by both vehicles from each joint state, and incorporate it in the form of a guiding terminal term in the short-horizon trajectory optimization, which is solved in a receding-horizon fashion with a high-fidelity model of the vehicles' dynamics.

A. Strategic planner: Closed-loop dynamic game

Let the approximate dynamics be given by

$$s^{k+1} = \phi(s^k, a_A^k, a_H^k) \ , \tag{3}$$

where $s^t \in \mathcal{S} \subset \mathbb{R}^{\tilde{n}}$ and $a_i^t \in \mathcal{A}_i \subset \mathbb{R}^{\tilde{m}_i}$ are the state and action in the simplified dynamics ϕ. The index k is associated to a discrete time step that may be equal to the low-level time step or possibly different (typically coarser). We generically assume that there exists a function $g : \mathcal{X} \to \mathcal{S}$ assigning a simplified state $s \in \mathcal{S}$ to every full state $x \in \mathcal{X} \subset \mathbb{R}^n$. The approximation is usually made seeking $\tilde{n} < n$ to improve tractability. This can typically be achieved by ignoring dynamic modes in f_i with comparatively small time constants. For example, we may assume that vehicles can achieve any lateral velocity within a bounded range in one time step, and treat it as an input instead of a state.

We model the dynamic game under feedback closed-loop information (both players' actions can depend on the current state s but not on the state history), allowing the human driver to condition her choice of a_H^k on the autonomous vehicle's current action a_A^k at every time step k, resulting in a Stackelberg (or leader-follower) dynamic game [24]. We need not assume that the human is an ideal rational player, but can instead allow her action to be drawn from a probability distribution. This is amenable to the use of human models learned through inverse optimal control methods [18, 25], and can also be used to naturally account for modeling inaccuracies, to the extent that the human driver's behavior will inevitably depart from the modeling assumptions [23].

We generalize the well-defined feedback Stackelberg dynamic programming solution [12] to the case in which one of the players, in this case the *follower*, has a noisy decision rule: $p(a_H^k | s^k, a_A^k)$. The autonomous vehicle, here in the role of the *leader*, faces at each time step k the nested optimization problem of selecting the action with the highest state-action Q value, which depends on the human's decision rule p, in turn affected by the human's own Q values:

$$\max_{a_A^k} \ Q_A^k(s^k, a_A^k) \tag{4a}$$

$$\text{s.t. } p(a_H^k \mid s^k, a_A^k) = \pi_H\big[Q_H^k(s^k, a_A^k, \cdot)\big](a_H^k) \tag{4b}$$

where Q_A^k and Q_H^k are the state-action value functions at time step k, and $\pi_H : L^\infty \to \Delta(\mathcal{A}_H)$ maps every utility function $q : \mathcal{A}_H \to \mathbb{R}$ to a probability distribution over $\mathcal{A}_H$.

Algorithm 1: Feedback Stackelberg Dynamic Program

Data: $\hat{r}_A(\hat{s}, \hat{a}_A, \hat{a}_H)$, $\hat{r}_H(\hat{s}, \hat{a}_A, \hat{a}_H)$
Result: $\hat{V}_A(\hat{s}, k)$, $\hat{V}_H(\hat{s}, k)$, $\hat{a}_A^*(\hat{s}, k)$, $\hat{a}_H^*(\hat{s}, k)$

Initialization
for $\hat{s} \in \hat{\mathcal{S}}$ **do**
A0 $\quad \hat{V}_A(\hat{s}, K+1) \leftarrow 0$;
H0 $\quad \hat{V}_H(\hat{s}, K+1) \leftarrow 0$;

Backward recursion
for $k \leftarrow K$ **to** 0 **do**
  for $\hat{s} \in \hat{\mathcal{S}}$ **do**
    for $\hat{a}_A \in \hat{\mathcal{A}}_A$ **do**
      for $\hat{a}_H \in \hat{\mathcal{A}}_H$ **do**
H1 $\quad\quad q_H(\hat{a}_H) \leftarrow \hat{r}_H(\hat{s}, \hat{a}_A, \hat{a}_H) + \hat{V}_H(\phi(\hat{s}, \hat{a}_A, \hat{a}_H), k+1)$;
H2 $\quad P(\hat{a}_H \mid \hat{a}_A) \leftarrow \pi_H[q_H](\hat{a}_H)$;
H3 $\quad q_H^*(\hat{a}_A) \leftarrow \sum_{\hat{a}_H} P(\hat{a}_H \mid \hat{a}_A) \times q_H(\hat{a}_H)$;
A1 $\quad q_A(\hat{a}_A) \leftarrow \sum_{\hat{a}_H} P(\hat{a}_H \mid \hat{a}_A) \times \Big(\hat{r}_A(\hat{s}, \hat{a}_A, \hat{a}_H^*(\hat{a}_A)) + \hat{V}_A(\phi(\hat{s}, \hat{a}_A, a_H^*(\hat{a}_A)), k+1)\Big)$;
A2 $\quad \hat{a}_A^*(\hat{s}, k) \leftarrow \arg\max_{\hat{a}_A} q_A(\hat{a}_A)$;
A3 $\quad \hat{V}_A(\hat{s}, k) \leftarrow q_A(\hat{a}_A^*(\hat{s}, k))$;
H4 $\quad \hat{a}_H^*(\hat{s}, k) \leftarrow a_H^*(\hat{a}_A^*(\hat{s}, k))$;
H5 $\quad \hat{V}_H(\hat{s}, k) \leftarrow q_H^*(\hat{a}_A^*(\hat{s}, k))$;

A common example of π_H (which we use in Section IV) is a noisy rational Boltzmann policy, for which:

$$P(a_H \mid s, a_A) \propto e^{\beta Q_H(s, a_A, a_H)} \ . \tag{5}$$

The values Q_A^k and Q_H^k are recursively obtained in backward time through successive application of the dynamic programming equations for $k = K, K-1, \ldots, 0$:

$$\pi_A^*(s) := \arg\max_a Q_A^{k+1}(s, a) \ , \quad \forall s \in \mathcal{S} \tag{6a}$$

$$a_H^i \sim \pi_H\big[Q_H^i(s^i, a_A^i, \cdot)\big] \ , \quad i \in \{k, k+1\} \tag{6b}$$

$$Q_H^k(s^k, a_A^k, a_H^k) = \tilde{r}_H(s^k, a_A^k, a_H^k) + \mathbb{E}_{a_H^{k+1}} Q_H^{k+1}(s^{k+1}, \pi_A^*(s^{k+1}), a_H^{k+1}) \tag{6c}$$

$$Q_A^k(s^k, a_A^k) = \mathbb{E}_{a_H^k} \tilde{r}_A(s^k, a_A^k, a_H^k) + Q_A^{k+1}(s^{k+1}, \pi_A^*(s^{k+1})) \tag{6d}$$

with s^{k+1} from (3) and letting $Q_A^{K+1} \equiv 0$, $Q_H^{K+1} \equiv 0$.

The solution approach is presented in Algorithm 1 for a discretized state and action grid $\hat{\mathcal{S}} \times \hat{\mathcal{A}}_A \times \hat{\mathcal{A}}_H$. This computation is typically intensive, with complexity $O\big(|\hat{\mathcal{S}}| \cdot |\hat{\mathcal{A}}_A| \cdot |\hat{\mathcal{A}}_H| \cdot K\big)$, but is also extremely parallelizable, since each grid element is independent of the rest and the entire grid can be updated simultaneously, in theory permitting a time complexity of $O(K)$. Although we precomputed the game-theoretic solution, our proposed computational method for the strategic planner can directly benefit from the ongoing advances in computer hardware for autonomous driving [26],

so we expect that it will be feasible to compute the strategic value in an online setting.

Once the solution to the game has been computed, rather than attempting to *execute* any of the actions in this simplified dynamic representation, the autonomous vehicle can use the resulting value $V(s) := \max_a Q^0(s,a)$ as a guiding terminal reward term for the short-horizon trajectory planner.

B. Tactical planner: Open-loop trajectory optimization

In this section we demonstrate how to incorporate the strategic value into a low-level trajectory planner. We assume that the planner is performing a receding-horizon trajectory optimization scheme, as is commonly the case in state-of-the-art methods [27]. These methods tend to plan over relatively short time horizons (on the order of 1 s), continually generating updated "open-loop" plans from the current state—in most cases the optimization is local, and simplifying assumptions regarding the interaction are made in the interest of real-time computability.

While, arguably, strategic interactions can be expected to have a smaller effect over very short time-scales, the vehicle's planning should be geared towards efficiency and safety beyond the reach of a single planning window. The purpose of incorporating the computed strategic value is to guide the trajectory planner towards states from which desirable long-term performance can be achieved.

We therefore formalize the tactical trajectory planning problem as an optimization with an analogous objective to (2) with a shorter horizon $M << N$ and instead introduce the strategic value as a terminal term representing an estimate of the optimal reward-to-go between $t = M$ and $t = N$:

$$R_A(x^{0:M}, u_A^{0:M}, u_H^{0:M}) = \sum_{t=0}^{M} r_A(x^t, u_A^t, u_H^t) + V_A\big(g(x^t)\big) \ . \tag{7}$$

The only modification with respect to a standard receding-horizon trajectory optimization problem is the addition of the strategic value term. Using the numerical grid computation presented in Section III-A, this can be implemented as an efficient look-up table, allowing fast access to values and gradients (numerically approximated directly from the grid).

The low-level optimization of (7) can thus be performed online by a trajectory optimization engine, based on some short-term predictive model of human decisions conditioned on the state and actions of the autonomous vehicle. In our results we implement trajectory optimization similar to [7] through a quasi-Newton scheme [28], in which the autonomous vehicle iteratively solves a nested optimization problem by estimating the human's best trajectory response to each candidate plan for the next M steps. We assume that the human has an analogous objective to the autonomous system, and can also estimate her strategic long-term value. We stress, however, that our framework is more general, and in essence agnostic to the concrete low-level trajectory optimizer used, and other options are possible (e.g. [20, 23]).

IV. Results

We analyze the benefit of solving the dynamic game by comparing our hierarchical approach to using a tactical planner only, as in the state of the art [7, 20]. We then compare against extended-horizon trajectory planning with an assumed open-loop information structure, showcasing the importance of reasoning with the fully coupled closed-loop feedback information of the dynamic game.

A. Implementation Details

1) Environment: We use a simulated two-lane highway environment with an autonomous car and human-driven vehicle. Similar to [7], both vehicles' rewards encode safety and performance features. For the purposes of these case studies, both players have a preference for the left lane, and the autonomous car is given a target speed slightly faster than the human's and a preference for being ahead of her.

2) Tactical Level: The dynamics of each vehicle are given by a dynamic bicycle model with states $[x_i, y_i, v_i, \theta_i]$ (position, speed, and heading). The planner uses a discrete time step $\Delta t = 0.1$ s and $M = 5$ time steps. For the tactical trajectory planning, we compute the partial derivative $\frac{\partial R_i}{\partial \mathbf{u}_i}$ for each player and allow the optimization to proceed by iterated local best response between candidate autonomous vehicle plans and predicted human trajectories. If convergence is reached, the result is a local (open-loop) Nash equilibrium between the short-horizon trajectories [9, 11, 29].

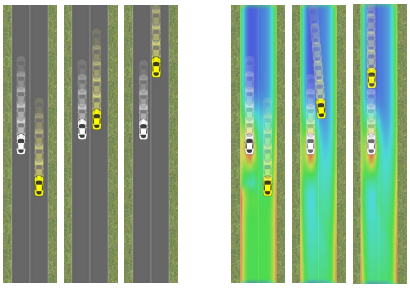
3) Strategic Level: The full joint human-autonomous state space is 8-dimensional, making dynamic programming challenging. Our strategic level simplifies the state and dynamics using an approximate, lower-order representation. We consider a larger time step of $\Delta k = 0.5$ s and a horizon $K = 10$ corresponding to 5 s. We consider one of two high-level models, depending on the setup.

Two-vehicle setup. If the environment is a straight empty highway, it is enough to consider the longitudinal position of the two vehicles relative to each other: $x_{\text{rel}} = x_{\mathcal{A}} - x_{\mathcal{H}}$. We assume the human-driven vehicle's average velocity is close to the nominal highway speed 30 m/s, and the vehicles' headings are approximately aligned with the road at all times. Finally, given the large longitudinal velocity compared to any expected lateral velocity, we assume that vehicles can achieve any desired lateral velocity up to ± 2.5 m/s^2 within one time step (consistent with a typical 1.5 s lane change). The approximate dynamics are then

$$[\dot{x}_{\text{rel}}, \dot{y}_A, \dot{y}_H, \dot{v}_{\text{rel}}] = [v_{\text{rel}}, w_A, w_H, a_A - a_H - \tilde{\alpha} v_{\text{rel}}] \ , \tag{8}$$

with the control inputs being the vehicles' lateral velocities w_A, w_H and accelerations a_A, a_H, and where $\tilde{\alpha}$ is the linearized friction parameter. This allows us to implement Algorithm 1 on a $75 \times 12 \times 12 \times 21$ grid and compute the feedback Stackelberg solution of the strategic game.

Additional-vehicle setup. If there are additional vehicles or obstacles present in the environment, it becomes necessary to explicitly consider absolute positions and velocities of the two players' vehicles (or at least relative to these other objects). In this scenario, we consider a truck driving in



(a) Low-level planner (b) Hierarchical planner

Fig. 2: Planner comparison for the merging scenario. The low-level trajectory planner overtakes but does not merge into the left lane. The game-theoretic hierarchical planner successfully merges in front of the human.

the right lane at a constant speed v_T, and assume that the human remains in her lane. Letting x_{AT}, x_{HT} denote the longitudinal position of each vehicle relative to the truck and α be the friction coefficient, the high-level dynamics are

$$[\dot{x}_{AT}, \dot{x}_{HT}, \dot{y}_A, \dot{v}_A, \dot{v}_H] = \\ [v_A - v_T,\ v_H - v_T,\ w_A,\ a_A - \alpha v_A^2,\ a_H - \alpha v_H^2]. \quad (9)$$

We implement Algorithm 1 on a $35 \times 35 \times 6 \times 8 \times 8$ grid.

4) Human simulation.: For consistency across our case studies, we simulated the human driver's behavior. We found that for the maneuvers considered, a low-level trajectory optimizer produced sufficiently realistic driving behavior. We assume that the human driver makes accurate predictions of the autonomous vehicle's imminent trajectory for 0.5 s.

B. Interaction Case Studies

We compare the tactical-only trajectory planner (baseline) against our hierarchical tactical-strategic planning scheme for 3 different driving scenarios.

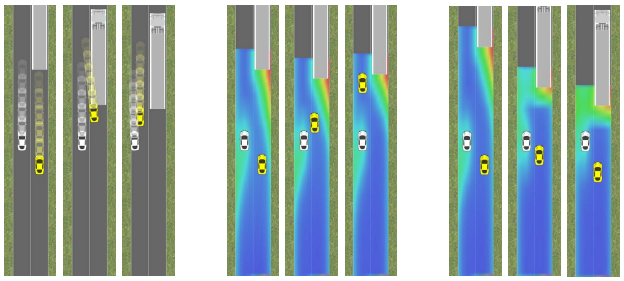
1) Merge: We begin with a simple merge maneuver where the autonomous vehicle starts *ahead* of the human in the adjacent lane. The tactical planner leads the autonomous car to successfully merge in front of the human. The hierarchical planner also succeeds, with the strategic value guiding the vehicle to merge more swiftly, improving performance.

Next, we consider the case where the autonomous car starts *behind* the human, as depicted in Fig. 2. The tactical autonomous car overtakes the human but does not merge into her lane (likely a local optimum). The hierarchical autonomous car overtakes and merges in front of the human.

2) Overtaking: We now study a complete overtaking maneuver in which the autonomous car starts behind the human in the *same* lane. The tactical autonomous car does not successfully complete the maneuver: it first accelerates but then brakes to remain behind the human, oblivious to the higher long-term performance achievable through overtaking. The hierarchical planner produces a policy that, depending on the human's behavior, can evolve into two alternative strategies, shown in Fig. 1. First, the autonomous vehicle approaches the human from behind, expecting her to have an incentive (based on her strategic value) to change lanes and let it pass. If this initial strategy is successful and the human changes lanes, the autonomous vehicle overtakes without leaving the left lane. Conversely, if the human does not begin a lane change, the strategic value guides the autonomous vehicle to merge into the right lane, accelerate to overtake the human, reaching a maximum speed of 37.83 m/s (2.83 m/s above its target speed), and merge back into the original lane.

3) Truck Cut-In: Finally we consider a scenario in which the two vehicles are approaching a truck, assumed to drive at a lower constant speed of 26.82 m/s. As shown in Fig. 3, the tactical-only planner may attempt merges with little safety margin. The hierarchical game-theoretic analysis allows us to reason through the leverages players may have on each other. If the autonomous vehicle has a sufficient initial speed, the human is incentivized to slow down to allow it to merge safely in front of her before reaching the truck. Otherwise, she will instead accelerate, incentivizing the autonomous car to slow down, abort the overtaking maneuver, and merge behind her instead to pass the truck safely.

Note that we are not proposing that autonomous vehicles should in fact carry out this type of overtaking maneuver. The remarkable result here is in the planner's ability to reason about the different possible strategies given the scenario and objectives. Also note that in this and the other example scenarios, the roles of the human and the autonomous vehicle can easily be interchanged, allowing the autonomous vehicle to e.g. discourage others' potentially unsafe maneuvers.



(a) Low-level planner (b) Hierarchical planner: human lets autonomous vehicle cut in (c) Hierarchical planner: human cuts off autonomous vehicle

Fig. 3: Tactical and hierarchical planning in the truck cut-in scenario. (a) The tactical-only planner executes an unsafe last-second merge. (b) With enough speed difference, the hierarchical planner first accelerates to incentivize the human to slow down and then safely merges in front. (c) If there is little margin, the human has an incentive to accelerate preventing the maneuver.

C. In-Depth Analysis

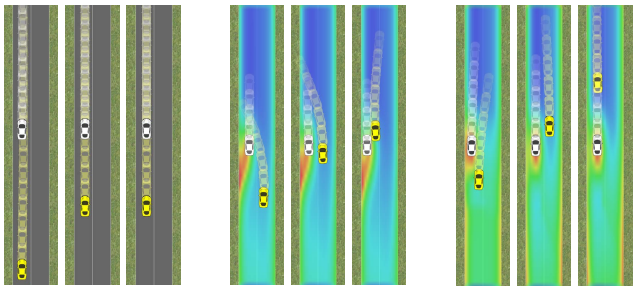
We now seek to shed light on *why* hierarchical planning obtains better performance than tactical alone. Is the strategic value merely lengthening the effective horizon, avoiding local or myopic optima, or is information structure important?

1) Hierarchical vs. long-horizon tactical planning: The hierarchical planning method provides the autonomous car with more information about the future via the strategic value of the long-term game, which guides the optimization to escape local optima. If those were the only benefits, extending the horizon of the tactical planner and re-initializing in different basins of attraction ought to perform similarly. We thus extend the horizon to 2 s (20 time steps) and perform multiple independent optimizations at each planning cycle, initialized

from diverse trajectories for each car: full-left steer, full-right steer, and straight steer (with acceleration input to maintain speed). This stronger tactical planner is unable to optimize in real time, unlike our other demonstrations, but is a good tool for analysis. Extension beyond 2 s was not tractable.

We tested this planner in the overtaking scenario alongside a human-driven car that is aware of the autonomous car's plan, which is this planner's assumed information structure. The planner still fails to complete the maneuver regardless of the initialization scheme and whether the influence term in [7] is used, resulting in the autonomous car remaining behind the human, as shown in Fig. 4. Moreover, we tested this planner against a human driver who maintains a constant slow speed of 24 m/s. In this case, the autonomous car brakes abruptly to avoid a collision and remains behind the human, at each time step expecting her to maximally accelerate for the next 1 s. Despite the longer horizon and more global optimization, this new tactical planner still assumes the wrong information structure, i.e. that the human knows the autonomous car's trajectory multiple seconds into the future. This causes poor performance when the human does not in fact adapt to the autonomous vehicle's plan ahead of time.

2) Information structure at the tactical level: When optimizing the autonomous car's trajectory at the tactical level, we used iterated local best response seeking a local open-loop Nash equilibrium between the vehicles' short-horizon trajectories. Conversely, the implicit differentiation proposed in [7], by which the autonomous planner estimates the influence of each local trajectory change on human's best response, is consistent with the local open-loop Stackelberg equilibrium concept, with the human as the *follower*. We observed that this latter approach resulted in more aggressive behavior in some situations, even when augmenting this tactical planner with the long-term strategic value. For example, in the hard merge scenario shown in Fig. 4, the hierarchical car attempted to merge into the left lane before fully overtaking the human, placing the burden on her to avoid an imminent collision. On the other hand, the traditional "pipeline" approach, in which the human's trajectory is predicted and fed to the planner as a moving obstacle, failed to overtake when used by itself, but succeeded in changing lanes and overtaking (comparably to the iterated best response scheme) when given the strategic value term.



(a) Long-horizon tactical-only planner

(b) Hierarchical planner with low-level influence estimate

(c) Hierarchical planner with predict-then-plan low-level scheme

Fig. 4: Study of alternative information structures. (a) In the overtaking scenario, the long-horizon tactical-only car accelerates, expecting the human to match its higher speed to avoid a collision. After the human speeds up, the autonomous car remains behind her. (b) Under the influence estimate [7] in the low-level trajectory gradient, the hierarchical car drives more aggressively in the merging scenario. (c) When augmented with the strategic value, the "pipeline" (predict-then-plan) low-level scheme is able to overtake.

The results suggest that, even in short horizons, assuming that the human can accurately anticipate and adapt to the autonomous vehicle's planned trajectory may lead to unsafe situations when the actual human driver fails to preemptively make way as expected. Running iterated local best response between trajectories or even assuming no short-term human adaptation at the tactical level seem to perform better as tactical schemes within our proposed hierarchical framework.

3) Confidence in Strategic Human Model: Finally, we discuss the effects of varying the autonomous planner's confidence in its high-level model of the human. Modeling the human as a Boltzmann noisily rational agent, we can naturally incorporate the planner's confidence in the human model via the inverse temperature parameter β in (5), as done in [23]. We can then compute different strategic values corresponding to varying levels of confidence in the human model. In the overtaking scenario, we observed that sufficiently lowering the inverse temperature parameter led to the autonomous vehicle choosing to remain behind the human car instead of attempting to overtake. A lower level of confidence in the human model discourages the autonomous car from overtaking because the human driver is more likely to act in an unexpected manner that may result in a collision.

V. Discussion

We have introduced a hierarchical trajectory planning formulation for an autonomous vehicle interacting with a human-driven vehicle on the road. To tractably reason about the mutual influence between the human and the autonomous system, our framework uses a lower-order approximate dynamical model solve a nonzero sum game with closed-loop feedback information. The value of this game is then used to inform the planning and predictions of the autonomous vehicle's low-level trajectory planner.

Even with a simplified dynamical model, solving the dynamic game will generally be computationally intensive. We note, however, that our high-level computation presents two key favorable characteristics for online usability. First, it is "massively parallel" in the sense that all states on the discretized grid may be updated simultaneously. The need for reliable real-time perception in autonomous driving has spurred the development of high-performance parallel computing hardware, which will directly benefit our method. Second, once computed, the strategic value can be readily stored as a look-up table, enabling fast access by the low-level trajectory planner. Of course, strategic values would then need to be pre-computed for a number of scenarios that autonomous vehicles might encounter.

We believe that our new framework can work in conjunction with and significantly enhance existing autonomous driving planners, allowing autonomous vehicles to more safely and efficiently interact with human drivers.

REFERENCES

[1] A. Carvalho, G. Palmieri, H. Tseng, et al. "Robust vehicle stability control with an uncertain driver model". 1239323 (2013).

[2] M. P. Vitus and C. J. Tomlin. "A probabilistic approach to planning and control in autonomous urban driving". *52nd IEEE Conference on Decision and Control* (2013).

[3] B. Luders, M. Kothari, and J. How. "Chance Constrained RRT for Probabilistic Robustness to Environmental Uncertainty". *AIAA Guidance, Navigation, and Control Conference* August (2010).

[4] C. Hermes, C. Wohler, K. Schenk, and F. Kummert. "Long-term vehicle motion prediction". *2009 IEEE Intelligent Vehicles Symposium* (2009).

[5] R. Felton. *Google's Self-Driving Cars Have Trouble With Basic Driving Tasks—Report*. `https://jalopnik.com/googles-self-driving-cars-have-trouble-with-basic-drivi-1828653280`. [Online; accessed 14 September 2018]. 2018.

[6] P. Trautman and A. Krause. "Unfreezing the robot: Navigation in dense, interacting crowds". *International Conference on Intelligent Robots and Systems (IROS)* (2010).

[7] D. Sadigh, S. Sastry, S. A. Seshia, and A. D. Dragan. "Planning for Autonomous Cars that Leverage Effects on Human Actions". *Robotics: Science and Systems (RSS)* (2016).

[8] D. Sadigh, S. Sastry, S. A. Seshia, and A. D. Dragan. "Information Gathering Actions over Human Internal State". *International Conference on Intelligent Robots and Systems (IROS)* (2016).

[9] A. Liniger and J. Lygeros. "A non-cooperative game approach to autonomous racing". *arXiv preprint arXiv:1712.03913* (2017).

[10] A. Dreves and M. Gerdts. "A generalized Nash equilibrium approach for optimal control problems of autonomous cars". *Optimal Control Applications and Methods* 39.1 (2018).

[11] R. Spica, D. Falanga, E. Cristofalo, et al. "A real-time game theoretic planner for autonomous two-player drone racing". *Robotics: Science and Systems* (2018).

[12] T. Basar and A. Haurie. "Feedback Equilibria in Differential Games with Structural and Modal Uncertainties". *Advances in Large Scale Systems* 1 (1984).

[13] M. Simaan and J. B. Cruz Jr. "On the Stackelberg strategy in nonzero-sum games". *Journal of Optimization Theory and Applications* 11.5 (1973).

[14] T. B. Sheridan. "Three Models of Preview Control". *IEEE Transactions on Human Factors in Electronics* HFE-7.2 (1966).

[15] H. Peng and M. Tomizuka. "Preview Control for Vehicle Lateral Guidance in Highway Automation". *Journal of Dynamic Systems, Measurement, and Control* (1993).

[16] C. C. MacAdam. "Understanding and Modeling the Human Driver". *Vehicle System Dynamics* 40.1-3 (2003).

[17] D. Silver, A. Huang, C. J. Maddison, et al. "Mastering the game of Go with deep neural networks and tree search". *Nature* 529.7587 (2016).

[18] B. D. Ziebart and A. Maas. "Maximum entropy inverse reinforcement learning". *Twenty-Second Conference on Artificial Intelligence (AAAI)* (2008).

[19] C. Finn, S. Levine, and P. Abbeel. "Guided Cost Learning: Deep Inverse Optimal Control via Policy Optimization". 48 (2016).

[20] E. Schmerling, K. Leung, W. Vollprecht, and M. Pavone. "Multimodal Probabilistic Model-Based Planning for Human-Robot Interaction". *International Conference on Robotics and Automation (ICRA)* (2018).

[21] J. F. Fisac and S. S. Sastry. "The Pursuit-Evasion-Defense Differential Game in Dynamic Constrained Environments". *Conference on Decision and Control (CDC)* (2015).

[22] M. Chen, S. Bansal, J. F. Fisac, and C. J. Tomlin. "Robust Sequential Path Planning Under Disturbances and Adversarial Intruder". *IEEE Transactions on Control Systems Technology* (2019).

[23] J. F. Fisac, A. Bajcsy, S. L. Herbert, et al. "Probabilistically Safe Robot Planning with Confidence-Based Human Predictions". *Robotics: Science and Systems*. 2018.

[24] M. Simaan and J. B. Cruz Jr. "Additional aspects of the Stackelberg strategy in non-zero sum games". *Journal of Optimization Theory and Applications* 11.6 (1973).

[25] K. Waugh, B. D. Ziebart, and J. A. Bagnell. "Inverse Correlated Equilibrium for Matrix Games". *Advances in Neural Information Processing Systems (NIPS)* (2010).

[26] B. Kenwell. *Nvidia Is Still Surging Toward Autonomous Driving Success*. `https://www.thestreet.com/technology/nvidia-is-still-surging-toward-autonomous-driving-success-14690750`. [Online; accessed 15 September 2018]. 2018.

[27] S. Pendleton, H. Andersen, X. Du, et al. "Perception, Planning, Control, and Coordination for Autonomous Vehicles". *Machines* 5.1 (2017).

[28] G. Andrew and J. Gao. "Scalable training of L 1-regularized log-linear models". *International Conference on Machine Learning (ICML)* (2007).

[29] L. J. Ratliff, S. A. Burden, and S. S. Sastry. "On the characterization of local Nash equilibria in continuous games". *IEEE Transactions on Automatic Control* 61.8 (2016).