Autonomous Driving with Interpretable Goal Recognition and Monte Carlo Tree Search

Cillian Brewitt*†, Stefano V. Albrecht*†, John Wilhelm*†,
Balint Gyevnar*†, Francisco Eiras*, Mihai Dobre*, Subramanian Ramamoorthy*†
*FiveAI Ltd., UK, {firstname.lastname}@five.ai
†School of Informatics, University of Edinburgh, UK

***Abstract*—The ability to predict the intentions and driving trajectories of other vehicles is a key problem for autonomous driving. We propose an integrated planning and prediction system which leverages the computational benefit of using a finite space of maneuvers, and extend the approach to planning and prediction of *sequences* of maneuvers via rational inverse planning to recognise the *goals* of other vehicles. Goal recognition informs a Monte Carlo Tree Search (MCTS) algorithm to plan optimal maneuvers for the ego vehicle. Our system constructs plans which are explainable by means of *rationality*. Evaluation in simulations of four urban driving scenarios demonstrate the system's ability to robustly recognise the goals of other vehicles while generating near-optimal plans.**

I. Introduction

The ability to predict the intentions and driving trajectories of other vehicles is a key problem for autonomous driving [19]. This problem is significantly complicated by the requirement to make fast and accurate predictions based on limited observation data originating from a dynamically evolving environment with coupled multi-agent interactions.

A standard approach in autonomous driving to make prediction tractable in such conditions is to assume that agents use one of a finite number of distinct high-level maneuvers such as lane-follow, lane-change, turn, stop, etc [1, 2, 9, 10, 13, 14, 20, 25]. A classifier is used to detect a vehicle's current executed maneuver based on its observed driving trajectory. Such methods are limited to predictions on the timescales of the detected maneuvers. An alternative approach is to specify a finite set of possible *goals* for each other vehicle (such as road exit points) and to plan a full trajectory to each goal from the vehicle's observed local state [5, 11, 26]. While this approach can generate longer-term predictions, it is limited by the fact that the generated trajectories must be relatively closely followed by a vehicle to yield high-confidence predictions.

Recent methods based on deep learning have shown promising results for trajectory prediction in autonomous driving [7, 8, 15, 18, 23, 24]. In our view, one of the most significant limitations of this class of methods is the difficulty in extracting interpretable predictions in a form that is amenable to efficient integration with planning methods that effectively represent multi-dimensional and hierarchical task objectives.

S.A. is supported by a personal fellowship from the Royal Society. C.B., J.W. and B.G were interns at FiveAI with partial financial support from the Royal Society and UKRI.

Our starting point[1] is that in order to predict the future maneuvers of a vehicle, we must reason about *why* – that is, to what end – the vehicle performed its current and past maneuvers. Knowledge of the goals of other vehicles enables long-term prediction of their future maneuvers and trajectories, which facilitates planning over extended timescales. One of the most important advantages of this approach is that we can generate predictions and plans which are intuitively explainable, by means of rationality.

We propose an integrated planning and prediction system which leverages the computational advantages of using a finite space of maneuvers, but extends the approach to planning and prediction of *sequences* (i.e., plans) of maneuvers. We achieve this via a novel integration of rational inverse planning [4, 17] to recognise the goals of other vehicles, with Monte Carlo Tree Search (MCTS) [6] to plan optimal maneuvers for the ego vehicle. Rather than matching plans directly as in prior work [5, 11], our approach instead evaluates the extent to which an observed trajectory is rational for a given goal, providing robustness with respect to variability in trajectories.

We evaluate our system in simulations of four urban driving scenarios, including junctions, roundabout entry, and dense lane merging. We extract intuitive explanations for the recognised goals and maneuver predictions in each scenario which justify the system's decisions.

II. Preliminaries and Problem Definition

Let $\mathcal{I}$ be the set of vehicles in the local neighbourhood. At time t, each vehicle $i \in \mathcal{I}$ is in a local state $s_t^i \in \mathcal{S}^i$, receives a local observation $o_t^i \in \mathcal{O}^i$, and can choose an action $a_t^i \in \mathcal{A}^i$. We write $s_t \in \mathcal{S} = (s_t^1, s_t^2, ...)$ for the joint state and $s_{a:b}$ for the tuple $(s_a, ..., s_b)$, and similarly for $o_t \in \mathcal{O}, a_t \in \mathcal{A}$. Observations depend on the joint state via $p(o_t^i|s_t)$, and actions depend on the observations via $p(a_t^i|o_{1:t}^i)$. A local state contains a vehicle's pose, velocity, and acceleration; an observation contains the states of nearby vehicles; and an action controls the vehicle's steering and acceleration. The probability of a

[1]An extended version of our paper is available which gives a more detailed description of our method: https://arxiv.org/abs/2002.02277

sequence of joint states $s_{1:n}$, $n \geq 1$, is given by

$$p(s_{1:n}) = \prod_{t=1}^{n-1} \int_{\mathcal{O}} \int_{\mathcal{A}} p(o_t|s_t)p(a_t|o_{1:t})p(s_{t+1}|s_t, a_t)\, do_t\, da_t \tag{1}$$

where $p(s_{t+1}|s_t, a_t)$ defines the joint vehicle dynamics, and we assume independent local observations and actions, $p(o_t|s_t) = \prod_i p(o_t^i|s_t)$ and $p(a_t|o_{1:t}) = \prod_i p(a_t^i|o_{1:t}^i)$. Vehicles react to other vehicles via their local observations $o_{1:n}^i$.

We define the planning problem as finding an optimal policy π^* which selects the actions for the ego vehicle, ε, to achieve a specified goal, $\mathcal{G}^\varepsilon$, while optimising the driving trajectory via a defined reward function. Here, a policy is a function $\pi : (\mathcal{O}^\varepsilon)^* \mapsto \mathcal{A}^\varepsilon$ which maps an observation sequence $o_{1:n}^\varepsilon$ to an action a_t^ε. A goal can be any subset of local states, $\mathcal{G}^\varepsilon \subset \mathcal{S}^\varepsilon$, but here we focus on goals that specify target locations. Formally, define

$$\Omega_n = \left\{ s_{1:n} \,\middle|\, s_n^\varepsilon \in \mathcal{G}^\varepsilon \wedge \nexists m < n : s_m^\varepsilon \in \mathcal{G}^\varepsilon \right\} \tag{2}$$

where $s_n^\varepsilon \in \mathcal{G}^\varepsilon$ means that s_n^ε satisfies $\mathcal{G}^\varepsilon$. The second condition in (2) ensures that $\sum_{n=1}^{\infty} \int_{\Omega_n} p(s_{1:n}) ds_{1:n} \leq 1$ for any policy π, which is needed for soundness of the sum in (3). The problem is to find π^* such that

$$\pi^* \in \arg\max_\pi \sum_{n=1}^{\infty} \int_{\Omega_n} p(s_{1:n}) R^\varepsilon(s_{1:n})\, ds_{1:n} \tag{3}$$

where $R^i(s_{1:n})$ is the reward of $s_{1:n}$ for vehicle i.

III. Method

A. System Overview

Our general approach relies on two assumptions: (1) each vehicle seeks to reach some (unknown) goal location from a set of possible goals, and (2) each vehicle follows a plan generated from a finite library of defined maneuvers.

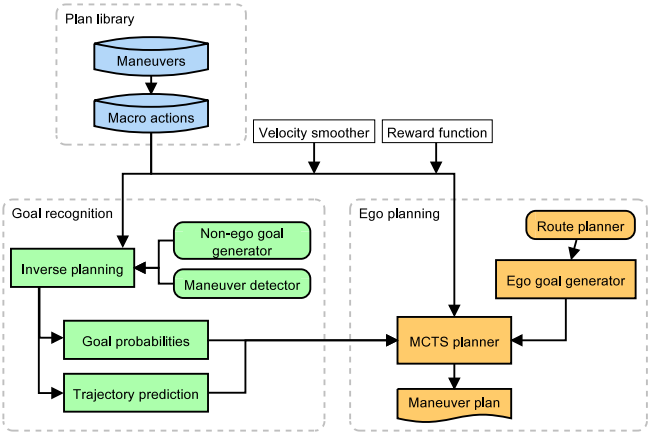


Fig. 1: System overview.

Our system (Figure 1) approximates the optimal policy π^* as follows: For each other vehicle, enumerate its possible goals and inversely plan for that vehicle to each goal, giving the probabilities and predicted trajectories to the goals. The resulting goal probabilities and trajectories inform the simulation process of a Monte Carlo Tree Search (MCTS) algorithm to generate an optimal maneuver plan for the ego vehicle. In order to keep the required search depth shallow and hence efficient, both inverse planning and MCTS plan over macro actions which flexibly concatenate maneuvers using context information.

B. Maneuvers

We assume that at any time, each vehicle is executing one of the following maneuvers: *lane-follow, lane-change-left/right, turn-left/right, give-way, stop.* Each maneuver ω specifies applicability and termination conditions. For example, lane-change-left is only applicable if there is a lane in same driving direction on the left of the vehicle. The maneuver terminates if the state satisfies the termination condition.

If applicable, a maneuver specifies a local trajectory $\hat{s}_{1:n}^i$ to be followed by the vehicle, which includes a reference path based on road topology and target velocities along the path. To obtain a feasible trajectory across maneuvers for a vehicle, we apply a smoothing operation[2] which optimises the target velocities in a given trajectory.

C. Macro Actions

Macro actions specify common sequences of maneuvers, and they automatically set the free parameters in maneuvers based on context information. Our system uses the following macro actions: *Continue, Continue to next exit, Change left/right, Exit left/right, Stop.* Similarly to maneuvers, each macro action has applicability and termination conditions.

D. Goal Recognition

By assuming that each vehicle $i \in \mathcal{I}$ seeks to reach one of a finite number of possible goal locations $\mathcal{G}^{i,1}, \mathcal{G}^{i,2}, ...$, using plans constructed from our defined macro actions, we use the framework of rational inverse planning [4, 17] to compute a Bayesian posterior distribution over vehicle i's goals at time t,

$$p(\mathcal{G}^i|s_{1:t}) \propto L(s_{1:t}|\mathcal{G}^i)p(\mathcal{G}^i) \tag{4}$$

where $L(s_{1:t}|\mathcal{G}^i)$ is the likelihood of i's observed trajectory given goal $\mathcal{G}^i$, and $p(\mathcal{G}^i)$ specifies the prior probability of $\mathcal{G}^i$.

The likelihood is a function of the reward difference between two plans: the reward $\hat{r}$ of the optimal trajectory from i's initial observed state s_1^i to goal $\mathcal{G}^i$ after velocity smoothing, and the reward $\bar{r}$ of the trajectory which follows the observed trajectory until time t and then continues optimally to goal $\mathcal{G}^i$, with smoothing applied only to the trajectory after t. Then, the likelihood is defined as

$$L(s_{1:t}|\mathcal{G}^i) = \exp(-\beta(\bar{r} - \hat{r})) \tag{5}$$

where β is a scaling parameter. This definition assumes that vehicles behave *rationally* by driving optimally to achieve goals, but allows for a degree of deviation.

[2]Full details available at https://arxiv.org/abs/2002.02277

1) **Goal Generation:** A heuristic function is used to generate the set of possible goals for vehicle i based on its location and context information such as road layout and traffic rules. In our system, we include some static goals, such as the visible end of the current road and connecting roads. We also include dynamic goals, such as stopping goals which model a vehicle's intention to allow the ego vehicle to merge.

2) **Maneuver Detection:** We assume a module which computes probabilities over current maneuvers, $p(\omega^i)$, for each vehicle i. One option is Bayesian changepoint detection algorithms such as CHAMP [16], as used in MPDM [10]. The details of maneuver detection are outside the scope of our paper and in our experiments we use a simulated detector.

3) **Inverse Planning:** Inverse planning is done using A* search [12] over macro actions. A* starts after completing the current maneuver ω^i which produces the initial trajectory $\hat{s}_{1:\tau}$. Each search node q corresponds to a state $s \in \mathcal{S}$, with initial node at state $\hat{s}_\tau$. A* chooses the next macro action leading to node q' which has lowest total cost to goal $\mathcal{G}^i$, given by $f(q') = l(q') + h(q')$. The cost $l(q')$ to reach node q' is given by the driving time from i's location in the initial search node to its location in q', following the trajectories returned by the macro actions leading to q'. The cost heuristic $h(q')$ to estimate remaining cost from q' to goal $\mathcal{G}^i$ is given by the driving time from i's location in q' to goal via straight line at speed limit.

As different current maneuvers may hint at different goals, we perform inverse planning for each possible current maneuver for which $p(\omega^i) > 0$. Thus, each current maneuver produces its own posterior probabilities over goals, denoted by $p(\mathcal{G}^i \mid s_{1:t}, \omega^i)$.

4) **Trajectory Prediction:** Our system predicts multiple plausible trajectories for a given vehicle and goal, by continuing to run A* search after the optimal trajectory has been found, up to some fixed number of plans. Given a set of computed trajectories $\{\hat{s}^{i,k}_{1:n} | \omega^i, \mathcal{G}^i\}_{k=1..K}$ to goal $\mathcal{G}^i$ with initial maneuver ω^i and associated reward $r_k = R^i(\hat{s}^{i,k}_{1:n})$ after smoothing, we compute a distribution over the trajectories:

$$p(\hat{s}^{i,k}_{1:n}) = \eta \, \exp(\gamma \, r_k) \tag{6}$$

where γ is a scaling factor (we use $\gamma = 1$) and η is a normaliser.

E. Ego Vehicle Planning

To compute an optimal plan for the ego vehicle, we use the goal probabilities and trajectory predictions to inform a Monte Carlo Tree Search (MCTS) algorithm [6]. MCTS combines the statistical back-propagation operators used in temporal-difference reinforcement learning [21] with a dynamic tree expansion to focus the search on the current state.

The algorithm performs a number of simulations $\hat{s}_{t:n}$, starting in the current state $\hat{s}_t = s_t$ down to some fixed search depth or until a goal state is reached. At the start of each simulation, for each other vehicle, we first sample a current maneuver, then goal, and then trajectory for the vehicle using the associated probabilities (cf. Section III-D). Exploration-exploitation trade-off in the tree level of the algorithm is achieved by employing UCB [3] for selecting a macro action. After selecting a macro action, the state in current search node is forward-simulated based on the trajectory generated by the macro action and the sampled trajectories of other vehicles, resulting in a partial trajectory $\hat{s}_{\tau:\iota}$ and new search node q' with state $\hat{s}_\iota$. If the ego vehicle collided, or the search reached its maximum depth d_{max} without achieving the goal, we set $r \leftarrow r_{term}$, where r_{term} represents the reward received in case of failure.

The reward r is back-propagated through search branches (q, ω, q') that generated the simulation, using a 1-step off-policy update function (similar to Q-learning [22]) defined by

$$Q(q,\mu) \leftarrow Q(q,\mu) + \begin{cases} \delta^{-1}[r - Q(q,\mu)] \text{ if } q \text{ leaf node, else} \\ \delta^{-1}[\max_{\mu'} Q(q',\mu') - Q(q,\mu)] \end{cases} \tag{7}$$

where δ is the number of times that macro action μ has been selected in q. After the simulations are completed, the algorithm selects the best macro action for execution in s_t from the root node, $\arg\max_\mu Q(root, \mu)$. We support two simulation modes for maneuvers: open-loop and closed-loop. Closed-loop simulation makes use of feedback from sensors, and uses a combination of proportional control and adaptive cruise control (ACC) to control steering and acceleration. Open-loop simulation sets the vehicle's position and velocity directly as specified in the generated trajectory.

IV. Evaluation

We evaluate our system in simulations of four urban driving scenarios with diverse scenario initialisations. We show that:

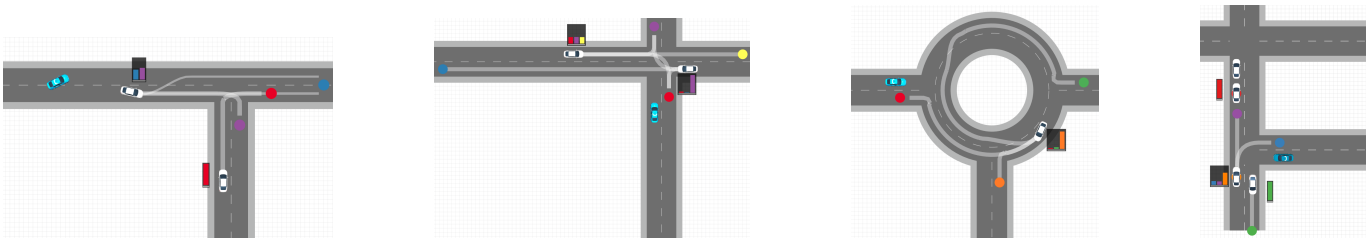
- Our method correctly recognises the goals of other vehicles
- Goal recognition leads to improved driving behaviour
- We can extract intuitive explanations for the recognised goals and predictions, to justify the system's decisions

A snapshot from each of the scenarios is shown in Figure 2. For each scenario, we generate 100 instances with randomly offset initial positions (offset $\sim [-10, +10]$ metres) and initial speed sampled from range $[5, 10]$ m/s for each vehicle.

A. Baselines & Parameters

We compare the following versions of our system. **Full:** full system using goal recognition and MCTS. **MAP:** like Full, but MCTS uses only the most probable goal and trajectory for each vehicle. **CVel:** MCTS without goal recognition, replaced by constant-velocity lane-following prediction. **CVel-Avg:** Like CVel, but predicts velocity to be the average velocity over the previous 2 seconds. **Cons:** like CVel-Avg, but using a conservative *give-way* maneuver which waits until all oncoming vehicles on priority lanes have passed.

For each other vehicle and generated goal, we generate up to 3 predicted trajectories. We simulate noisy detection of current maneuvers for each other vehicle by giving 0.90 probability to correct current maneuver and the rest uniformly to other maneuvers. MCTS is run at a frequency of 1 Hz, performs $D = 30$ simulations, with a maximum search depth of $d_{max} = 5$. Rewards for collision and maximum search depth are set to $r_{term} = -1$. Prior probabilities for goals are uniform.



(a) Scenario 1 (S1) (b) Scenario 2 (S2) (c) Scenario 3 (S3) (d) Scenario 4 (S4)

Fig. 2: In each scenario, the ego vehicle is coloured blue. (a) **S1**: Ego's goal is the blue goal. Vehicle V_1 is on the ego's road, V_1 changes from left to right lane, biasing the ego prediction towards the belief that V_1 will exit, since a lane change would be irrational if V_1's goal was to go east. As exiting will require a significant slowdown, the ego decides to switch lanes to avoid being slowed down too. (b) **S2**: Vehicle V_1 is approaching the junction from west, and vehicle V_2 approaching it from east. As V_2 approaches the junction, slows down and waits to take a turn, the ego's belief that V_2 will turn right increases significantly, since it would be irrational to stop if the goal was to turn left or go straight. Since ego recognised V_2's goal is to go north, it predicts that V_2 will wait until V_1 has passed, giving the ego an opportunity to enter the road. (c) **S3**: As V_1 changes from the inside to the outside lane of the roundabout and decreases its speed, it significantly biases the ego prediction towards the belief that V_1 will leave in the next exit since that is the rational course of action for that goal. This encourages the ego to enter the roundabout while V_1 is still in roundabout. (d) **S4**: With two vehicles stopped at the junction at a traffic light, vehicle V_1 is approaching them from behind, and vehicle V_2 is crossing in the opposite direction. When V_1 reaches zero velocity this reveals a stopping goal in its current position, shifting the distribution towards it, since stopping is not rational for north/east goals. The interpretation is that V_1 wants the ego to merge in. Given the recognised goal, the ego can merge onto the road in front of V_1

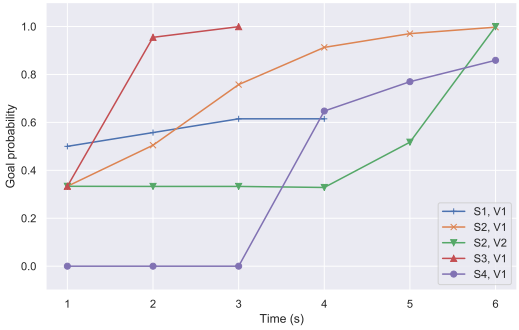


Fig. 3: Evolution of probability given to correct goal for selected vehicles in four scenario instances.

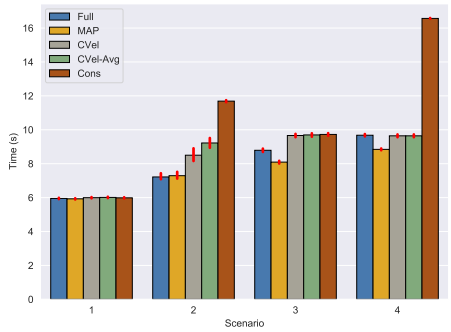


Fig. 4: Average driving time (seconds) required to complete scenario instances, with 95% confidence intervals.

B. Results

Figure 2 shows a snapshot from each scenario instance using our Full system. The bar plots give the goal probabilities for each other vehicle associated with their most probable current maneuver. For each goal, we show the most probable trajectory prediction from vehicle to goal, with thickness proportional to its probability. We extract intuitive explanations for the goal recognition and maneuver predictions in each scenario, which are given in the figure captions.

Figure 3 shows the evolution of probability assigned to correct goal over time, in four scenario instances. In all scenario instances, the probability approaches the correct goal as other goals are being ruled out by means of rationality principles. Figure 4 shows the average times and 95% confidence intervals required by each baseline to complete a scenario instance.

(S1) All baselines switch lanes in response to V_1 switching lanes. Full and MAP anticipate V_1's slowdown earlier than other baselines due to inverse planning, allowing them to switch lanes slightly earlier. CVel, CVel-Avg and Cons only switch lanes once V_1 already started to slow down, and are unable to explain V_1's behaviour.

(S2) Cons requires substantially more time to complete the scenario since it waits for V_2 to clear the lane, which in turn must wait for V_1 to pass. Full and MAP anticipate this behaviour, allowing them to safely enter the road earlier. CVel and CVel-Avg require more time as the ego must wait for V_2 to reach near-zero velocity for entry to be deemed safe.

(S3) CVel, CVel-Avg and Cons require more time to complete the scenario. Here, the constant-velocity prediction in CVel, and the waiting for actual clearance as in Cons, amount to approximately equal time of entry for ego vehicle. Full and MAP are able to enter earlier as they recognise V_1's goal, which is to exit the roundabout. MAP enters earlier than Full since it fully commits to the most probable goal for V_1, while Full exhibits more cautious behaviour due to residual uncertainty about V_1's goal hypothetically leading to crashes.

(S4) Cons must wait until V_1 decides to close the gap, after which the ego can enter the road, hence requiring more time. Full and MAP recognise V_1's goal and can enter safely. MAP enters earlier than Full once it realises the waiting goal of V_1 which has the highest posterior probability so it fully commits to this goal, while Full waits longer due to residual uncertainty

about the goals of V_1. CVel and CVel-Avg produce the same behaviour as Full based on constant velocity of V_1, but cannot explain its waiting behaviour.

V. Conclusion

We proposed an autonomous driving system which integrates planning and prediction over extended horizons, by leveraging the computational benefit of utilising a finite maneuver library. Prediction over extended horizons is made possible by recognising the goals of other vehicles via a process of rational inverse planning. Our evaluation showed that the system robustly recognises the goals of other vehicles in diverse urban driving scenarios, resulting in improved decision making while allowing for intuitive interpretations of the predictions to justify the system's decisions. While this work focused on prediction of other vehicles, our system could be extended to include prediction of other traffic participants such as cyclists, or applied to other human/robot interaction domains.

References

[1] S. Albrecht and P. Stone. Autonomous agents modelling other agents: A comprehensive survey and open problems. *Artificial Intelligence*, 258:66–95, 2018.

[2] S. Albrecht, J. Crandall, and S. Ramamoorthy. Belief and truth in hypothesised behaviours. *Artificial Intelligence*, 235:63–94, 2016.

[3] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3):235–256, 2002.

[4] C. Baker, R. Saxe, and J. Tenenbaum. Action understanding as inverse planning. *Cognition*, 113(3):329–349, 2009.

[5] T. Bandyopadhyay, K. S. Won, E. Frazzoli, D. Hsu, W. S. Lee, and D. Rus. Intention-aware motion planning. In *Algorithmic Foundations of Robotics X*, pages 475–491. Springer, 2013.

[6] C. Browne, E. Powley, D. Whitehouse, S. Lucas, P. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton. A survey of Monte Carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(1):1–43, 2012.

[7] S. Casas, W. Luo, and R. Urtasun. IntentNet: learning to predict intention from raw sensor data. In *Conference on Robot Learning*, pages 947–956, 2018.

[8] Y. Chai, B. Sappm, M. Bansal, and D. Anguelov. MultiPath: multiple probabilistic anchor trajectory hypotheses for behavior prediction. In *Proceedings of the 3rd Conference on Robot Learning*, 2019.

[9] C. Dong, J. M. Dolan, and B. Litkouhi. Smooth behavioral estimation for ramp merging control in autonomous driving. In *IEEE Intelligent Vehicles Symposium*, pages 1692–1697. IEEE, 2018.

[10] E. Galceran, A. Cunningham, R. Eustice, and E. Olson. Multipolicy decision-making for autonomous driving via changepoint-based behavior prediction: Theory and experiment. *Autonomous Robots*, 41(6):1367–1382, 2017.

[11] J. Hardy and M. Campbell. Contingency planning over probabilistic obstacle predictions for autonomous road vehicles. *IEEE Transactions on Robotics*, 29(4):913–929, 2013.

[12] P. Hart, N. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. In *IEEE Transactions on Systems Science and Cybernetics*, volume 4, pages 100–107, July 1968.

[13] C. Hubmann, M. Becker, D. Althoff, D. Lenz, and C. Stiller. Decision making for autonomous driving considering interaction and uncertain prediction of surrounding vehicles. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 1671–1678. IEEE, 2017.

[14] C. Hubmann, J. Schulz, M. Becker, D. Althoff, and C. Stiller. Automated driving in uncertain environments: Planning with interaction and uncertain maneuver prediction. *IEEE Transactions on Intelligent Vehicles*, 3(1): 5–17, 2018.

[15] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. Torr, and M. Chandraker. DESIRE: distant future prediction in dynamic scenes with interacting agents. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 336–345, 2017.

[16] S. Niekum, S. Osentoski, C. Atkeson, and A. Barto. Online Bayesian changepoint detection for articulated motion models. In *IEEE International Conference on Robotics and Automation*. IEEE, 2015.

[17] M. Ramírez and H. Geffner. Probabilistic plan recognition using off-the-shelf classical planners. In *24th AAAI Conference on Artificial Intelligence*, pages 1121–1126, 2010.

[18] N. Rhinehart, R. McAllister, K. Kitani, and S. Levine. PRECOG: prediction conditioned on goals in visual multi-agent settings. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2821–2830, 2019.

[19] W. Schwarting, J. Alonso-Mora, and D. Rus. Planning and decision-making for autonomous vehicles. *Annual Review of Control, Robotics, and Autonomous Systems*, 1:187–210, 2018.

[20] W. Song, G. Xiong, and H. Chen. Intention-aware autonomous driving decision-making in an uncontrolled intersection. *Mathematical Problems in Engineering*, 2016, 2016.

[21] R. Sutton and A. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2018.

[22] C. Watkins and P. Dayan. Q-learning. *Machine Learning*, 8(3-4):279–292, 1992.

[23] M. Wulfmeier, D. Z. Wang, and I. Posner. Watch this: Scalable cost-function learning for path planning in urban environments. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2089–2095. IEEE, 2016.

[24] Y. Xu, T. Zhao, C. Baker, Y. Zhao, and Y. N. Wu. Learning trajectory prediction with continuous inverse optimal

control via Langevin sampling of energy-based models. *arXiv preprint arXiv:1904.05453*, 2019.

[25] B. Zhou, W. Schwarting, D. Rus, and J. Alonso-Mora. Joint multi-policy behavior estimation and receding-horizon trajectory planning for automated urban driving. In *IEEE International Conference on Robotics and Automation*, pages 2388–2394. IEEE, 2018.

[26] B. Ziebart, N. Ratliff, G. Gallagher, C. Mertz, K. Peterson, J. Bagnell, M. Hebert, A. Dey, and S. Srinivasa. Planning-based prediction for pedestrians. pages 3931–3936, 12 2009. doi: 10.1109/IROS.2009.5354147.