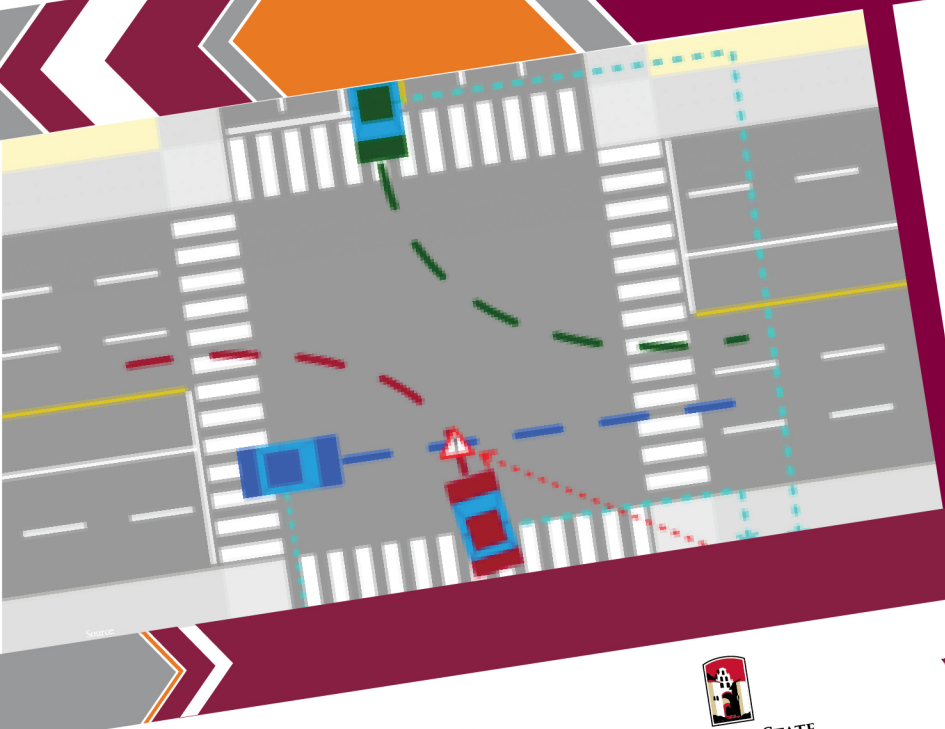
Prediction of Vehicle Trajectories at Intersections Using Inverse Reinforcement Learning

May 2021 | Final Report



SAFE-D
SAFETY THROUGH DISRUPTION

Texas A&M Transportation Institute

San Diego State University
Leadership Starts Here

VT

Virginia Tech Transportation Institute
Virginia Tech

Disclaimer

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated in the interest of information exchange. The report is funded, partially or entirely, by a grant from the U.S. Department of Transportation's University Transportation Centers Program. However, the U.S. Government assumes no liability for the contents or use thereof.

TECHNICAL REPORT DOCUMENTATION PAGE

1. Report No. SDSU-01-01	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Predicting Vehicle Trajectories at Intersections using Advanced Machine Learning Techniques		5. Report Date May 2021 6. Performing Organization Code:
7. Author(s) Mohammad Sadegh Jazayeri (SDSU) Arash Jahangiri (SDSU)* Sahar Ghanipoor Machiani (SDSU)		8. Performing Organization Report No. SDSU-01-01
9. Performing Organization Name and Address: San Diego State University 5500 Campanile Dr San Diego, CA 92182 USA		10. Work Unit No. 11. Contract or Grant No. 69A3551747115/SDSU-01-01
12. Sponsoring Agency Name and Address Office of the Secretary of Transportation (OST) U.S. Department of Transportation (US DOT)		13. Type of Report and Period Final Research Report 14. Sponsoring Agency Code

15. Supplementary Notes
This project was funded by the Safety through Disruption (Safe-D) National University Transportation Center, a grant from the U.S. Department of Transportation – Office of the Assistant Secretary for Research and Technology, University Transportation Centers Program.

16. Abstract
The ability to accurately predict vehicle trajectories is essential in infrastructure-based safety systems that aim to identify critical events such as near-crash situations and traffic violations. In a connected environment, important information about these critical events can be communicated to road users or the infrastructure to avoid or mitigate potential crashes. Intersections require special attention in this context because they are hotspots for crashes and involve numerous and complex interactions between road users. In this project, we developed an advanced machine learning method for trajectory prediction using B-spline curve representations of vehicle trajectories and Inverse Reinforcement Learning (IRL). B-spline curves were used to represent vehicle trajectories, and a neural network model was trained to predict the coefficients of these curves. Small perturbations of these predicted coefficients were used to create candidate trajectories. These candidate trajectories were then ranked according to a reward function that was obtained by training an IRL model on the (spline smoothed) vehicle trajectories and the surroundings of the vehicles. In our experiments we found that the neural network model outperforms a Kalman filter baseline and the addition of the IRL ranking module further improves the performance of the overall model.

17. Key Words Trajectory prediction, connected vehicles, machine learning	18. Distribution Statement No restrictions. This document is available to the public through the Safe-D National UTC website, as well as the following repositories: VTechWorks, The National Transportation Library, The Transportation Library, Volpe National Transportation Systems Center, Federal Highway Administration Research Library, and the National Technical Reports Library.

19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 28	22. Price $0

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

Abstract

The ability to accurately predict vehicle trajectories is essential in infrastructure-based safety systems that aim to identify critical events such as near-crash situations and traffic violations. In a connected environment, important information about these critical events can be communicated to road users or the infrastructure to avoid or mitigate potential crashes. Intersections require special attention in this context because they are hotspots for crashes and involve numerous and complex interactions between road users. In this project, we developed an advanced machine learning method for trajectory prediction using B-spline curve representations of vehicle trajectories and Inverse Reinforcement Learning (IRL). B-spline curves were used to represent vehicle trajectories, and a neural network model was trained to predict the coefficients of these curves. Small perturbations of these predicted coefficients were used to create candidate trajectories. These candidate trajectories were then ranked according to a reward function that was obtained by training an IRL model on the (spline smoothed) vehicle trajectories and the surroundings of the vehicles. In our experiments we found that the neural network model outperforms a Kalman filter baseline and the addition of the IRL ranking module further improves the performance of the overall model.

Acknowledgements

This project was funded by the Safety through Disruption (Safe-D) National University Transportation Center, a grant from the U.S. Department of Transportation – Office of the Assistant Secretary for Research and Technology, University Transportation Centers Program. Special thanks are extended to Bin Ren who served as the Subject Matter Expert and provided suggestions for the successful completion of this report.

Table of Contents

List of Figures

List of Tables

Introduction

Trajectory prediction involves forecasting the path a vehicle is going to take given its past trajectory and surroundings. Trajectory prediction has applications in surrogate safety analysis [1], road safety evaluation, and infrastructure-based safety systems [2]. Reliable trajectory prediction is of critical importance for advanced driver assistance systems [3][4][5] and autonomous vehicles (AVs) [6][3][7]. Additionally, accurate trajectory prediction would enable more realistic intersection simulations that better conform to the reality of human driving, allowing for better safety assessments [8].

Vehicle trajectory prediction is of particular interest at intersections where a great number of conflicts between road users could increase the likelihood of accidents [11]. According to the National Highway Traffic Safety Administration, between 2014 and 2018 about 40% of all crashes and 24% of fatal crashes occurred at intersections. With the advent of smart cities and smart vehicles, infrastructure-to-vehicle and vehicle-to-vehicle communications will be possible. In conjunction with a trajectory prediction system, these advances in vehicle and infrastructure technology will enable us to enhance the safety of intersections by predicting collisions [12][13] and risky driving behavior [14] (e.g., red light running) and deploying countermeasures to help avoid or mitigate crashes, such as early crash warnings [15][16][17][18][19] or real-time signal timing adjustments [20].

Being able to project vehicles' trajectories into the future is also important in automated driving applications; as long as AVs share roads with human-driven vehicles, they need to know how human drivers act in different situations and they must also behave in ways that conform to human drivers' expectation of other vehicles. That is, they must respond like other human drivers. It is therefore important that AVs have a model of vehicle motion in different situations, including at intersections.

When cast as a control problem (i.e., a problem of finding the correct control behavior), solving trajectory prediction is equivalent to training a model to drive like human drivers. This enables applications where human-like driving is desired. It is partly related to the problem of vehicle tracking (i.e., the problem of identifying and following the motion of vehicles in a video feed). While vehicle tracking deals with identifying the current motion of vehicles, trajectory prediction deals with predicting their future movements. The data required for trajectory prediction is the output of solving the vehicle tracking problem. In this work, we focus solely on the prediction problem, as there are several commercial tracking solutions available from companies such as Brisk Synergies [9] and Miovision [10].

A wide range of approaches have been used in tackling the trajectory prediction problem, from models that assume the vehicle will maintain its velocity or acceleration and (rate of change of) heading for the duration for which trajectory prediction is going to be performed [21], to those that try to capture more of the complexities of vehicle motion by modeling different maneuvers but still disregard the influence of other vehicles [22], to models that take the interactions between traffic actors into account when predicting the future motion of vehicles [23]. The tools used in developing these approaches are also quite varied and include Kalman filters [17], hidden Markov models [24], Gaussian processes [22], Bayesian networks [16], Gaussian mixture models [11], and neural networks [6]. These studies all formulate the problem of trajectory prediction as a prediction task, which is to say they directly predict the entire future trajectory of the vehicle, but it can also be formulated indirectly as a control task in which control actions (e.g., changes in heading and velocity) are determined at each timestep and the trajectory can then be predicted by tracing

the motion of the vehicle based on these actions. In this case, we will be dealing with a learning from demonstration (LfD) problem [25], in which we are interested in learning, from human driving data, what actions should be taken to properly control a vehicle. Given that a solution to the control problem can be easily transformed into a solution to the prediction problem, in our review of the literature we will consider not just the problem of trajectory prediction but also the associated control problem.

In this work, we develop a new solution based on a hybrid approach combining elements from the prediction formulation and the control formulation. We adopted a two-step approach to solving the problem. In the first step, based on the prediction formulation, we generated an initial prediction and generated candidates for the final prediction based on this initial prediction. In the second step, based on the control formulation, we ranked these candidates and chose one as the final prediction.

The first step consisted of representing vehicle trajectories as B-spline curves and training a neural network model to predict the coefficients of these B-spline curves. Similar approaches to trajectory representation have been used before, such as representing trajectories using Chebyshev polynomials [11], but to the best of our knowledge this is the first work to use B-spline curves for this purpose. These predicted B-spline curve coefficients represented the initial prediction, and candidate trajectories were generated from this initial prediction by random perturbations to the coefficients of the B-spline curves.

In the second step, an Inverse Reinforcement Learning (IRL) [26] model was used to rank the candidate trajectories that were generated in the first step. IRL is a technique for solving control problems by learning from demonstration and has previously been used to solve the trajectory prediction problem for highways [27][28], but to the best of our knowledge this is the first work to investigate its application to the problem at intersections. Trajectory prediction at intersections involves challenges not encountered in highways such as the presence of various conflict types, multiple types of road users (vehicles, pedestrians, and bicycles), and more complicated traffic control devices. This is also the first work to use MaxEnt IRL to select from a set of candidate trajectories. The work in [29] also used an IRL-like approach to rank candidate trajectories, but used an ad hoc formulation. Here we used IRL to develop methods that can address some of these complexities. The IRL model was trained using the B-spline–smoothed trajectories and the context of the vehicle at the intersection (i.e., the other vehicles present at the intersection). The second step allows us to predict trajectories that are more human-like and also to take interactions between the vehicles at the intersection into account. For the training and evaluation of our method, we used the Lankershim Boulevard dataset from the Next Generation Simulation (NGSIM) dataset collection. Using our method, we were able to achieve an approximately 20% improvement on prediction accuracy (as measured by the root mean square error [RMSE] of the predicted vs. actual trajectory) over a baseline method based on a Kalman Filter.

Related Work

Trajectory prediction methods can be classified into three broad categories [3]: physics based [17][30][31][32][33][12][15][21], maneuver based [5][7][11][23][18][19], and interaction aware [34][35][36][37]. Physics-based models, as the name suggests, deal with the physics of vehicle motion and assume that vehicles' trajectories are determined solely by physical forces, disregarding driver decisions that affect steering and acceleration. Consequently, these models fail to accurately predict vehicle motion beyond a short horizon. Maneuver-based models take driver actions into account but only in a vacuum; they consider these decisions to be determined solely by the position and the preceding trajectory of the

vehicle of interest, ignoring the influence other road users have on these actions, which leads to less reliable projections of future motion. Interaction-aware models predict trajectory by taking the presence of other road users into account. Here we have limited our focus to maneuver-based and interaction-aware models with a section dedicated to models using IRL. What follows is a summary of the literature on these models; comprehensive reviews of the subject can be found in [3] and [38]. We start with a general overview of previous work and then move on to provide further detail first on works that have applied IRL to the problem of trajectory prediction and then to works that involve the application of trajectory prediction to intersection safety.

Maneuver-based models: In [5], the authors used a radial basis function (RBF) network and a particle filter framework to predict future trajectories. An RBF network was trained to distinguish between "turn-left" and "straight-on" trajectories in the training dataset, and the outputs of this network were used as a priori probabilities for the trajectory hypotheses in the particle filter framework. A method based on an extension to hidden Markov models (HMMs) called growing hidden Markov models was proposed in [7] aiming to incrementally learn the structure and parameters for a trajectory prediction model. In [11], the authors proposed a probabilistic trajectory prediction method based on Gaussian mixture models and their Bayesian counterpart, variational Gaussian mixture models. Gaussian processes and a particle filter framework were used in [22] to perform maneuver recognition and multimodal trajectory prediction. An approach integrating a physics-based model and a maneuver-based model was proposed in [39]. The physics-based and maneuver-based models were respectively based on the constant turn rate and acceleration model with unscented Kalman filters to account for uncertainty and dynamic Bayesian networks. In [14], the authors proposed a method for trajectory prediction at intersections that treated leading and following vehicles differently. For leading vehicles, trajectory prediction was performed by a k-nearest neighbors (k-NN) using historical trajectory data, while the intelligent driver model (IDM) was used for following vehicles. In both cases, the initial predicted trajectories were refined using an adaptive Kalman filter.

Interaction-aware models: In [34], the trajectory prediction framework proposed in [5] was used to predict the joint trajectory of two vehicles at intersections. This was done by penalizing those trajectories that lead to avoidable collisions (i.e., trajectories for which the time to collision is larger than the drivers' reaction times). Coupled HMMs [24] were used in [23] with the assumption of asymmetric interactions; i.e., other vehicles influence the vehicle of interest but not vice versa, to predict driver behavior. In [35], the IDM was used to infer the intent of drivers at intersections in the presence of a preceding vehicle. A probabilistic graphical model and recursive Bayesian filtering were used in [40] and [36] to perform interaction-aware driving behavior prediction. In [37], a dynamic Bayesian network (DBN) was used in conjunction with a factored state space that allows for a model with less computational complexity. DBNs were also used in [41] to jointly model what drivers intend to do and what they are expected to do in a traffic context. In [6], traffic contexts were rasterized into two-dimensional images and a deep convolutional neural network was then used to perform trajectory prediction. In [42], a generative adversarial network was used to model driver behavior in highways. A solution to a restricted version of the trajectory prediction problem, that of predicting the changes in velocity along a predetermined path, at unsignalized intersections was proposed in [43]. This work modeled the problem as a partially observable Markov decision process in which the intended paths of the other vehicles constitute the hidden variables. Partially observable Markov decision processes were also used in [44] for AV decision-making in scenarios, including roundabouts and T-junctions. In [45] deep neural networks and long short term memory (LSTM) networks were used to predict vehicle trajectories at intersections. A technique called social pooling was used with LSTM and deep

convolutional neural networks (CNNs) in [46] to address the interactions between vehicles in trajectory prediction in a highway setting. In [47], a specially designed "Influence network" is used in conjunction with a DBN to perform vehicle trajectory prediction at intersections. Another solution to the trajectory prediction problem based on DBNs was proposed in [16]. Our method differs from these works in its use of MaxEnt IRL for accounting for interactions in a two-step process. In particular, the second step can be added on top of any prediction method that might not be interaction-aware to make it so.

Trajectory Prediction Using IRL

Several studies have used IRL to model driving, mostly in the context of highways. In [27], IRL was used to learn driving in highways from human demonstrations in a simulated environment. The use of IRL was motivated by the desire to achieve more humanlike behavior and a better ability to handle new scenarios. Deep Q-networks were used to address the exploding state space issue encountered in using IRL in a setting with a large state space. In addition to using a simulated environment instead of real-world data, this study had several other limitations. These included using constant speed and having at most two cars in front of the vehicle. The authors in [28] had similar motivations in using IRL for the task of learning individual driving styles on highways. The driving behavior of a number of drivers was recorded as they drove a car fitted with a variety of sensors on a highway. Maximum entropy IRL was then used to train a model to make driving decisions in styles similar to each of the individual drivers. This work used a reward function that was a linear function of a number of manually defined features such as acceleration, deviation from lane center, and distance to other vehicles. These last two works considered the control problem that was mentioned earlier in the introduction section. In both studies, the use of IRL allowed for faithful replication of human driving behavior and an ability to generalize to new situations. In [48], a hierarchical learning framework was proposed in which IRL was used to predict interactive driving behavior on two levels with a case study of ramp merging. The different levels of decision-making in their framework consisted of discrete, high-level decisions (e.g., whether to merge after or before a given car in their case study) and low-level continuous actions (e.g., the acceleration and heading changes at each time step.) Similar to the previous study, the reward function in this work was formulated as a linear function of several manually defined features. A notable limitation of this work is that the high-level discrete decisions and their corresponding low-level continuous features need to be manually defined based on the particular scenario (e.g., ramp merging) at hand. In [29], a generative framework based on conditional variational autoencoders using recurrent neural networks was used to generate possible future trajectories. An IRL approach was used to rank and refine the trajectories generated by the generative framework. It is noteworthy that this work did not use any of the commonly employed IRL formulation, but rather integrated a reward function into a larger framework where the reward function parameters were optimized in tandem with the rest of the architecture and the optimization method was dependent upon the sample generating component of the framework. IRL was used in [49] to choose from a set of trajectories generated using a rule-based method in a highway environment. IRL was chosen as the approach for this study because it allowed for a hybrid method that did not require mappings from circumstances to vehicle control to be manually engineered and at the same time produced interpretable results. In [50], a trajectory prediction method based on an encoder-decoder approach using RNNs was proposed that used IRL as a regularizer for the training of the encoder-decoder network. The use of IRL as a regularizer was intended to help the model better utilize the scene context information. IRL was used to directly predict trajectories in a highway environment in [51]. A summary of the studies above is presented in the Appendix A.

Trajectory Prediction for Intersection Safety

In this subsection, we explore in more detail those studies that have considered the trajectory prediction problem from the viewpoint of the infrastructure and whose proposed solutions cover the problem at intersections.

Trajectory prediction has a number of applications for intersection safety. One such application is the detection of risky driving behaviors, such as dangerous turns [18], red light running [18][14][20], abrupt stops, aggressive passes, speeding passes, and aggressive following [14]. Trajectory prediction is also instrumental to the early prediction of turning movements, which is helpful in avoiding accidents [45]. Collision prediction, avoidance/mitigation [21][15][17][16], and risk assessment [19][12][13] also make use of trajectory prediction. Each of the studies reviewed used their solutions to the problem of trajectory prediction to tackle one or more of these applications. For each study, we note the application(s) that it used trajectory prediction to address and also summarize this information under the "Tested Applications" column in Appendix B, which presents an overview of these studies. Our work is the first to investigate the application of IRL to the problem of intersection safety. It differs from the works discussed here in its two-step design and the use of B-spline curves to represent trajectories. It has some similarity to [47] and [45] in using neural networks. This table contains, for each study, the features used for trajectory prediction (Predictors), the sensors used for collecting these features' data (Data Collection Sensors), the number of intersections where data was gathered for training (if applicable), the frequency at which the predictors were sampled (sampling frequency), the duration for which data had to be collected before starting to make predictions (monitoring period), how far into the future the predicted trajectories stretch (prediction horizon), what evaluation metric was used for measuring the performance of either the trajectory prediction method or the safety system as a whole (evaluation metric), what possible applications were mentioned (applications), interactions between which types of road users were considered (interaction type), and what movements leading to possible hazards were considered.

Most studies have focused on predicting and mitigating crashes. In [12], the authors proposed a method for collision risk estimation between vehicles based on real-time trajectory prediction. The method used for trajectory prediction in this work was a linear Kalman filter. GPS data was used for determining the position of vehicles, and risk estimation was performed using the time to collision (TTC) predicted from the predicted trajectories. Another work to use TTC from predicted trajectories for collision risk estimation was [15], which also used a Kalman filter for trajectory prediction and differential GPS as the position sensor. A system for threat assessment and decision-making was proposed in [17] that used an unscented Kalman filter for trajectory prediction. A probabilistic threat assessment method was also developed for threat assessment along with a decision-making protocol for whether or not an intervention is necessary. In [16], an accident prewarning system was developed with a trajectory prediction method based on a DBN and a risk assessment method based on the identification of risky driving behavior. They also presented a method for deciding the collision avoidance strategy, which is based on TTC and time-to-

avoidance matrices. An intersection safety system was developed in [13] that used video data to predict the trajectory of vehicles at intersections and detect dangerous situations involving both vehicles and pedestrians using TTC and post encroachment time (PET). For trajectory prediction, it was assumed that vehicles drive according to "average drive lines," which were predefined average trajectories for vehicles. In [19], a trajectory prediction method based on extended Kalman filters was developed and used to identify conflict areas between vehicles and other road users and calculate time to enter and time to leave for these road users and conflict areas. An object-oriented Bayesian network was then used to estimate collision probability. In [18], a maneuver prediction model was presented for use in an infrastructure-based intersection safety system. The proposed system used location, speed, and acceleration data transmitted by vehicles and also roadside sensors for maneuver prediction. The objective of the system was to provide warnings for red light violations and right and left turning hazards.

Other studies have focused on applications such as the identification of certain behaviors. In [14], the authors developed a trajectory prediction method for identifying risky behavior caused by the lengthy warning sequence at the end of the green phase at high-speed intersections. A notable feature of their method is that it divides the problem into two cases: the case where the vehicle has enough distance from its leading vehicle that it acts independently, and the case where the vehicle's movements are influenced by the behavior of the leading vehicle (i.e., time headway to the leading vehicle is less than 6 s). A trajectory prediction method was developed in [45] for predicting turning movements at intersections. Video data from three intersections was used to extract vehicle trajectories and to train neural network models for predicting vehicle trajectories. In the process of predicting the turning movement of the vehicles, after a vehicle's trajectory has been predicted, it was compared against "typical paths" in order to obtain the final turning prediction (left, right, or through). In [47], trajectory data transcribed from a video camera was used to train neural network models for trajectory prediction of both vehicles and pedestrians, which can be used for predicting high-level behavior. A method to predict red light running was proposed in [20] that used trajectory prediction to detect red light running ahead of time and dynamically extend the all-red phase of the intersection signals to mitigate accidents. A method for collision risk prediction and warning was proposed in [21] which estimated the minimal future distance between possibly conflicting vehicles using a physics-based trajectory prediction method.

Data Description

For this study, we used the Lankershim Boulevard dataset from the Next Generation Simulation (NGSIM) dataset collection. This dataset was selected because it is one of the largest datasets of vehicle trajectories at intersections in the United States. This dataset contains vehicle trajectories transcribed from video data providing complete coverage of three signalized intersections and covering approximately 500 meters in length. The dataset comprises a total of 30 minutes of data starting from 8:15 a.m. These 30 minutes of data cover a wide range of traffic conditions at the intersections, including the intersection being nearly empty and the intersections being heavily

populated by vehicles. The data is in a tabular format, with each row corresponding to the state of a specific vehicle at a specific time. The data is sampled at 10 Hz and has the vehicle's position, lane number, velocity, acceleration, and the intersection at which it is currently located among its columns. In addition to trajectory data, this dataset also contains street marking data.

Data Cleaning and Organization

The trajectory data in the NGSIM dataset is provided as a single tabular file (in csv format) which provides data on the location (in latitude and longitude based both on the California State Plane III coordinates and also locally relative to the center of the boulevard in feet), type (auto/truck/motorcycle), speed (in feet per second), and the size (length and width in feet) of each vehicle at each point in time. A new column was added to the data to indicate whether each row corresponds to a vehicle being in the area of influence of an intersection and, if so, which one. This new column was used to remove the data pertaining to the times when vehicles were outside an intersection's area of influence. A vehicle was considered to be within an intersection's area of influence if it was no more than 60 meters away from the closest edge of the intersection. The 60-meter threshold was chosen to correspond with the length of the longest monitoring period that we wanted to consider. Moreover, the rows belonging to each vehicle were grouped and sorted with respect to time in order to obtain the vehicle trajectories. We also calculated the heading (in radians) for each vehicle at each point in time and added it as a column. Finally, the trajectories were rotated and translated such that their point of entry into the intersection was at the origin of the plane and straight movement through the intersection corresponded to movement along the y-axis. Table 1 provides an overview of the statistics of the dataset:

Table 1. Dataset Statistics

Intersection	Total Rows	Total Trajectories	Right Turns	Left Turns	Through	# of Autos	# of Trucks	# of Motorcycles
2	574,398	2,210	157	616	1,437	2,144	62	4
3	193,028	1,973	24	82	1,867	1,915	54	4
4	218,049	1,980	214	619	1,147	1,917	59	4

Methodology

Our method is made up of the two steps as shown in Figure 1. In the first step, B-spline curves were fit to vehicle trajectories in order to represent each vehicle trajectory using the coefficients of the B-splines. A neural network was then trained to predict these coefficients. Candidate trajectories were then generated using random perturbations of these coefficients. In the second step, the B-spline smoothed trajectories of the vehicles were embedded into images containing the geometry of the intersection and the other vehicles present at the intersection. These images were then used to train an IRL model, which evaluated the candidate trajectories and chose the best

among them. In the following two subsections we provide an overview of B-spline curves and IRL.

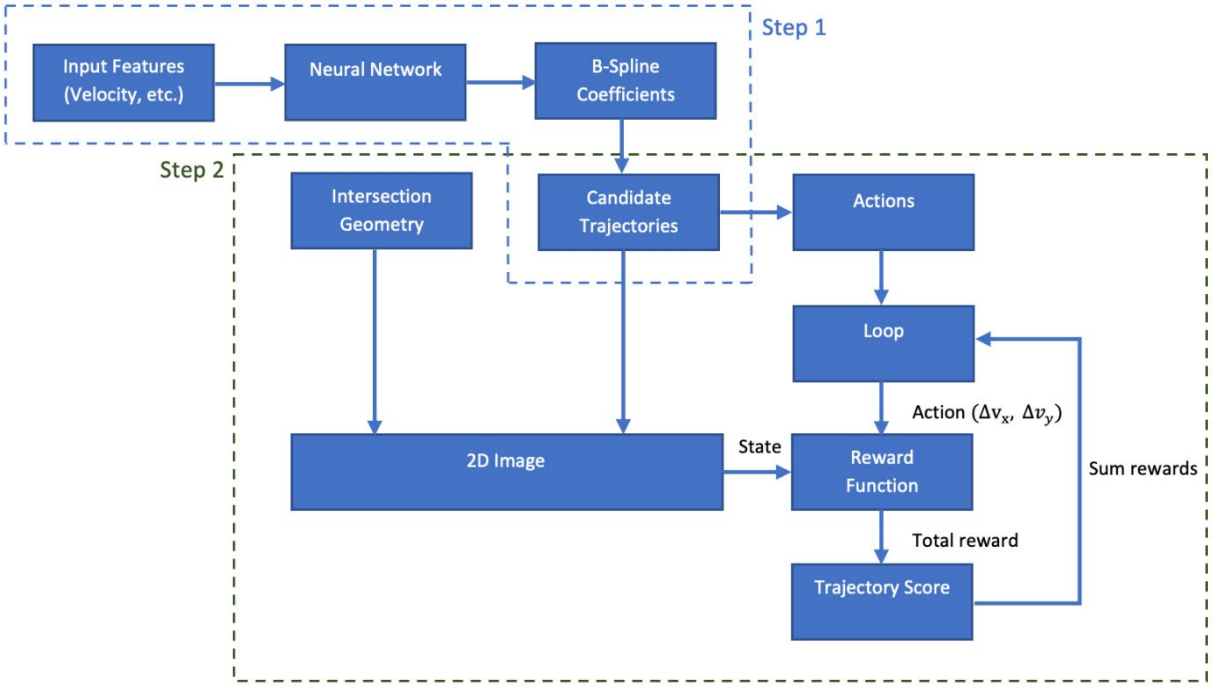


Figure 1. Flowchart. Overview of the method.

B-Splines

For a given *knot sequence*, $t_0 \leq t_1 \leq \cdots \leq t_{n+d+1}$, the B-spline basis functions are defined recursively as follows:

$$N_{i,0}(t) = \begin{cases} 1, & t_i \leq t \leq t_{i+1} \\ 0, & otherwise \end{cases}$$

$$N_{i,j}(t) = \frac{t - t_i}{t_{i+j} - t_i} N_{i,j-1}(t) + \frac{t_{i+j+1} - t}{t_{i+j+1} - t_{i+1}} N_{i+1,j-1}(t)$$

where $1 \leq j \leq d$ and $0 \leq i \leq n + d - j$. A one-dimensional B-spline curve is then defined in the following way:

$$x(t) = \sum_{i=0}^{n} c_i N_{i,d}(t)$$

For a given knot sequence and value of d, the c_is uniquely determine $f(t)$ and are referred to as the spline coefficients. In the training phase, these coefficients are estimated by finding the values of c_i that minimize the following objective function:

$$\sum_t \left(x(t) - \sum_{i=0}^{n} c_i N_{i,d}(t) \right)$$

In the test phase, these coefficients are predicted by a neural network, and the corresponding B-spline curve is the predicted trajectory. Note that we used univariate splines, which means that in order to represent each trajectory we need two spline curves, $x(t)$ and $y(t)$, corresponding to the x and y coordinates of the trajectory, respectively.

Inverse Reinforcement Learning

The Reinforcement Learning Problem

The Reinforcement Learning (RL) problem involves learning what actions to take in an interactive environment to maximize an objective function (called reward.) The main elements of reinforcement learning are the decision-making entity called the *agent*, the *environment* with which the agent interacts, and a *reward signal*, which is a numerical value provided by the environment to the agent at each timestep. The goal of the agent is to maximize the sum of the reward it receives over time.

Formally, an RL problem is defined by a Markov decision process (MDP.) An MDP is a tuple (S, A, p, γ, r) in which S is the set of all the states the environment can be in, A is the set of actions the agent can take, $p(s'|s,a)$ is the probability of the environment transitioning from state s to state s' if the agent takes action a, γ is the discount factor, and $r(s,a,s')$ is the expected reward given to the agent when the environment transitions from state s to state s' after the agent has taken action a. A policy $\pi(a|s)$ defines the probability of the agent taking action a when in state s. The expected *return* for a state s under a given policy π is the expected sum of the discounted reward values received by an agent starting from s and making a decision based on π and is denoted by $v_\pi(s)$. Thus, we have $v_\pi(s) = \sum_a \pi(a|s) \sum_{s'} p(s'|s,a)[r(s,a,s') + \gamma v_{\pi(s')}]$. In reinforcement learning, the objective is to find the optimal policy $\pi^\star$ which maximizes $v_\pi^\star(s)$ for every state s.

The Inverse Reinforcement Learning Problem

While the RL problem involves finding an optimal policy given a reward function, the IRL problem involves finding a reward function for which a given policy (represented by a set of samples from expert demonstrations) is optimal. Finding this reward function allows us to derive the policy and reproduce the behavior of the expert. The IRL problem as stated is ill posed because there are multiple reward functions for which a given policy is optimal; for instance, the set of reward functions that are constant everywhere is optimal for every policy. There have been several approaches to addressing this issue, one of which is the maximum entropy formulation [52]. In this formulation, it is assumed that the probability of a specific sequence of states and actions (denoted by τ) being observed is equal to $p(\tau) = \frac{1}{Z}\exp(r_\theta(\tau))$ in which $r_\theta(\tau) = \sum_{s,a\in\tau} r_\theta(s,a)$, where r_θ is the reward function parametrized by θ. This formulation posits that the expert acts

probabilistically and is most likely to traverse the optimal sequence of actions and states, with suboptimal sequences being exponentially less probable as their associated reward decreases. The central problem in this formulation is calculating or estimating the value of Z (often called the partition function). Several approaches have been proposed for solving this problem. In Guided Cost Learning [53] (GCL), the algorithm we use, this is achieved by importance sampling from the set of all possible sequences of states and actions. This importance sampling involves generating samples not present in the dataset. This is explored in more detail in the "Experiments" section of this report. The reason for choosing GCL here is that it enables tractably working with high-dimensional and continuous state spaces and actions while allowing for a nonlinear function approximator (here a neural network) to be used for approximating the reward function.

In our method, we used GCL with a convolutional neural network as the approximator for the reward function to recover the reward function of the human drivers and then used the recovered reward function to rank the candidate trajectories generated in the first step of the method. To this end, we first had to convert each candidate trajectory to a sequence of states and actions. The state at time t was specified by creating a two-dimensional image of the intersection containing the intersection geometry and the trajectories of all the vehicles at the intersection up to time t. The action at time t was a two-dimensional value specifying the change in velocity of the vehicle in the x and y directions at time t. If we denote the recovered reward function with $r(s_t, a_t)$ in which s_t denotes the state at time t and $a_t = \left(\Delta v_x, \Delta v_y\right)_t$ is the ordered pair representing the action at time t, the score, denoted by u, assigned to a trajectory $\tau = \langle(\mathrm{s}_1, \mathrm{a}_1), \ldots, (\mathrm{s}_\mathrm{n}, \mathrm{a}_\mathrm{n})\rangle$ is calculated using the following:

$$u = \sum_{t=1}^{n} r(s_t, a_t)$$

The value of u was calculated for every candidate trajectory and the candidate trajectory with the highest value was chosen as the final predicted trajectory.

Evaluation Metrics

A variety of different metrics have been used to evaluate the performance of trajectory prediction methods. One broad categorization of these metrics is by whether they directly measure the performance of the trajectory prediction algorithm or the application for which the method has been developed (e.g., collision avoidance). Given that we are not testing for any specific applications, we will only consider metrics of the former kind.

The most commonly used metric in evaluating trajectory prediction algorithms is the RMSE [14][37][47], which is calculated using the following formula:

$$\text{RMSE} = \frac{1}{\text{n}}\sqrt{\sum_{\text{i=1}}^{\text{n}}(\text{a}_\text{i} - \text{b}_\text{i})^2}$$

in which a_i and b_i are respectively the ith point along the original and the predicted trajectory. A related metric that has been used in at least one work for the purpose of evaluating a trajectory prediction algorithm is the mean absolute error (MAE) [39], defined as follows:

$$\text{MAE} = \frac{1}{\text{n}}\sum_{\text{i=1}}^{\text{n}}|\text{a}_\text{i} - \text{b}_\text{i}|$$

RMSE and MAE are the two most commonly used measures of performance for solutions to regression problems. It is therefore no surprise that they are also the performance measures of choice for the task of trajectory prediction. The reason why RMSE is used much more often and why we have opted to use it here is that RMSE penalizes large errors more than MAE, which is desirable in many regression problems (including ours).

Another method of measuring the accuracy of a trajectory prediction system, given in [54] and used in [6], divides the trajectory prediction error into two components: the parallel component called "along track error" and the perpendicular component called "cross track error." This is done by segmenting the ground truth and baseline trajectories and then computing the following:

$$\text{Along Track Error} = \sum_{\text{i=1}}^{\text{n}}(\text{x}_\text{i} - \text{x}_\text{i}^{'})\sin\theta_\text{i} + (\text{y}_\text{i} - \text{y}_\text{i}^{'})\cos\theta_\text{i}$$

$$\text{Cross Track Error} = \sum_{\text{i=1}}^{\text{n}}(\text{x}_\text{i} - \text{x}_\text{i}^{'})\cos\theta_\text{i} - (\text{y}_\text{i} - \text{y}_\text{i}^{'})\sin\theta_\text{i}$$

in which x_i, y_i, $x_i^{'}$, $y_i^{'}$ are the coordinates for the ground truth and predicted points, respectively, and θ_i is the predicted angle.

Another less commonly used metric is the longest common subsequence (LCS) [5]. $LCS(A, B)$ is the maximum number of points between trajectories A and B that are "close enough" and in the correct order. The LCS distance is then defined as $dist_{LCS}(A, B) = 1 - \frac{LCS(A,B)}{\min(N_A, N_B)}$. This metric has the advantage of being resilient to outliers but is mostly useful in the context of comparing trajectories for similarity as opposed to measuring the accuracy of a prediction model. Another closely related metric is the Quaternion Based Rotationally Invariant LCS (QRLCS), which adds rotational invariance to the translational invariance of the LCS.

Experiments and Results

In our experiments, we used the Lankershim Boulevard data from the NGSIM dataset. We extracted vehicle trajectories from this data and fit B-spline curves to the extracted trajectories. Of the resulting data, 10% was set aside as test data (distributed uniformly over the three different movement types.) We then trained a neural network to predict the coefficients of the B-spline

curves corresponding to the trajectories using 10-fold cross-validation on the rest of the data. The neural network had the following input features: the x and y distance from the center of the approach the vehicle entered the intersection from to the centers of the three road segments the vehicle can exit the intersection, the distance of the vehicle from the center of the approach, velocity before entering the intersection, vehicle acceleration before entering the intersection, vehicle heading before entering the intersection, average vehicle velocity over the monitoring period (2 seconds in the final model), average vehicle acceleration over the monitoring period, and the turning movements allowed for the lane the vehicle was on. We then generated candidate trajectories by randomly perturbing the predicted coefficients. An IRL model was trained in the following manner: the B-spline smoothed trajectories of the vehicles were embedded into images containing the geometry of the intersection, as well as the trajectories of the other vehicles present at the intersection (at test time, the trajectories predicted in the first step were used.) For the reward function approximator, we used a pretrained convolutional neural network, namely MobileNetV2, with the final softmax layer removed. As noted in the "Methods" section, training an IRL model using the GCL algorithm involved sample generation. This was done by changing the trajectory of the ego vehicle with respect to the sampled actions while maintaining the original trajectory of other vehicles. The trained IRL model gave us a recovered reward function, which was subsequently used to score the candidate trajectories generated in the first step of the algorithm. The candidate trajectory scoring the highest was the final prediction of the model.

The results of our experiments are summarized in Table 2. We see that the first step of our method without ranking by the IRL module already outperforms the baseline model. The addition of the IRL module further improves the performance of the model. Together with the IRL module, we achieve a 1-m (~20%) improvement in RMSE, which can be the difference between recognizing a potential crash and missing it.

Table 2. Summary of Results

Method	Avg. RMSE (m)
Baseline (Kalman Filter)	5.1
Neural Network	4.6
Neural Network + IRL Ranking	4.1

In order to get a qualitative assessment of the performance of the model, we can consider the trajectories in Figure 2. Here, we have the ground truth trajectory of a left turn in blue (lowest curve) with the prediction of the first step in red (uppermost curve), and finally the trajectory assigned the highest score by the IRL method in green (middle curve). We can observe that not only is the trajectory selected by the IRL module closer in location to the ground truth trajectory, but also it is more similar to it in shape and direction.

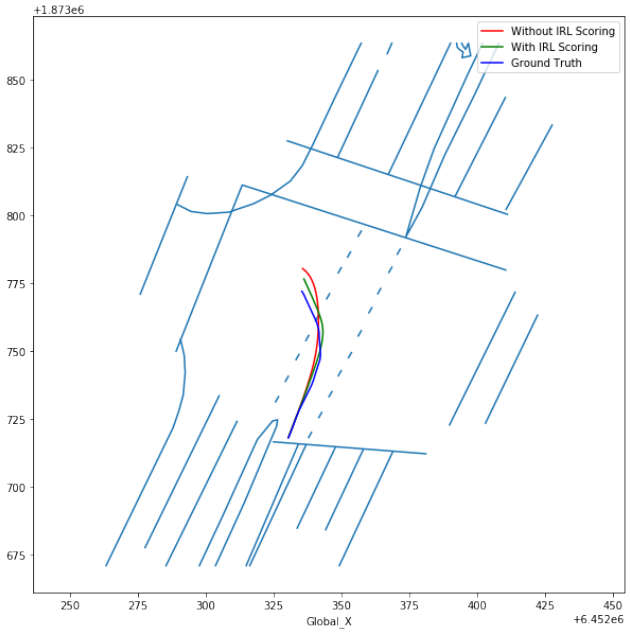


Figure 2. Illustration. Trajectory of a left turn and predictions of proposed method.

In order to better understand the performance of the model, as well as how the IRL module improves predictions, we consider the errors of the models broken down by movement type (i.e., whether the vehicle in question was going through the intersection, turning right, or turning left). The error values for different movement types are reported in Table 3.

Table 3. RMSE by Movement Type and Model

Movement Type	Avg. RMSE (m) Without IRL Scoring	Avg. RMSE (m) With IRL Scoring
Through	2.9	2.6
Right	14.7	12.8
Left	13.1	11.3

We can see from the results in Table 3 that the effect of the IRL scoring module is more pronounced in predicting turning movements. This can be explained by the fact that predicting the trajectory of turning movements is more difficult, and the IRL scoring module is therefore more likely to find a better trajectory among the generated candidates and return it as the top scoring trajectory.

We can also look at the error of the models as a function of the prediction horizon. These figures are reported in Table 4.

Table 4. RMSE by Prediction Horizon

Prediction Horizon (s)	Avg. RMSE Without IRL Scoring	Avg. RMSE With IRL Scoring
1	0.7	0.6
2	2.1	1.9
3	4.6	4.1

We again notice that as the task gets more difficult, the impact of the IRL scoring module increases. Here we see that the further the prediction horizon is, the more the IRL scoring module is able to improve predictions. This can be explained in the same way as the previous observation with through and turning movements: as the trajectories get more difficult to predict, the IRL scoring module is more likely to select a trajectory that is considerably more accurate from the set of candidate trajectories.

Conclusions and Recommendations

We have presented a two-step method for vehicle trajectory prediction at intersections. The first step of our method involved representing vehicle trajectories using B-spline curves and training a neural network to predict the coefficients of these B-spline curves. The second step of our method consisted of generating candidate trajectories based on the prediction of the neural network and using a reward function recovered by training an IRL model to the data to score these candidate trajectories and produce the final prediction. We have shown that a hybrid approach mixing elements from conventional supervised methods with elements from imitation learning can yield viable results for trajectory prediction. Our results indicate that IRL is an effective tool for addressing the shortcomings of conventional supervised methods with regard to the problem of trajectory prediction. We have furthermore demonstrated the suitability of B-spline curves for representing vehicle trajectories in such a way as to enable prediction. An avenue for future work lies in making context information available to the first step of the method. By making the model aware of interactions between vehicles from the first step, it should be possible to provide better input to the IRL scoring module and further improve the accuracy of the overall model. Another possible area for improvement would be modifications that allow the model to provide flexibility in terms of the starting point of the predictions before or after the vehicle reaches the intersection.

Additional Products

The Education and Workforce Development and Technology Transfer products created as part of this project are located on the project page of the Safe-D website. The final project dataset is located in the Safe-D Collection on the VTTI Dataverse.

Education and Workforce Development Products

The following Education and Workforce Development items resulted from project activities:

1. One master's student has been involved in all project activities. The student developed a state-of-the-art understanding of trajectory prediction models, utilized advanced machine learning methods to build trajectory prediction models, and improved a baseline model. The project contributed to the student's master's thesis. The student is also currently working on a journal paper. It is expected that the student will defend his MS thesis in the summer of 2021.

2. The project contributed to a new graduate course on intelligent transportation systems at San Diego State University (CIVE 696 Intelligent Transportation Systems). Materials are currently being developed from this project for this course.

3. Jazayeri, S., & Jahangiri, A. (2020). *Exploring inverse reinforcement learning architectures for vehicle trajectory prediction*. 2020 Student Research Symposium, San Diego State University.

4. The project team had a plan originally to present the project at the SDSU Explore Day event. It is likely that the event will be canceled due to COVID-19. If a virtual platform is provided for this event, the team will use that to present this project.

Technology Transfer Products

The following technology transfer products resulted from project activities:

- A journal paper was submitted to the *IEEE Transactions on Intelligent Transportation Systems*.

- Jazayeri, S., & Jahangiri, A. (2020). *Trajectory prediction at intersections using inverse reinforcement learning*. 11th International Conference on Applied Human Factors and Ergonomics (AHFE 2020), held virtually due to COVID-19, July 16–20.

- The project team is planning to present the work at the Caltrans Innovation Fair.

Data Products

- Link to Dataset – The data has been uploaded to the Safe-D Dataverse - DOI:10.15787/VTT1/AKKZ6V.

- Project Description – The goal of this project was to investigate the effectiveness of inverse reinforcement learning for predicting vehicle trajectories at intersections.
- Data Scope – Thirty minutes of vehicle trajectory data at three intersections from the Lankershim Boulevard Dataset (from the NGSIM dataset collection) was processed to obtain 6,100 vehicle trajectories in csv format, with each row corresponding to one vehicle trajectory.
- Data Specification – A detailed description of each variable in the dataset can be found in Appendix C.
- Citation Metadata:
    - Title of datasets: "SafeD-SDSU-01-01-Data.csv"
    - Author list with researcher ORCIDs
        - Mohammad Sadegh Jazayeri, 0000-0002-7814-8852
        - Arash Jahangiri, 0000-0002-8825-961X
        - Sahar Ghanipoor Machiani, 0000-0002-7314-2689
    - Contact information (email) for corresponding author: AJahangiri@sdsu.edu

Keywords: vehicle trajectory prediction, inverse reinforcement learning, neural networks, B-spline curves

References

[1] M. G. Mohamed and N. Saunier, "Motion Prediction Methods for Surrogate Safety Analysis," *Transp. Res. Rec.*, vol. 2386, no. 1, pp. 168–178, Jan. 2013, doi: 10.3141/2386-19.

[2] M. Wolterman, "Infrastructure-based collision warning using artificial intelligence," US7317406B2, Jan. 08, 2008.

[3] S. Lefèvre, D. Vasquez, and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," *ROBOMECH J.*, vol. 1, no. 1, p. 1, Dec. 2014, doi: 10.1186/s40648-014-0001-z.

[4] M. Schreier, V. Willert, and J. Adamy, "An Integrated Approach to Maneuver-Based Trajectory Prediction and Criticality Assessment in Arbitrary Road Environments," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 10, pp. 2751–2766, Oct. 2016, doi: 10.1109/TITS.2016.2522507.

[5] C. Hermes, C. Wohler, K. Schenk, and F. Kummert, "Long-term vehicle motion prediction," in *2009 IEEE Intelligent Vehicles Symposium*, Xi'an, China, Jun. 2009, pp. 652–657, doi: 10.1109/IVS.2009.5164354.

[6] N. Djuric *et al.*, "Motion prediction of traffic actors for autonomous driving using deep convolutional networks," *ArXiv Prepr. ArXiv180805819*, 2018.

[7] D. Vasquez, T. Fraichard, and C. Laugier, "Growing hidden markov models: An incremental tool for learning and predicting human and vehicle motion," *Int. J. Robot. Res.*, vol. 28, no. 11–12, pp. 1486–1506, 2009.

[8] D. Gettman and L. Head, "Surrogate safety measures from traffic simulation models," *Transp. Res. Rec.*, vol. 1840, no. 1, pp. 104–115, 2003.

[9] "Overview," *Transoft Solutions (ITS) Inc.* https://brisksynergies.com/overview/ (accessed Apr. 26, 2020).

[10] "Multimodal Detection," *Miovision*. https://miovision.com/solutions/multimodal-detection/ (accessed Apr. 26, 2020).

[11] J. Wiest, M. Höffken, U. Kreßel, and K. Dietmayer, "Probabilistic trajectory prediction with Gaussian mixture models," in *2012 IEEE Intelligent Vehicles Symposium*, 2012, pp. 141–146.

[12] S. Ammoun and F. Nashashibi, "Real time trajectory prediction for collision risk estimation between vehicles," in *2009 IEEE 5th International Conference on Intelligent Computer Communication and Processing*, 2009, pp. 417–422.

[13] P. Pyykönen, M. Molinier, and G. A. Klunder, "Traffic monitoring and modeling for intersection safety," in *Proceedings of the 2010 IEEE 6th International Conference on Intelligent Computer Communication and Processing*, 2010, pp. 401–408.

[14] C. Tan, N. Zhou, F. Wang, K. Tang, and Y. Ji, "Real-Time Prediction of Vehicle Trajectories for Proactively Identifying Risky Driving Behaviors at High-Speed Intersections," *Transp. Res. Rec. J. Transp. Res. Board*, vol. 2672, no. 38, pp. 233–244, Dec. 2018, doi: 10.1177/0361198118797211.

[15] Y. Wang, E. Wenjuan, D. Tian, G. Lu, G. Yu, and Y. Wang, "Vehicle collision warning system and collision detection algorithm based on vehicle infrastructure integration," 2011.

[16] Y. Fu, C. Li, T. H. Luan, Y. Zhang, and G. Mao, "Infrastructure-cooperative algorithm for effective intersection collision avoidance," *Transp. Res. Part C Emerg. Technol.*, vol. 89, pp. 188–204, 2018.
[17] G. R. de Campos, A. H. Runarsson, F. Granum, P. Falcone, and K. Alenljung, "Collision avoidance at intersections: A probabilistic threat-assessment and decision-making system for safety interventions," in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2014, pp. 649–654.
[18] T. Schendzielorz, P. Mathias, and F. Busch, "Infrastructure-based vehicle maneuver estimation at urban intersections," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, 2013, pp. 1442–1447.
[19] G. Weidl, G. Breuel, and V. Singhal, "Collision risk prediction and warning at road intersections using an object oriented Bayesian network," in *Proceedings of the 5th International Conference on Automotive User Interfaces and Interactive Vehicular Applications - AutomotiveUI '13*, Eindhoven, Netherlands, 2013, pp. 270–277, doi: 10.1145/2516540.2516577.
[20] L. Wang, L. Zhang, W.-B. Zhang, and K. Zhou, "Red light running prediction for dynamic all-red extension at signalized intersection," in *2009 12th International IEEE Conference on Intelligent Transportation Systems*, St. Louis, Oct. 2009, pp. 1–5, doi: 10.1109/ITSC.2009.5309545.
[21] P. Wang and C.-Y. Chan, "Vehicle collision prediction at intersections based on comparison of minimal distance between vehicles and dynamic thresholds," *IET Intell. Transp. Syst.*, vol. 11, no. 10, pp. 676–684, Dec. 2017, doi: 10.1049/iet-its.2017.0065.
[22] Q. Tran and J. Firl, "Online maneuver recognition and multimodal trajectory prediction for intersection assistance using non-parametric regression," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*, 2014, pp. 918–923.
[23] N. Oliver and A. P. Pentland, "Graphical models for driver behavior recognition in a SmartCar," in *Proceedings of the IEEE Intelligent Vehicles Symposium 2000 (Cat. No.00TH8511)*, Dearborn, MI, USA, 2000, pp. 7–12, doi: 10.1109/IVS.2000.898310.
[24] M. Brand, N. Oliver, and A. Pentland, "Coupled hidden Markov models for complex action recognition," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Juan, Puerto Rico, 1997, pp. 994–999, doi: 10.1109/CVPR.1997.609450.
[25] S. Schaal, "Learning from Demonstration," in *Advances in Neural Information Processing Systems 9*, M. C. Mozer, M. I. Jordan, and T. Petsche, Eds. MIT Press, 1997, pp. 1040–1046.
[26] S. J. Russell, "Learning agents for uncertain environments," in *COLT*, 1998, vol. 98, pp. 101–103.
[27] S. Sharifzadeh, I. Chiotellis, R. Triebel, and D. Cremers, "Learning to drive using inverse reinforcement learning and deep q-networks," *ArXiv Prepr. ArXiv161203653*, 2016.
[28] M. Kuderer, S. Gulati, and W. Burgard, "Learning driving styles for autonomous vehicles from demonstration," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 2641–2646.
[29] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. S. Torr, and M. Chandraker, "DESIRE: Distant Future Prediction in Dynamic Scenes with Interacting Agents," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, Jul. 2017, pp. 2165–2174, doi: 10.1109/CVPR.2017.233.

[30] Chiu-Feng Lin, A. G. Ulsoy, and D. J. LeBlanc, "Vehicle dynamics and external disturbance estimation for vehicle path prediction," *IEEE Trans. Control Syst. Technol.*, vol. 8, no. 3, pp. 508–518, May 2000, doi: 10.1109/87.845881.
[31] A. Barth and U. Franke, "Where will the oncoming vehicle be the next second?," in *2008 IEEE Intelligent Vehicles Symposium*, 2008, pp. 1068–1073.
[32] H. Veeraraghavan, N. Papanikolopoulos, and P. Schrater, "Deterministic sampling-based switching Kalman filtering for vehicle tracking," in *2006 IEEE Intelligent Transportation Systems Conference*, 2006, pp. 1340–1345.
[33] Q. B. Shao, H. Guan, and X. Jia, "Vehicle Trajectory Prediction based on Road Recognition," in *Applied Mechanics and Materials*, 2014, vol. 599, pp. 760–766.
[34] E. Käfer, C. Hermes, C. Wöhler, H. Ritter, and F. Kummert, "Recognition of situation classes at road intersections," 2010.
[35] M. Liebner, M. Baumann, F. Klanner, and C. Stiller, "Driver intent inference at urban intersections using the intelligent driver model," in *2012 IEEE Intelligent Vehicles Symposium*, Alcal de Henares , Madrid, Spain, Jun. 2012, pp. 1162–1167, doi: 10.1109/IVS.2012.6232131.
[36] G. Agamennoni, J. I. Nieto, and E. M. Nebot, "Estimation of Multivehicle Dynamics by Considering Contextual Information," *IEEE Trans. Robot.*, vol. 28, no. 4, pp. 855–870, Aug. 2012, doi: 10.1109/TRO.2012.2195829.
[37] T. Gindele, S. Brechtel, and R. Dillmann, "A probabilistic model for estimating driver behaviors and vehicle trajectories in traffic environments," in *13th International IEEE Conference on Intelligent Transportation Systems*, Funchal, Madeira Island, Portugal, Sep. 2010, pp. 1625–1631, doi: 10.1109/ITSC.2010.5625262.
[38] J. Wiest, *Statistical long-term motion prediction*. $ nUniversität Ulm, Institut für Mess-, Regel-und Mikrotechnik, 2016.
[39] G. Xie, H. Gao, L. Qian, B. Huang, K. Li, and J. Wang, "Vehicle Trajectory Prediction by Integrating Physics- and Maneuver-Based Approaches Using Interactive Multiple Models," *IEEE Trans. Ind. Electron.*, vol. 65, no. 7, pp. 5999–6008, Jul. 2018, doi: 10.1109/TIE.2017.2782236.
[40] G. Agamennoni, J. I. Nieto, and E. M. Nebot, "A bayesian approach for driving behavior inference," in *2011 IEEE Intelligent Vehicles Symposium (IV)*, Baden-Baden, Germany, Jun. 2011, pp. 595–600, doi: 10.1109/IVS.2011.5940407.
[41] S. Lefevre, C. Laugier, and J. Ibanez-Guzman, "Risk assessment at road intersections: Comparing intention and expectation," in *2012 IEEE Intelligent Vehicles Symposium*, Alcal de Henares , Madrid, Spain, Jun. 2012, pp. 165–171, doi: 10.1109/IVS.2012.6232198.
[42] A. Kuefler, J. Morton, T. Wheeler, and M. Kochenderfer, "Imitating driver behavior with generative adversarial networks," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 204–211.
[43] C. Hubmann, J. Schulz, M. Becker, D. Althoff, and C. Stiller, "Automated driving in uncertain environments: Planning with interaction and uncertain maneuver prediction," *IEEE Trans. Intell. Veh.*, vol. 3, no. 1, pp. 5–17, 2018.
[44] W. Liu, S.-W. Kim, S. Pendleton, and M. H. Ang, "Situation-aware decision making for autonomous driving on urban road using online POMDP," in *2015 IEEE Intelligent Vehicles Symposium (IV)*, Seoul, South Korea, Jun. 2015, pp. 1126–1133, doi: 10.1109/IVS.2015.7225835.

[45] M. S. Shirazi and B. T. Morris, “Trajectory prediction of vehicles turning at intersections using deep neural networks,” *Mach. Vis. Appl.*, vol. 30, no. 6, pp. 1097–1109, 2019.
[46] N. Deo and M. M. Trivedi, “Convolutional social pooling for vehicle trajectory prediction,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 1468–1476.
[47] A. Sarkar, K. Czarnecki, M. Angus, C. Li, and S. Waslander, “Trajectory prediction of traffic agents at urban intersections through learned interactions,” in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 2017, pp. 1–8.
[48] L. Sun, W. Zhan, and M. Tomizuka, “Probabilistic prediction of interactive driving behavior via hierarchical inverse reinforcement learning,” in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 2111–2117.
[49] R. Sun, S. Hu, H. Zhao, M. Moze, F. Aioun, and F. Guillemard, “Human-like Highway Trajectory Modeling based on Inverse Reinforcement Learning,” in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 1482–1489.
[50] D. Choi, K. Min, and J. Choi, “Regularizing Neural Networks for Future Trajectory Prediction via Inverse Reinforcement Learning Framework,” *ArXiv190704525 Cs*, Dec. 2019, Accessed: Jun. 05, 2020. [Online]. Available: http://arxiv.org/abs/1907.04525.
[51] B. Hjaltason, *Predicting vehicle trajectories with inverse reinforcement learning*. 2019.
[52] B. D. Ziebart, A. Maas, J. A. Bagnell, and A. K. Dey, “Maximum Entropy Inverse Reinforcement Learning,” p. 6.
[53] C. Finn, S. Levine, and P. Abbeel, “Guided Cost Learning: Deep Inverse Optimal Control via Policy Optimization,” p. 10.
[54] C. Gong and D. McNally, “A methodology for automated trajectory prediction analysis,” in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2004, p. 4788.

Appendix/Appendices

Appendix A: An Overview of the Literature on Trajectory Prediction

Study	Approach	Intersection/Road Segment	AV/Infrastructure	Methods	Predictors	Accuracy
[17]	Physics Based	Intersections	AV	UKF	Previous trajectory	
[31]	Physics Based	Road Segment	AV	EKF	Previous trajectory	
[15]	Physics Based	Intersection	Infrastructure	KF	Previous trajectory	
[18]	Physics Based	Intersection	Infrastructure	KF	Previous trajectory	
[21]	Physics Based	Intersection	Infrastructure	KF	Previous trajectory	
[19]	Maneuver Based	Intersection	Infrastructure	EKF, OOBN	Previous trajectory	
[5]	Maneuver Based	Intersection	AV	RBF Network, Particle Filter	Previous Trajectory	RMSE 5 m @ 3 s
[7]	Maneuver Based	Both	Not Specified	GHMM	Previous Trajectory	
[22]	Maneuver Based	Intersection	AV	Gaussian Process, Particle Filter	Previous Trajectory	
[11]	Maneuver Based	Intersection	AV	Gaussian Mixture Model	Previous Trajectory	
[39]	Maneuver and Physics Based	Road Segment	AV	Dynamic Bayesian Network and UKF	Previous Trajectory	Average Error: 0.69
[14]	Maneuver Based	Intersection	Infrastructure	k-NN, Adaptive KF, IDM	Previous trajectory	RMSE: 5.02

Study	Approach	Intersection/Road Segment	AV/Infrastructure	Methods	Predictors	Accuracy
[34]	Interaction Aware	Intersection	AV	Particle Filter	Previous Trajectory and Surroundings	
[23]	Interaction Aware	Both	AV	CHMM	Previous Trajectory and Surroundings	
[35]	Interaction Aware	Intersection	AV	IDM	Previous Trajectory and Surroundings	Only maneuver prediction
[40]	Interaction Aware	Mine	AV	DBN	Previous Trajectory and Surroundings	
[37]	Interaction Aware	Road Segment	AV	DBN	Previous Trajectory and Surroundings	Average RMS: 0.21 m
[6]	Interaction Aware	Both	AV	CNN	Previous trajectory and Surroundings	
[36]	Interaction Aware	Both	AV	Probabilistic Graphical Model	Previous Trajectory and Surroundings	
[41]	Interaction Aware	Intersection	AV	Dynamic Bayesian Network	Previous Trajectory and Surroundings	
[42]	Interaction Aware	Highway	Infrastructure	GAIL	Previous Trajectory and Surroundings	
[43]	Interaction Aware	Intersection	AV	Partially Observable MDP	Previous Trajectory and Surroundings	
[44]	Interaction Aware	Roundabout, T-Junction, Urban Road	AV	Partially Observable MDP	Previous Trajectory	

Study	Approach	Intersection/Road Segment	AV/Infrastructure	Methods	Predictors	Accuracy
					and Surroundings	
[45]	Interaction Aware	Intersection	Infrastructure	Deep Neural Networks	Previous Trajectory and Surroundings	92% avg. accuracy
[46]	Interaction Aware	Highway	AV	LSTM	Previous Trajectory and Surroundings	
[47]	Interaction Aware	Intersection	AV	Neural Net and DBN	Previous Trajectory and Surroundings	RMSE: 3 m
[16]	Interaction Aware	Intersection	Infrastructure	DBN	Previous Trajectory and Surroundings	
[27]	Interaction Aware	Highway	AV	IRL and Deep Q-Nets	Surroundings	
[28]	Interaction Aware	Highway	AV	IRL	Surroundings	
[48]	Interaction Aware	Road Segment	AV	Hierarchical IRL	Surroundings	
[29]	Interaction Aware	Both	AV	Recurrent Neural Networks and IRL	Previous Trajectory and Surroundings	
[49]	Interaction Aware	Highway	AV	IRL	Surroundings	
[51]	Interaction Aware	Highway	AV	IRL	Previous Trajectory and Surroundings	

Appendix B: Summary of Studies on Trajectory Prediction for Intersection Safety

Study	Predictors (detail)	Data Collection Sensors	# of Intersections in Training Data	Sampling Frequency	Monitoring Period	Prediction Horizon	Evaluation Metric	Evaluation Metric Value	Applications	Interaction Type	Movement Type
[14]	Position, velocity, distance to preceding vehicle, speed difference from preceding vehicle	Video camera	3	8 Hz	1 s	12 s	RMSE of difference between predicted and actual trajectory	5.02 m	Detect red light running, abrupt stops, aggressive passes, speeding passes, and aggressive following	Vehicle, vehicle-vehicle	All
[45]	Vehicle position over a number of preceding frames	Video camera	2+1	15/5 Hz	1/3 of each trajectory	2 s	Turning prediction accuracy	92%	Early prediction of turning movements	Vehicle-vehicle, vehicle-pedestrian	All
[12]	Vehicle position, velocity, and acceleration	GPS	Not applicable	10 Hz	Up to the prediction point	10 s	No quantitative evaluation	NA	Collision detection and risk assessment	Vehicle-vehicle	All
[15]	Vehicle position and velocity	DGPS	Not applicable	Not specified	Not specified	Not specified	No quantitative evaluation	NA	Collision detection and warning	Vehicle-Vehicle	All
[47]	Vehicle position, velocity, and previous trajectory + surroundings	Video camera	1	33 Hz	Not specified	0-3 s	RMSE of difference between predicted and actual trajectory	3 m	This one is actually AV, but because it is at intersections and fits our theme it is included in this section	Vehicle-vehicle, vehicle-Pedestrian	All

Study	Predictors (detail)	Data Collection Sensors	# of Intersections in Training Data	Sampling Frequency	Monitoring Period	Prediction Horizon	Evaluation Metric	Evaluation Metric Value	Applications	Interaction Type	Movement Type
[17]	Vehicle position, speed, acceleration, and yaw	GPS + inertial sensors	Not applicable	Not specified	Not specified	Not specified	No quantitative evaluation	NA	Frontal collision prevention/mitigation	Vehicle-vehicle	Frontal collisions caused by any movement
[16]	Vehicle position, velocity, acceleration, distance traveled, turn signal, road condition	Simulation	1	Not specified	Not specified	Not specified	TPR, FPR, and FNR for collision prediction and collision avoidance success	TP: 97%, FPR: 1.25%, FNR: 0.95% Collision avoidance success: 96.65%-98.99% depending on road condition	Collision detection and prevention/mitigation by finding conflicts between drivers' intentions and expected behavior; red light running/dangerous stopping, etc.	Vehicle-vehicle	All movements
[13]	Vehicle position, velocity	Video camera	2	25 Hz	Prediction performed at every time step	Not specified	No quantitative evaluation	NA	Collision detection and prevention/mitigation	Vehicle-vehicle, vehicle-pedestrian	All movements
[18]	Vehicle position, velocity, acceleration	Roadside sensors, onboard GPS	1	Not specified	Not specified	Maximum of 10 s	Levels of accident mitigation		Detection and mitigation of red light running, dangerous right turns, dangerous left turns	Vehicle-vehicle, vehicle-cyclist, vehicle-pedestrian	Turns and red light running
[20]	Vehicle position, velocity, and acceleration	Video camera	1	Not specified	Not specified	Not specified	Simulated SOC curve	0.9 detection at 0.06 false alarm rate	Red light running prediction and mitigation through all-red extension	N/A	Red light running
[19]	Vehicle position, velocity, acceleration	Intersection mounted cameras and laser sensors + onboard sensors	Not applicable	Not specified	Not specified	2 s	No quantitative evaluation	NA	Collision risk prediction	Vehicle-vehicle, vehicle-pedestrian, vehicle-cyclist	All movements

Study	Predictors (detail)	Data Collection Sensors	# of Intersections in Training Data	Sampling Frequency	Monitoring Period	Prediction Horizon	Evaluation Metric	Evaluation Metric Value	Applications	Interaction Type	Movement Type
[21]	Vehicle position, velocity, acceleration	Not specified	Not applicable	10 Hz	Not specified	3 s	False positive + false negative	Fp = 2.4%, fn = 3.6%	Collision prediction, collision warning	Vehicle-vehicle	All movements

Appendix C: Data Specification

#	Variable	Description
1	dist_1	the distance (in meters) from the center of the approach the vehicle is entering the intersection from to the center of the approach to the vehicle's right
2	x_dist_1	the distance (in meters) along the x-axis from the center of the approach the vehicle is entering the intersection from to the center of the approach to the vehicle's right
3	y_dist_1	the distance (in meters) along the y-axis from the center of the approach the vehicle is entering the intersection from to the center of the approach to the vehicle's right
4	dist_2	the distance (in meters) from the center of the approach the vehicle is entering the intersection from to the center of the approach opposite the vehicle
5	x_dist_2	the distance (in meters) along the x-axis from the center of the approach the vehicle is entering the intersection from to the center of the approach opposite the vehicle
6	y_dist_2	the distance (in meters) along the y-axis from the center of the approach the vehicle is entering the intersection from to the center of the approach opposite the vehicle
7	dist_3	the distance (in meters) from the center of the approach the vehicle is entering the intersection from to the center of the approach to the vehicle's left
8	x_dist_3	the distance (in meters) along the x-axis from the center of the approach the vehicle is entering the intersection from to the center of the approach to the vehicle's left
9	y_dist_3	the distance (in meters) along the y-axis from the center of the approach the vehicle is entering the intersection from to the center of the approach to the vehicle's left
10	x	the distance (in meters) along the x-axis between the vehicle and the center of the approach
11	y	the distance (in meters) along the y-axis between the vehicle and the center of the approach
12	acc	vehicle acceleration (in m/s^2) before entering the intersection
13	acc_x	vehicle acceleration (in m/s^2) along the x-axis before entering the intersection
14	acc_y	vehicle acceleration (in m/s^2) along the y-axis before entering the intersection
15	vel	vehicle velocity (in m/s) before entering the intersection
16	vel_x	vehicle velocity (in m/s) along the x-axis before entering the intersection
17	vel_y	vehicle velocity (in m/s) along the y-axis before entering the intersection
18	acc_avg	average vehicle acceleration (in m/s^2 over 2 s) before entering the intersection
19	acc_x_avg	average vehicle acceleration (in m/s^2 over 2 s) along the x-axis before entering the intersection

#	Variable	Description
20	acc_y_avg	average vehicle acceleration (in m/s^2 over 2 s) along the y-axis before entering the intersection
21	vel_avg	average vehicle velocity (in m/s over 2 s) before entering the intersection
22	vel_x_avg	average vehicle velocity (in m/s over 2 s) along the x-axis before entering the intersection
23	vel_y_avg	average vehicle velocity (in m/s over 2 s) along the y-axis before entering the intersection
24	heading	vehicle heading (in radians) before entering the intersection
25	right_movement	variable specifying whether right turns are permitted (=1) or not (=0) for the lane the vehicle is on
26	through_movement	variable specifying whether through movements are permitted (=1) or not (=0) for the lane the vehicle is on
27	left_movement	variable specifying whether left turns are permitted (=1) or not (=0) for the lane the vehicle is on
28	x_coeff_{n}	the n^{th} B-spline curve coefficient for the vehicle's trajectory along the x-axis
29	y_coeff_{n}	the n^{th} B-spline curve coefficient for the vehicle's trajectory along the y-axis