

Real-Time Prediction of Intermediate-Horizon Automotive Collision Risk

Blake Wulfe
Stanford University
Stanford, CA
wulfebw@stanford.edu

Sunil Chintakindi
The Allstate Corporation
Menlo Park, CA
sunil.chintakindi@allstate.com

Sou-Cheng T. Choi
The Allstate Corporation
Northbrook, IL
soucheng.choi@allstate.com

Rory Hartong-Redden
The Allstate Corporation
Menlo Park, CA
rory.hartong-redden@allstate.com

Anuradha Kodali
The Allstate Corporation
Menlo Park, CA
akoda@allstate.com

Mykel J. Kochenderfer
Stanford University
Stanford, CA
mykel@stanford.edu

ABSTRACT

Advanced collision avoidance and driver hand-off systems can benefit from the ability to accurately predict, in real time, the probability a vehicle will be involved in a collision within an intermediate horizon of 10 to 20 seconds. The rarity of collisions in real-world data poses a significant challenge to developing this capability because, as we demonstrate empirically, intermediate-horizon risk prediction depends heavily on high-dimensional driver behavioral features. As a result, a large amount of data is required to fit an effective predictive model. In this paper, we assess whether simulated data can help alleviate this issue. Focusing on highway driving, we present a three-step approach for generating data and fitting a predictive model capable of real-time prediction. First, high-risk automotive scenes are generated using importance sampling on a learned Bayesian network scene model. Second, collision risk is estimated through Monte Carlo simulation. Third, a neural network domain adaptation model is trained on real and simulated data to address discrepancies between the two domains. Experiments indicate that simulated data can mitigate issues resulting from collision rarity, thereby improving risk prediction in real-world data.

KEYWORDS

Policy Evaluation; Monte Carlo Simulation; Bayesian Networks; Importance Sampling; Domain Adaptation; Transfer Learning

ACM Reference Format:

Blake Wulfe, Sunil Chintakindi, Sou-Cheng T. Choi, Rory Hartong-Redden, Anuradha Kodali, and Mykel J. Kochenderfer. 2018. Real-Time Prediction of Intermediate-Horizon Automotive Collision Risk. In *Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018)*, Stockholm, Sweden, July 10–15, 2018, IFAAMAS, 10 pages.

1 INTRODUCTION

The ability to accurately predict intermediate-horizon automotive risk from scene information is valuable for safety applications. For example, this capability can be used to inform the control-hand off problem in level-3 autonomous vehicles, or to allow for preemptive rerouting of autonomous vehicles away from high-risk situations.

There are three major challenges when predicting automotive risk from scene information. The first challenge is partial observability, which can arise due to occluded vehicles and sensor uncertainty along with unobserved driver information, such as degree of aggressiveness or maneuver intention. We assume full observability in this work for simplicity. The second challenge is that, as the prediction horizon increases, the set of possible vehicle trajectories grows exponentially. As a result, measures of risk associated with real-world data are generally imprecise, and estimating risk in simulation is computationally expensive. We restrict ourselves to an intermediate horizon in this paper.

The third challenge in intermediate-horizon risk prediction, and the focus of this paper, is a lack of data for fitting predictive models. This lack of data results from collision rarity and the high-dimensional nature of the problem. We demonstrate empirically that driver behavioral features have an increasingly significant impact as the prediction horizon increases. Learning a predictive model of risk that depends upon relatively high-dimensional behavioral features for many nearby vehicles requires a large amount of data, which in high-risk cases is unavailable. With collisions occurring approximately twice per million vehicle miles traveled in the U.S. [53], an economical and safe method for addressing collision rarity when fitting predictive models is desired.

A variety of methods have been considered for addressing the challenge of collision rarity in learning a predictive model. A common approach is to focus on predicting risk surrogates, such as hard braking or low time-to-collision events [10, 15]. While prediction of these quantities largely avoids issues resulting from rarity, it relies on the assumptions that these events correlate well with collisions and that they capture many collision modalities [18]. A second approach to addressing collision rarity is to augment existing data, for example through random noise or transforms [28], or more sophisticated oversampling [8]. These approaches typically assume smooth relationships between covariate and response variables, and it is not clear whether this holds in a risk prediction setting.

This paper aims to determine whether simulated data can help mitigate issues resulting from collision rarity. This approach uses prior knowledge about the automotive domain, for example that vehicles behave according to physical laws or follow certain driver models, to generate sufficiently realistic data to improve prediction performance through transfer learning [37]. A simulation-based approach to addressing collision rarity faces two primary challenges:

Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), M. Dastani, G. Sukthankar, E. André, S. Koenig (eds.), July 10–15, 2018, Stockholm, Sweden. © 2018 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

(i) efficient generation of high-risk simulated data, and (ii) reducing or compensating for inevitable differences between simulated and real-world data, thereby enabling effective transfer.

This paper presents a method that addresses these two challenges. The first challenge is addressed using importance sampling, which allows us to focus computational effort on higher-risk scenes. Determining proposal distributions from which to sample events can be challenging, so we propose an approach for automatically learning high-risk automotive scene distributions using the cross entropy method [11, 43]. We address the second challenge of effective transfer learning through a two-pronged approach. We first allow for simulated data to closely resemble real data when possible. We accomplish this by learning scene models from data, and by employing a risk estimation framework that imposes few assumptions on the dynamics or driver behavior models. Second, since simulated data will necessarily differ from real data, we use recent adversarial domain adaptation methods [14].

We validate our approach to intermediate-horizon risk prediction in two experiments. First, in a fully simulated setting, we demonstrate the effectiveness of the system in mitigating issues resulting from collision rarity. Second, we demonstrate that simulated data can be effectively transferred to improve a predictive model applied to real-world data.

2 RELATED WORK

Automotive risk models often address problems arising from collision rarity by predicting collision surrogates such as initiating conditions and evasive maneuvers [10, 15, 49]. These surrogate events are assumed to correlate with collisions [17, 20]. Lord et al. review approaches to statistical analysis of crash frequency data [33], which use *static* scene features. A variety of approaches have been considered in this context, including negative-binomial regression [39], support vector machines [32], and neural networks [7].

In contrast, real-time systems use *dynamic* features of a scene to estimate risk [31]. These approaches leverage some form of motion prediction, which range in complexity from physics-based models [3, 21] to those accounting for driver maneuver intention [30, 45]. Monte Carlo simulation methods permit complex dynamics and driver models by not making assumptions about these factors, and have been widely considered in automotive risk estimation [2, 6, 12].

Our framework uses a Monte Carlo approach, though it differs from existing methods in two important ways. First, previous research has emphasized short-term prediction [31]. Second, whereas existing approaches propose to estimate risk in real-time by executing simulations on-vehicle, our approach amortizes the cost of running this optimization by learning a predictive model that generalizes across scenes.

Driver behavior models determine the actions taken by vehicles in automotive simulations. Heuristic driver models, such as the intelligent driver model (IDM) [51] and MOBIL [23], can be used to generate collision-free trajectories. A variety of collision-inclusive, heuristic models exist, for example, the Errorable Driver Model [58] and Less-Than-Perfect driver [57]. While these models exhibit collisions, truly human-like behavior and failure modes are better captured by fully parametric models that are learned from data

[29, 35]. We focus on the former heuristic models in this work, and leave more sophisticated models for future consideration.

Automotive scenes can be generated through heuristic means, or by learning a generative scene model from data. Wheeler et al. consider Bayesian networks [56] and factor graphs [55] for learning scene models. In this paper, we adapt the Bayesian network approach proposed by Wheeler et al. [56].

Due to collision rarity, many scene samples may be required to produce a collision, and many Monte Carlo simulations of those scenes may be needed to arrive at risk estimates with small relative error. Importance sampling provides a means of addressing these inefficiencies by oversampling dangerous scenes, and then weighing them based on their relative likelihood in an estimate or objective function [16]. Importance sampling has been applied for sampling lane-change scenarios, in particular with proposal distributions learned through the cross entropy method (CEM) [60]. CEM is a general optimization method common in rare event simulation [11, 43] that we also employ, but do so in learning full Bayesian network proposal distributions.

We use methods from the field of domain adaptation (DA) in order to improve transfer between simulated and real-world data. This problem has been considered in the context of reinforcement learning for robotic control [19, 44], as well as in supervised tasks [54]. Unsupervised DA approaches typically seek to identify both domain invariant and discriminative features [14], or to infer shared and private latent spaces of the two domains [5], and have been applied, for example, to object classification [47].

3 PROBLEM STATEMENT

This section first introduces Markov decision processes and formulates risk estimation as policy evaluation. We then discuss three traits of the problem that will later motivate our approach.

3.1 Background

We formulate risk estimation within the Markov decision process (MDP) framework [4, 25, 40] due to its natural interpretation as policy evaluation. A finite-horizon MDP consists of a set of states \mathcal{S} , actions \mathcal{A} , a probability distribution $P(s' | s, a)$ over the next state s' given the current state s and action a , a reward function $R(s)$, which we assume is deterministic and a function of only the current state, a horizon $H < \infty$, and an initial distribution over states ρ_1 .

A stochastic policy π_θ parameterized by $\theta \in \Theta$ defines a probability distribution over actions given the current state: $\pi_\theta(a | s)$. The return of a policy starting at time t is the sum of rewards $r_t + r_{t+1} + \dots + r_H$ it receives when interacting with an environment. The expected return of using policy π_θ starting from state s_t is referred to as the value of the policy, which can be expressed using the Bellman equation:

$$\begin{aligned} V^{\pi_\theta}(s_t) &= \mathbb{E}_{\pi_\theta} \left[\sum_{k=t}^H R(s_k) \mid S_t = s_t \right] \\ &= R(s_t) + \sum_a \pi_\theta(a | s_t) \sum_{s'} P(s' | s_t, a) V^{\pi_\theta}(s'). \end{aligned} \quad (1)$$

Policy evaluation is the task of computing the value of a policy for all states, and can be performed through a dynamic programming

procedure that iterates equation (1) as an update. This approach can be intractable in large state or action spaces, in which case Monte Carlo policy evaluation may be employed instead, wherein sample returns are averaged to approximate the value of a policy.

3.2 Problem Formulation

We formulate automotive risk estimation as policy evaluation. A particular vehicle of concern, the “ego vehicle”, operates according to a policy π_θ parameterized by θ . We refer to these parameters as the “behavioral features” of the vehicle because they are the parameters of the driver behavior models (e.g., IDM and MOBIL) that control vehicle action selection.

We assume full observability, and thus the state contains the physical attributes as well as the behavioral features of the ego vehicle and neighboring vehicles. Because we assume that behavioral traits are fully observed, this definition of the state corresponds most closely with the notion of a *scene* within the automotive safety literature [52], and we use “state” and “scene” interchangeably.

We are interested in automotive risk, which can be defined as the “likelihood and severity of the damage that a vehicle of interest may suffer in the future” [31]. We focus only on the probability of collisions, or occasionally on surrogate measures of risk, and therefore we would like to design a reward function such that the value of a policy corresponds to the probability of that particular policy suffering a collision. We accomplish this by defining an indicator function $C(s_t)$ equal to 1 if s_t contains a collision involving the policy of interest, and 0 otherwise. We focus on collisions occurring after some initial timestep h , and define a collision indicator function that depends on this value:

$$Y_h(s_t) = \begin{cases} 1, & \text{if } C(s_t) = 1 \text{ and } t \geq h \\ 0, & \text{otherwise} \end{cases}. \quad (2)$$

States containing collisions of the ego vehicle are terminal states.

The value of a policy with respect to a state is the probability that policy enters into a collision starting in state s_t and acting according to π_θ until the horizon H :

$$V^{\pi_\theta}(s_t) = \mathbb{E}_{\pi_\theta} \left[\sum_{k=t}^H R(s_k) \mid S_t = s_t \right] \quad (3)$$

$$= \sum_{k=t}^H \mathbb{E}_{\pi_\theta} [Y_h(s_k) \mid S_t = s_t] \quad (4)$$

$$= \sum_{k=t}^H P(Y_h(s_k) = 1 \mid S_t = s_t, \pi_\theta). \quad (5)$$

Above, we use the properties of linearity of expectation and the expectation of an indicator random variable. Defining states containing collisions as terminal states makes the events $Y_h(s_i) = 1$ and $Y_h(s_j) = 1$ mutually exclusive for all i and j such that $i \neq j$.

We define a trajectory as $\tau = (s_1, s_2, \dots, s_H)$ in a space $\mathcal{T} = \mathcal{S} \times \dots \times \mathcal{S}$, along with a function indicating that a trajectory contains a collision involving the ego vehicle:

$$Y_h(\tau) = \begin{cases} 1, & \text{if } C(s_t) = 1 \text{ and } t \geq h \text{ for any } s_t \in \tau \\ 0, & \text{otherwise} \end{cases}. \quad (6)$$

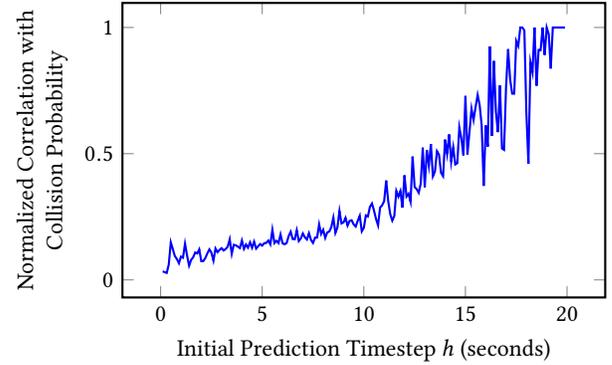


Figure 1: This plot shows, for a dataset of 70,000 vehicle trajectories each simulated 100 times, the maximum correlation across all behavioral features with rear-end collisions, normalized by the maximum correlation across all features as a function of the prediction horizon. The upward trend of this plot indicates that as h increases, behavioral features become increasingly informative of collision risk.

We can define the value as the probability of a collision in a trajectory from an initial state s_t . Hence (5) can be rewritten as

$$V^{\pi_\theta}(s_t) = P(Y_h(\tau) = 1 \mid S_t = s_t, \pi_\theta). \quad (7)$$

3.3 Traits of Intermediate-Horizon Automotive Risk Prediction

In this section, we discuss three traits of intermediate-horizon automotive risk prediction that will later inform our approach.

3.3.1 Rarity of Collisions. Validating autonomous systems frequently involves analyzing critical events that are rare. Formally, define the state-visitation distribution [42]

$$\rho_{\pi_\theta}(s_t) = \frac{1}{H} \sum_{k=1}^H P(S_k = s_t \mid \pi_\theta). \quad (8)$$

The expected value, $\sum_{s_t} \rho(s_t)C(s_t)$, is extremely small. This poses a challenge for learning to predict risk using real-world data, and for Monte Carlo policy evaluation in simulation.

3.3.2 State Space Size. The state space in automotive risk prediction is high-dimensional and contains continuous variables. These factors motivate a method of generalizing risk estimates across scenes. Accomplishing this using function approximation implies learning a predictive model for risk conditioned on the scene.

3.3.3 Influence of Behavioral Parameters. In automotive risk prediction, behavioral parameters, θ , of vehicles in the scene increase in importance with the prediction horizon. Figure 1 demonstrates this trend in an artificial¹ dataset. This fact, coupled with the relatively high-dimensional nature of behavioral features and the need to account for these features for all neighboring vehicles in the scene, makes intermediate-horizon automotive risk prediction a high-dimensional problem.

¹We use artificial data due to the unavailability of collision-inclusive real-world data. Section 4.1 discusses the method used for generating this data in detail.

Table 1: Scene Vehicle Features

Feature Name	Symbol	Description
fore distance	s_f	distance to vehicle in front
fore velocity	v_f	velocity of vehicle in front
relative velocity	Δv	rear velocity minus fore velocity
length	l	length of vehicle
width	w	width of vehicle
attentiveness	att	whether or not driver is attentive
aggressiveness	agg	aggressiveness of driver

4 APPROACH

This section first introduces the simulator we use for data generation, which entails defining the components of the MDP. We then discuss our approach to policy evaluation (i.e., risk estimation). Finally, we describe how we fit a model to risk estimates in order to allow for real-time prediction.

4.1 Simulator

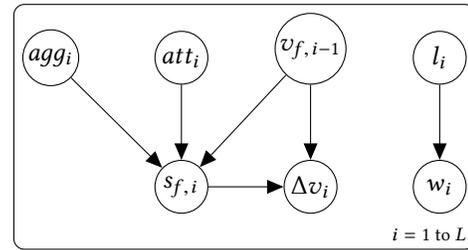
We define the simulator by specifying an initial scene distribution, ρ_1 , and the transition function, $P(s' | s, a)$. This second component is composed of the physical dynamics model, and the driver behavior models that perform decision making for vehicles.

4.1.1 Scene Generation. Scene generation involves specifying the initial state distribution, ρ_1 of the MDP. As previously defined, the state includes the physical attributes and behavioral parameters, θ , of vehicles in the scene. We assume a scene decomposes into L individual vehicles as $S = \{S^{(1)}, S^{(2)}, \dots, S^{(L)}\}$.

A simple approach to scene generation is to select from a database of real scenes with inferred behavioral features. The problems with this approach are that (i) scene data must be available and (ii) only previously observed scenes can be sampled, meaning generalization to new roadways is not possible. Scenes may instead be initialized heuristically, or according to a constant configuration that is simulated for a burn-in period. While these approaches are simple, specifying rules for initial configurations can be challenging, and simulation-based approaches rely on driver behavior models to produce realistic distributions over scenes.

Alternatively, a generative model for scenes can be learned from data. This approach allows for generalization to new settings, and provides likelihood estimates of scenes. For these reasons, we employ a learned generative model in the form of a Bayesian network [27], which defines a probability distribution over a set of variables S as a product of conditional probability distributions. The distribution of each variable depends on the values of its parent variables as defined by a directed acyclic graph. The distribution decomposes as $P(S) = \prod_{S^{(i)} \in S} P(S^{(i)} | \text{parents}(S^{(i)}))$.

Forward sampling allows for efficient generation of scenes from a Bayesian network provided that all observed nodes precede sampled nodes in a topological ordering [27]. We implement a single-lane, highway model similar to that of Wheeler et al. [56], with the primary difference being that we incorporate behavioral features in the model. Table 1 describes the variables of the network, and Figure 2 shows the plate model used for scene generation.

**Figure 2: Bayesian network single-lane scene model.**

To sample a lane from the model, the variables of the first vehicle are sampled after marginalizing v_f . Subsequent vehicles are then sampled conditioning on the velocity of the prior vehicle. The scene distribution decomposes as

$$\rho_1(S) = P(S^{(1)}) \prod_{i=2}^L P(S^{(i)} | S^{(i-1)}). \quad (9)$$

The scene model exclusively contains discrete variables. For sampling continuous values, the variables define bounds of a uniform distribution from which the value is sampled. This approach allows for the use of discrete variables, while still approximating arbitrary distributions as the discretization granularity increases, and has been used in the context of aircraft scene generation [26].

4.1.2 Driver Behavior Models. The previous section described how a scene is generated. We now discuss how this scene is simulated to produce vehicle trajectories. At each timestep, each vehicle samples a longitudinal and lateral acceleration determined by the driver models of that vehicle. For the longitudinal model, we employ a collision-inclusive variant of the IDM, based upon the Errorable Driver Model [58]. This model introduces collisions through a reaction time parameter that delays observations. The model also includes attentiveness parameters that determine the probability of a driver becoming distracted and not updating its action, as well as the probability of a driver becoming attentive from an inattentive state. We use MOBIL for lane changes, and sample longitudinal and lateral accelerations from Gaussian distributions. The means of these distributions are the acceleration values selected by the driver models, and the standard deviations are 0.5 m/s^2 and 0.1 m/s^2 for IDM and MOBIL, respectively.

For generating IDM and MOBIL parameters, we employ a correlated behavior model similar to that used in Sunberg et al. [48]. This approach samples, for each vehicle, an aggressiveness parameter from a uniform distribution over the unit interval. Aggressiveness then determines the mean of a truncated Gaussian distribution by interpolating between bounds for IDM and MOBIL as specified in Table 2. The standard deviation of this distribution is set to be 0.03 times the range of values. Given these actions, the simulator then propagates vehicles through space in discrete time increments of 0.1 seconds using a simple bicycle model [41].

4.2 Risk Estimation

This section discusses our approach to Monte Carlo policy evaluation, which we refer to as risk estimation in the automotive context.

Table 2: Driver model parameters and associated bounds

IDM Parameter	Most Agg.	Least Agg.
maximum acceleration (m/s ²)	6.0	2.0
desired velocity (m/s)	35	25
min distance to fore vehicle (m)	0	4
safe time headway (s)	0.2	1
comfortable deceleration (m/s ²)	5	2
MOBIL Parameter	Most Agg.	Least Agg.
politeness	0.1	0.5
safe deceleration (m/s ²)	2.0	2.0
advantage threshold (m/s ²)	0.01	0.7
Errorable Parameter	Value	
reaction time (s)	0.2	
$p(\text{inattentive}_{t+1} \mid \text{attentive}_t)$	0.05	
$p(\text{attentive}_{t+1} \mid \text{inattentive}_t)$	0.3	

4.2.1 Monte Carlo Simulation. The previous sections described the initial scene generation, driver behavior, and dynamics, which together specify the environment. Given this simulator, we can perform policy evaluation to estimate risk. We use Monte Carlo policy evaluation because it imposes minimal restrictions on the type of driver model used.

Our goal is to compute the value of a policy, which is the probability that that policy is involved in a collision conditional on a scene s . Ideally, we would compute this probability exactly:

$$P(Y_h(\tau) = 1 \mid S = s, \pi_\theta) = \mathbb{E}_{\tau \sim P(T=\tau \mid S=s), \pi_\theta} [Y_h(\tau)]. \quad (10)$$

This value cannot be computed in closed form, so we instead approximate the value through sampling n trajectories and averaging:

$$\mathbb{E}_{\tau \sim p(T=\tau \mid S=s), \pi_\theta} [Y_h(\tau)] \approx \frac{1}{n} \sum_{i=1}^n Y_h(\tau_i), \quad (11)$$

where $\tau_i \sim P(T = \tau \mid S = s_i, \pi_\theta)$.

4.2.2 Importance Sampling. Generating collision-inclusive data from the learned scene and driver models can require a large number of samples from (i) the scene generation model or (ii) the driver and dynamics models. We would like to reduce the quantity of samples needed so as to improve computational efficiency. Importance sampling samples collisions or other events with greater frequency relative to a baseline probability distribution, and then reweighs those samples according to their relative likelihood, thereby producing a lower variance, unbiased estimator [16]. This method can be applied in either of the sampling cases, but we focus on sampling initial scenes from the generative model.

Our approach to importance sampling biases sampling towards scenes that are dangerous with respect to the ego vehicle. Formally, the challenge is that the unconditional probability of a collision

$$P(Y_h(\tau) = 1) = \mathbb{E}_{s \sim \rho_1(S=s), \tau \sim P(T=\tau \mid S=s), \pi_\theta} [Y_h(\tau)]$$

is small. Instead of sampling s from $\rho_1(S)$, we sample from a proposal distribution $Q(S)$, which is biased towards dangerous scenes:

$$p(Y_h(\tau) = 1) = \mathbb{E}_{s \sim Q(S=s), \tau \sim P(T=\tau \mid S=s), \pi_\theta} \left[\frac{\rho_1(S=s)}{Q(S=s)} Y_h(\tau) \right].$$

This method has been employed in evaluating the safety of autonomous systems, for example, vehicles [60] and aircraft [24]. In these settings, importance sampling results in a lower variance estimator of the *unconditional* probability of collision or other event.

In our setting, we are *not* interested in computing the *unconditional* probability of a collision. Instead, we are interested in fitting a predictive model to the *conditional* probability of a collision with the goal of achieving high performance on certain evaluation metrics. Due to this altered goal, importance sampling in this context can be interpreted as a heuristic approach to active learning [9]. From this perspective, we make the assumption that dangerous scenes will improve performance by subjecting the model to a greater number of positive-risk events, thereby addressing the low sample size problem associated with those events.

4.2.3 Cross Entropy Method. A remaining task is determining the proposal distribution $Q(S)$. For this distribution, we use another Bayesian network, the conditional probability distributions of which have been altered to generate collisions more frequently. Because a scene filled with dangerous vehicles would result in extremely low likelihood values, we instead sample a single vehicle in each scene from Q , and the remaining vehicles as usual from ρ_1 . Recall that the scene model decomposes across the L vehicles as (9). The proposal distribution $Q(S)$ differs from $\rho_1(S)$ only in the j th vehicle sampled, and thus all vehicle probabilities cancel in the ratio of ρ_1 and Q except for the j th vehicle:

$$\frac{\rho_1(S)}{Q(S)} = \frac{P(S^{(j)} \mid S^{(j-1)})}{Q(S^{(j)} \mid S^{(j-1)})}.$$

We refer to this likelihood ratio as w .

We could manually alter the Bayesian network parameters to increase the likelihood of collisions involving the ego vehicle, but we would like to automate this process. We accomplish this using the cross entropy method (CEM).

CEM [43] is an optimization algorithm originally designed for finding proposal distributions in the context of rare events, which operates by iteratively sampling from some distribution, and then updating the parameters of that distribution based on the fitness of the samples [11]. In the particular instantiation of CEM we employ, the proposal distribution is trained to produce collisions involving the ego vehicle within 10 to 20 seconds by changing the conditional probability distributions of the variables in the Bayesian network.

Figure 3 shows the mean values of three variables and collision probability during the optimization process. The changes to these variables, which may initially seem counterintuitive, result in a higher probability of collision. The reason for this is that only intermediate-horizon collisions (i.e., those occurring in the 10 to 20 second range) are counted. Thus, the method learns to avoid early collisions (i.e., those occurring before 10 seconds) by making the ego vehicle both attentive and slower than the vehicle in front. Because the ego vehicle initially travels at a lower velocity than the fore vehicle, it tends to speed up. If the ego vehicle then becomes inattentive, it will continue accelerating until colliding with the fore

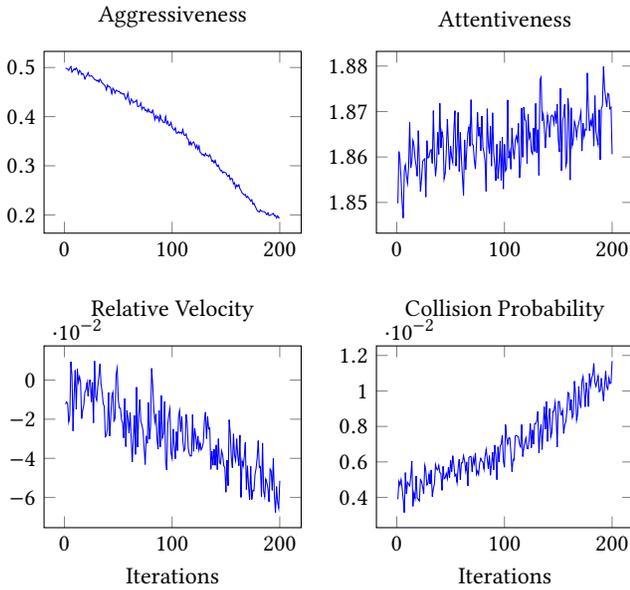


Figure 3: Mean values for three of the Bayesian network variables and collision probability throughout optimization.

vehicle. Aggressiveness decreases because this reduces the initial acceleration (thereby preventing early collisions), as well as the magnitude of the comfortable deceleration rate (thereby limiting the ability of the ego vehicle to slow down).

A dataset of scenes is sampled from the learned distributions ρ_1 and Q , and each scene evaluated using Monte Carlo simulation. The resulting dataset $\{s^{(i)}, y^{(i)}, w^{(i)}\}_{i=1}^m$, containing m scene-risk pairs, each with an associated likelihood ratio w , is then used to augment real data in learning a predictive model.

4.3 Risk Prediction

Risk estimation must be performed in real-time on-vehicle. A common approach to this task is to simulate future trajectories online to derive a risk estimate [2, 38, 59]. While this method may be feasible for short-term risk estimation, we believe that longer-horizon risk is impractical to estimate in this manner. There are two reasons why this would be the case. First, because driver behavior increases in significance with longer horizons, effective simulators will likely use learned, computationally expensive driver models [29]. Second, and more challenging, is the number of simulations that will need to be run due to the prediction horizon in order to arrive at estimates of risk with low relative error. We propose to instead fit a predictive model offline, the online computational complexity of which grows slowly as a function of simulator complexity and the number of simulations run. The following sections describe this prediction model, and our method for compensating for discrepancies between simulated and real-world data.

4.3.1 Prediction Model. The response variable Y in collected datasets is the estimated proportion of successes p (e.g., collisions) in a binomial experiment with n samples. Because this value is estimated from a finite number of samples, it can only assume a

finite number of values, and is therefore a discrete variable. This variable has two unusual properties. First, valid values are bounded between 0 and 1 inclusively. Second, in real-world data, predicting this variable reduces to binary classification.

Zhao et al. [61] compare linear and logistic regression in fitting proportion data, finding logistic regression to have better performance [61]. Three alternative approaches are (i) to take $\log(Y)$ as the response and fit a linear regression model; (ii) to perform beta regression of the values, which requires a transform of response values of 0 or 1 [13]; and (iii) to treat the response as counts and fit a negative binomial regression [39], which would complicate transfer to real-world data where the sample size n differs from simulation. While these alternatives may have merit, we elect to use the cross entropy loss because of its simplicity and good performance in initial experiments.

In risk *estimation*, we considered the risk associated with the ego vehicle in a scene s . In risk *prediction*, we take the perspective of the ego vehicle directly to reflect the information available in a real-world situation. We refer to the features from this perspective with the random variable X , which we assume is still a Markovian representation of the environment.

We fit a predictive model $V(x; \phi)$ with parameters ϕ as the value function, minimizing the loss

$$\mathcal{L}_{pred}(X, Y) = -\mathbb{E}_{(x, y) \sim (X, Y)} [y \log(V(x; \phi)) + (1 - y) \log(1 - V(x; \phi))]. \quad (12)$$

We empirically evaluate this loss on the collected dataset as

$$-\sum_{i=1}^m w^{(i)} [y^{(i)} \log(V(x^{(i)}; \phi)) + (1 - y^{(i)}) \log(1 - V(x^{(i)}; \phi))]. \quad (13)$$

This model is fit to a dataset containing risk estimates for many different policy parameters, θ . We include these parameters as features, effectively learning a value function over a distribution of policies. In all experiments, we use a neural network prediction model with logistic activation function. This choice of model is motivated by the high-dimensional feature space, and the ability to employ recent domain adaptation approaches as we discuss next.

4.3.2 Domain Adaptation. Even high-fidelity simulations will necessarily differ from the real-world. Formally, the joint distribution of the simulated, ego-centric scene representations X_s and risk estimates Y_s , will differ from the (target t) real-world joint distribution: $P_s(X_s, Y_s) \neq P_t(X_t, Y_t)$. This “domain shift” degrades performance when transferring from the source to target domain [44].

Domain adaptation (DA) methods attempt to address this problem. Both supervised and unsupervised variants exist, and it is not immediately clear which approach is better suited for automotive risk prediction. The reason for this is that the target, real-world automotive domain can be viewed as “weakly supervised” in the sense that its risk estimates are highly imprecise reflections of the true underlying collision probability.

While supervised DA methods have been shown to be effective [36], we focus on an unsupervised approach, domain adversarial neural networks (DANN) [14]. DANN attempts to learn features that are (i) invariant to the domain and (ii) useful for the task being performed. To accomplish this task, the activations of certain layers in the network, M_s and M_t , are encouraged to match across

Table 3: Prediction Features

Feature Name	Example Quantities
Ego physical	lane offset and heading, velocity, vehicle length and width, acceleration, turn rate, time-to-collision
Ego well-behaved	ego vehicle currently colliding, out of lane, or has a negative velocity
Ego behavioral	IDM and MOBIL parameters
Neighboring vehicle state	same physical and behavioral features as ego vehicle and relative dist. to ego

domains. This requires a measure of similarity, and for that DANN employs a domain classification network, D , that attempts to distinguish between the learned features of the two domains. The risk prediction network incurs an additional adversarial loss

$$\begin{aligned} \mathcal{L}_{adv}(X_s, X_t, M_s, M_t) & \quad (14) \\ &= -\mathbb{E}_{x_s \sim X_s} [\log D(M_s(x_s))] - \mathbb{E}_{x_t \sim X_t} [\log(1 - D(M_t(x_t)))], \end{aligned}$$

with D being trained to minimize the negative. The overall objective then becomes $\mathcal{L} = \mathcal{L}_{pred} + \lambda \mathcal{L}_{adv}$, where λ controls the weight of the adversarial loss.

Minimizing this loss with respect to both M_s and M_t encourages the source and target distributions to match, but this might come at the cost of learning a good target feature distribution. In the supervised setting, it is possible to only optimize the adversarial loss with respect to the source feature distribution M_s , and we consider this approach as well in our experiments.

5 EXPERIMENTS AND RESULTS

Our goal is to assess whether simulated automotive data can improve intermediate-horizon risk prediction in real-world data. We perform two experiments. The first experiment employs the full approach as described, and seeks to determine whether importance sampled collision events can improve prediction in an artificial scenario. The second experiment considers the simulated to real transfer challenge using real-world vehicle trajectory data.

Throughout the experiments, we use a set of scene features, which we summarize in Table 3. Our primary focus in this research is on addressing the challenge of collision rarity, but not on partial observability. As a result, we assume access to features that are difficult to obtain in real scenarios, for example, information about vehicles far in front of the ego vehicle, and behavioral features that typically must be inferred [48].

5.1 Simulation Experiments

In this section, we assess whether importance sampling of collisions in artificial settings can improve prediction performance. Our focus on exclusively simulated data in this experiment is motivated by the unavailability of real-world data containing collisions.

We generate an artificial dataset to act as the “real” data containing few collisions. We generate this data heuristically, randomly initializing vehicles and their behavioral parameters on a single lane, circular track, which is simulated for 600 timesteps to arrive

at a random configuration. This configuration is then simulated 100 times for 200 timesteps, and risk estimates for the period between 100 and 200 timesteps are computed. Each scene involves 70 vehicles, and we take each vehicle as a single sample. This produces a dataset in which samples are not independent since they may both be from the same scene. For the sake of efficiency, we ignore this lack of independence in the training data; however, the validation set contains samples generated from entirely different simulations from those of the training set.

We apply the proposed system by fitting a scene model, running the cross entropy method to derive a proposal distribution, generating a dataset, and learning predictive models. We compare four methods: (i) training only on the target domain, (ii) training on both domains without adaptation, (iii) training on both domains with adaptation, and (iv) training on both domains, applying adversarial loss only to the source features. Because the number of collision events in the target domain plays a critical role in determining the value of simulated data, we compare performance between the models for various numbers of target domain collision events. To better reflect real-world data, we convert the target *training* set into binary values by sampling collisions according to the observed probabilities. The validation set we leave as continuous values because this provides greater precision during evaluation.

A hyperparameter search was performed for each method over a predefined set of three neural network architectures, learning rates between 10^{-4} and 10^{-3} , and dropout probabilities between 0.5 and 1. The three architectures consisted of encoding hidden layers containing (512, 256, 128, 64), (256, 128, 64), or (128, 64) units, and classifier hidden layers containing either no hidden units (i.e., direct classification of encoder output), (64) or (64, 64) hidden units. All networks used rectified linear activations, and for each training run the best validation performance during the run was selected for use in the results. The domain adversarial loss was applied to the features output from the encoder. Thirty networks were trained for each method. Figure 4(a) shows the average negative log likelihood evaluated on the validation set across these training runs.

These results support two main conclusions. First, all models that learn from simulated data exhibit improved performance on positive-risk instances, and source-only adaptation and no adaptation improve over target-only training overall. This result is likely due to the fact that the jointly-trained models observe a far greater number of positive-risk samples from the source dataset, thereby enabling them to learn a better predictive model. Second, when few positive-risk samples are provided in the training set, the adaptation models achieve the best performance. With more positive-risk samples, the model without adaptation performs best across all validation samples, likely because this model is better able to decide what source data to leverage, rather than being explicitly constrained to learning a shared feature space.

5.2 Real-world Transfer Experiments

We next seek to determine whether simulated automotive data can be transferred to a prediction task involving real-world data. We consider the task of predicting low time-to-collision (TTC) events [15], which we define as a vehicle having a TTC less than 3.0 seconds, in the US Highway 101 portion of the Next-Generation

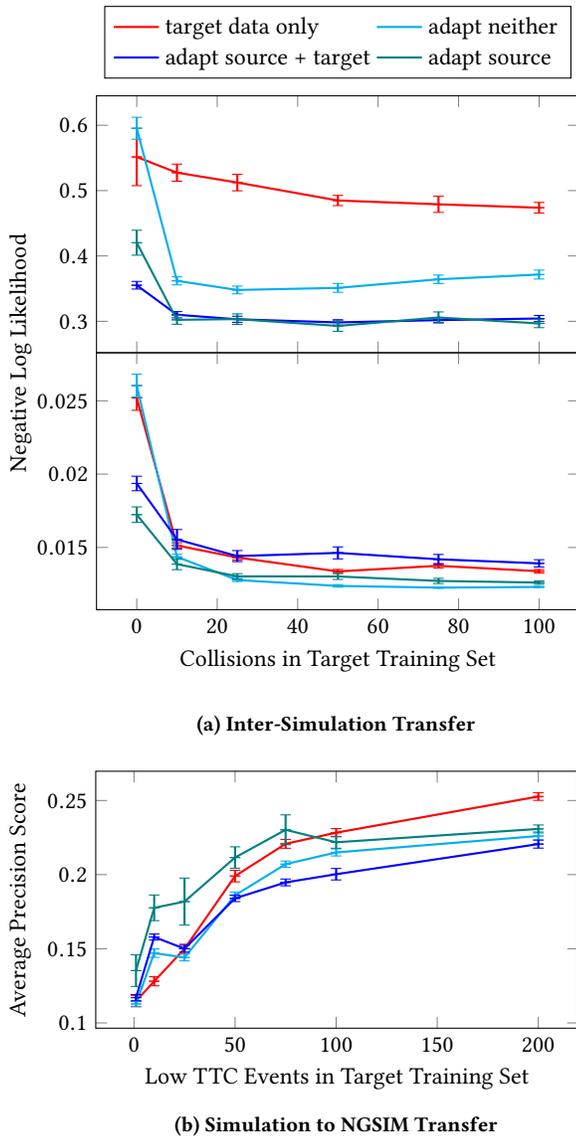


Figure 4: These plots show validation results with varying amounts of *positive-risk* samples in the target domain training set. Each point represents the mean performance across runs, with vertical bars indicating standard error. The top and bottom plots of (a) show performance evaluated on *positive-risk* and all target domain validation samples, respectively. Plot (b) shows results for transferring from simulation to NGSIM data.

Simulation (NGSIM) dataset [1]. NGSIM contains vehicle trajectories on a five-lane highway, collected over three 15-minute time periods. These trajectories contain no collisions, so we instead focus on a collision surrogate. Furthermore, low TTC events are not sufficiently rare to merit application of importance sampling, so we generate 50,000 source samples heuristically using the same method as used in the simulated experiments for mimicking real-world data. Because NGSIM contains binary labels, we now evaluate

classification performance using the average precision score [46], which we choose in order to better account for class imbalance.

The NGSIM dataset does not contain explicit behavioral parameters, so we first extract IDM parameters using the local maximum likelihood method of Kesting et al. [22]. We smooth the U.S. Highway 101 subset of the NGSIM data according to the method of Theimann et al. [50], and then extract samples from it in 20 second increments. We train on the first and third time periods of the NGSIM dataset, and validate on the second.

We again compare performance of the four methods across varying amounts of positive-risk target data, and visualize results in Figure 4(b). These results indicate that when training data is limited, simulated data can significantly improve prediction performance. As the number of positive-risk target samples increases, the target-only model performs best because enforcing domain-invariance begins to outweigh the benefits of additional positive-risk samples from simulated data. Approximately 5% of the samples have a positive response value. In this imbalanced setting, a random classifier achieves an average precision score of 0.05. All models therefore perform significantly above random, but nevertheless quite poorly due to the high-variance nature of the *classification* task.

6 CONCLUSION

This paper aimed to answer the question of whether simulated data could be used to improve automotive risk prediction. We concluded that simulated data can be helpful in cases where the amount of data available is insufficient to learn a good predictive model. We demonstrated that intermediate-horizon prediction suffers from this low sample size issue due to the significance of high-dimensional behavioral features in learning a predictive model. We then showed that domain adaptation methods could be used with simulated data to improve prediction in both artificial and real-world settings. The significance of this result is that systems that perform intermediate-horizon risk prediction may potentially be improved through the offline training of predictive models on simulated data.

Our approach can be improved through more sophisticated modeling, estimation, and prediction. Initial scene generation could employ the factor graph approach of Wheeler et al. [55] or deep generative models. Driver models learned from data [29] are likely to provide substantial benefit in transfer performance by better capturing human failure modes. A multi-lane approach to importance sampling would allow for application of the system to real-life collision prediction transfer.

Prediction performance could be enhanced through the use of domain adaptation models that explicitly model shared and private latent feature spaces [5]. Finally, the local maximum likelihood estimation approach to inferring behavioral parameters of NGSIM vehicles is limited because the model poorly captures individual driving behavior, and it does not generalize across drivers. A method proposed by Morton et al. [34] would allow for arbitrarily complex behavior encodings, while also generalizing across samples, thereby alleviating issues resulting from the lack of individual-vehicle data.

7 ACKNOWLEDGMENT

The authors wish to thank Gina Madigan, Tim Wheeler, Zach Sunberg, and Katie Driggs-Campbell for helpful conversations and feedback. This research was funded by The Allstate Corporation.

REFERENCES

- [1] Vassili Alexiadis, James Colyar, John Halkias, Rob Hranac, and Gene McHale. 2004. The next generation simulation program. *Institute of Transportation Engineers. ITE Journal* 74, 8 (2004), 22–26.
- [2] Matthias Althoff and Alexander Mergel. 2011. Comparison of Markov chain abstraction and Monte Carlo simulation for the safety assessment of autonomous cars. *IEEE Transactions on Intelligent Transportation Systems* 12, 4 (2011), 1237–1247.
- [3] Samer Ammoun and Fawzi Nashashibi. 2009. Real time trajectory prediction for collision risk estimation between vehicles. In *IEEE Intelligent Computer Communication and Processing*. 417–422.
- [4] Richard Bellman. 1957. A Markovian decision process. *Journal of Mathematics and Mechanics* 6, 5 (1957), 679–684.
- [5] Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. 2016. Domain separation networks. In *Advances in Neural Information Processing Systems (NIPS)*. 343–351.
- [6] Adrian Broadhurst, Simon Baker, and Takeo Kanade. 2005. Monte Carlo road safety reasoning. In *IEEE Intelligent Vehicles Symposium*. 319–324.
- [7] Li-Yen Chang. 2005. Analysis of freeway accident frequencies: negative binomial regression versus artificial neural network. *Safety Science* 43, 8 (2005), 541–557.
- [8] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research* 16 (2002), 321–357.
- [9] David A Cohn, Zoubin Ghahramani, and Michael I Jordan. 1996. Active learning with statistical models. *Journal of Artificial Intelligence Research* (1996), 129–145.
- [10] Gary A Davis, John Hourdos, Hui Xiong, and Indrajit Chatterjee. 2011. Outline for a causal model of traffic conflicts and crashes. *Accident Analysis & Prevention* 43, 6 (2011), 1907–1919.
- [11] Pieter-Tjerk De Boer, Dirk P Kroese, Shie Mannor, and Reuven Y Rubinfeld. 2005. A tutorial on the cross-entropy method. *Annals of Operations Research* 134, 1 (2005), 19–67.
- [12] Andreas Eidehall and Lars Petersson. 2008. Statistical threat assessment for general road scenes using Monte Carlo sampling. *IEEE International Conference on Intelligent Transportation Systems (ITSC)* 9, 1 (2008), 137–147.
- [13] Silvia Ferrari and Francisco Cribari-Neto. 2004. Beta regression for modelling rates and proportions. *Journal of Applied Statistics* 31, 7 (2004), 799–815.
- [14] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *Journal of Machine Learning Research* 17, 59 (2016), 1–35.
- [15] Douglas Gettman and Larry Head. 2003. Surrogate safety measures from traffic simulation models. *Transportation Research Record: Journal of the Transportation Research Board* 1840 (2003), 104–115.
- [16] Peter W Glynn and Donald I Iglehart. 1989. Importance sampling for stochastic simulations. *Management Science* 35, 11 (1989), 1367–1392.
- [17] Feng Guo and Youjia Fang. 2013. Individual driver risk assessment using naturalistic driving data. *Accident Analysis & Prevention* 61 (2013), 3–9.
- [18] Feng Guo, Sheila Klauer, Jonathan Hankey, and Thomas Dingus. 2010. Near crashes as crash surrogate for naturalistic driving studies. *Transportation Research Record: Journal of the Transportation Research Board* 2147 (2010), 66–74.
- [19] Abhishek Gupta, Coline Devin, YuXuan Liu, Pieter Abbeel, and Sergey Levine. 2017. Learning invariant feature spaces to transfer skills with reinforcement learning. *International Conference on Learning Representations (ICLR)* (2017).
- [20] John Hourdos, Vishnu Garg, Panos Michalopoulos, and Gary Davis. 2006. Real-time detection of crash-prone conditions at freeway high-crash locations. *Transportation Research Record: Journal of the Transportation Research Board* 1968 (2006), 83–91.
- [21] Nico Kaempchen, Kristian Weiss, Michael Schaefer, and Klaus CJ Dietmayer. 2004. IMM object tracking for high dynamic driving maneuvers. In *IEEE Intelligent Vehicles Symposium*. 825–830.
- [22] Arne Kesting and Martin Treiber. 2008. Calibrating car-following models by using trajectory data: Methodological study. *Transportation Research Record: Journal of the Transportation Research Board* 2088 (2008), 148–156.
- [23] Arne Kesting, Martin Treiber, and Dirk Helbing. 2007. General lane-changing model MOBIL for car-following models. *Transportation Research Record: Journal of the Transportation Research Board* 1999 (2007), 86–94.
- [24] Youngjun Kim and Mykel J Kochenderfer. 2016. Improving aircraft collision risk estimation using the cross-entropy method. *Journal of Air Transportation* (2016).
- [25] Mykel J Kochenderfer. 2015. *Decision Making Under Uncertainty: Theory and Application*. MIT Press.
- [26] Mykel J Kochenderfer, Matthew WM Edwards, Leo P Espindle, James K Kuchar, and J Daniel Griffith. 2010. Airspace encounter models for estimating collision risk. *Journal of Guidance, Control, and Dynamics* 33, 2 (2010), 487–499.
- [27] Daphne Koller and Nir Friedman. 2009. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press.
- [28] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*. 1097–1105.
- [29] Alex Kuefler, Jeremy Morton, Tim A Wheeler, and Mykel J Kochenderfer. 2017. Imitating driver behavior with generative adversarial networks. In *IEEE Intelligent Vehicles Symposium*. 204–211.
- [30] Christian Laugier, Igor E Paromtchik, Mathias Perrollaz, Mao Yong, John-David Yoder, Christopher Tay, Kamel Mekhnacha, and Amaury Nègre. 2011. Probabilistic analysis of dynamic scenes and collision risks assessment to improve driving safety. *IEEE Intelligent Transportation Systems Magazine* 3, 4 (2011), 4–19.
- [31] Stéphanie Lefèvre, Dizan Vasquez, and Christian Laugier. 2014. A survey on motion prediction and risk assessment for intelligent vehicles. *Robotics Journal* 1, 1 (2014), 1.
- [32] Xiugang Li, Dominique Lord, Yunlong Zhang, and Yuanchang Xie. 2008. Predicting motor vehicle crashes using support vector machine models. *Accident Analysis & Prevention* 40, 4 (2008), 1611–1618.
- [33] Dominique Lord and Fred Mannering. 2010. The statistical analysis of crash-frequency data: a review and assessment of methodological alternatives. *Transportation Research Part A: Policy and Practice* 44, 5 (2010), 291–305.
- [34] Jeremy Morton and Mykel J Kochenderfer. 2017. Simultaneous policy learning and latent state inference for imitating driver behavior. In *IEEE International Conference on Intelligent Transportation Systems (ITSC)*. 1–6.
- [35] Jeremy Morton, Tim A Wheeler, and Mykel J Kochenderfer. 2017. Analysis of recurrent neural networks for probabilistic modeling of driver behavior. *IEEE Transactions on Intelligent Transportation Systems* 18, 5 (2017), 1289–1298.
- [36] Saeid Motiian, Marco Piccirilli, Donald A Adjeroh, and Gianfranco Doretto. 2017. Unified deep supervised domain adaptation and generalization. In *IEEE International Conference on Computer Vision (ICCV)*.
- [37] Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering* 22, 10 (2010), 1345–1359.
- [38] Jae-Hyuck Park and Yu-Wing Tai. 2015. A simulation based method for vehicle motion prediction. *Computer Vision and Image Understanding* 136 (2015), 79–91.
- [39] Mark Poch and Fred Mannering. 1996. Negative binomial analysis of intersection-accident frequencies. *Journal of Transportation Engineering* 122, 2 (1996), 105–113.
- [40] Martin L Puterman. 2014. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons.
- [41] Rajesh Rajamani. 2011. *Vehicle Dynamics and Control*. Springer Science & Business Media.
- [42] Stéphane Ross, Geoffrey J Gordon, and Drew Bagnell. 2011. A reduction of imitation learning and structured prediction to no-regret online learning. In *International Conference on Artificial Intelligence and Statistics*. 627–635.
- [43] Reuven Y Rubinfeld. 1997. Optimization of computer simulation models with rare events. *European Journal of Operational Research* 99, 1 (1997), 89–112.
- [44] Andrei A Rusu, Matej Vecerik, Thomas Rothörl, Nicolas Heess, Razvan Pascanu, and Raia Hadsell. 2017. Sim-to-Real Robot Learning from Pixels with Progressive Nets. In *Conference on Robot Learning (CoRL)*.
- [45] Matthias Schreier, Volker Willert, and Jürgen Adamy. 2014. Bayesian, maneuver-based, long-term trajectory prediction and criticality assessment for driver assistance systems. In *IEEE International Conference on Intelligent Transportation Systems (ITSC)*. 334–341.
- [46] Himrich Schütze, Christopher D Manning, and Prabhakar Raghavan. 2008. *Introduction to information retrieval*. Vol. 39. Cambridge University Press.
- [47] Baochen Sun and Kate Saenko. 2014. From Virtual to Reality: Fast Adaptation of Virtual Object Detectors to Real Domains. In *Proceedings of the British Machine Vision Conference*.
- [48] Zachary N Sunberg, Christopher J Ho, and Mykel J Kochenderfer. 2017. The value of inferring the internal state of traffic participants for autonomous freeway driving. In *American Control Conference (ACC)*. 3004–3010.
- [49] Andrew P Tarko. 2012. Use of crash surrogates and exceedance statistics to estimate road safety. *Accident Analysis & Prevention* 45 (2012), 230–240.
- [50] Christian Thiemann, Martin Treiber, and Arne Kesting. 2008. Estimating acceleration and lane-changing dynamics from next generation simulation trajectory data. *Transportation Research Record: Journal of the Transportation Research Board* 2088 (2008), 90–101.
- [51] Martin Treiber, Ansgar Hennecke, and Dirk Helbing. 2000. Congested traffic states in empirical observations and microscopic simulations. *Physical Review E* 62, 2 (2000), 1805–1824.
- [52] Simon Ulbrich, Till Menzel, Andreas Reschka, Fabian Schuldt, and Markus Maurer. 2015. Defining and substantiating the terms scene, situation, and scenario for automated driving. In *IEEE International Conference on Intelligent Transportation Systems (ITSC)*. 982–988.
- [53] U.S. Department of Transportation, National Highway Traffic Safety Administration. 2016. 2016 Quick Facts. <https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812451>. (2016). Accessed: 2017-11-14.
- [54] David Vazquez, Antonio M Lopez, Javier Marin, Daniel Ponsa, and David Geronimo. 2014. Virtual and real world adaptation for pedestrian detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36, 4 (2014), 797–809.
- [55] Tim A Wheeler and Mykel J Kochenderfer. 2016. Factor Graph Scene Distributions for Automotive Safety Analysis. In *IEEE International Conference on Intelligent Transportation Systems (ITSC)*.

- [56] Tim A Wheeler, Mykel J Kochenderfer, and Philipp Robbel. 2015. Initial scene configurations for highway traffic propagation. In *IEEE International Conference on Intelligent Transportation Systems (ITSC)*. 279–284.
- [57] Wuping Xin, John Hourdos, Panos Michalopoulos, and Gary Davis. 2008. The less-than-perfect driver: a model of collision-inclusive car-following behavior. *Transportation Research Record: Journal of the Transportation Research Board* 2088 (2008), 126–137.
- [58] Hsin-Hsiang Yang and Huei Peng. 2010. Development of an errorable car-following driver model. *Vehicle System Dynamics* 48, 6 (2010), 751–773.
- [59] William Young, Amir Sobhani, Michael G Lenné, and Majid Sarvi. 2014. Simulation of safety: A review of the state of the art in road safety simulation modeling. *Accident Analysis & Prevention* 66 (2014), 89–103.
- [60] Ding Zhao, Henry Lam, Huei Peng, Shan Bao, David J LeBlanc, Kazutoshi Nobukawa, and Christopher S Pan. 2017. Accelerated evaluation of automated vehicles safety in lane-change scenarios based on importance sampling techniques. *IEEE Transactions on Intelligent Transportation Systems* 18, 3 (2017), 595–607.
- [61] Lihui Zhao, Yuhuan Chen, and Donald W Schaffner. 2001. Comparison of logistic regression and linear regression in modeling percentage data. *Applied and Environmental Microbiology* 67, 5 (2001), 2129–2135.