AN ACTIVE INFERENCE MODEL OF CAR FOLLOWING: ADVANTAGES AND APPLICATIONS

Ran Wei

Anthony D. McDonald, Corresponding Author

Alfredo Garcia

Gustav Markkula

Johan Engström*[1]

Matthew O'Kelly*

Submission Date: March 28, 2023

[1] *Advisors

Abstract

Driver process models play a central role in the testing, verification, and development of automated and autonomous vehicle technologies. Prior models developed from control theory and physics-based rules are limited in automated vehicle applications due to their restricted behavioral repertoire. Data-driven machine learning models are more capable than rule-based models but are limited by the need for large training datasets and their lack of interpretability, i.e., an understandable link between input data and output behaviors. We propose a novel car following modeling approach using active inference, which has comparable behavioral flexibility to data-driven models while maintaining interpretability. We assessed the proposed model, the Active Inference Driving Agent (AIDA), through a benchmark analysis against the rule-based Intelligent Driver Model, and two neural network Behavior Cloning models. The models were trained and tested on a real-world driving dataset using a consistent process. The testing results showed that the AIDA predicted driving controls significantly better than the rule-based Intelligent Driver Model and had similar accuracy to the data-driven neural network models in three out of four evaluations. Subsequent interpretability analyses illustrated that the AIDA's learned distributions were consistent with driver behavior theory and that visualizations of the distributions could be used to directly comprehend the model's decision making process and correct model errors attributable to limited training data. The results indicate that the AIDA is a promising alternative to black-box data-driven models and suggest a need for further research focused on modeling driving style and model training with more diverse datasets.

Keywords: Driver behavior modeling, Active Inference, Behavior Cloning, Intelligent Driver Model, Car Following, Learning from Demonstration

1 INTRODUCTION

The rapid development of automated and connected vehicle technologies has created a corresponding demand for models of driver behavior that can be used to calibrate design parameters [1, 2], evaluate technologies [3, 4], and refine real-time decision making [5]. To be effective in these tasks, driver models must be flexible, generalizable, and interpretable. Model flexibility is the ability of the model to mimic nuanced social behavior of human drivers [6]. Generalizability is the ability of the model to extend to new environments with minimal modeler intervention. Interpretability refers to both a clear connection between model mechanics and predicted behavior and a grounding in human psychology [7]. These elements facilitate model inspection and diagnostics which are essential for interpretable models [8]. Car following is an important driving sub-task as it represents a large portion of current driving time and crashes involving automated vehicles [9, 10]. Moreover, it requires a complex expression of social behaviors through physical vehicle positioning [6], e.g., speeding up to prevent a vehicle from merging. Therefore, it is important to develop flexible, generalizable, and interpretable car following models for automated vehicles and future transportation systems.

Existing car following models can be partitioned into rule-based models and data-driven models [11, 12]. Rule-based models generate acceleration behavior based on a function specified by the modeler. Typically, this function is grounded in known observations or driver behavior theory [11]. For example, the Intelligent Driver Model (IDM) predicts driver acceleration based on deviations from a desired speed and distance headway [13]. While rule-based models have a clear connection between model mechanics and predicted behavior, they are limited in their flexibility and generalizability. Because the rules in rule-based models are designed to replicate driving behavior in specific contexts and depict driver characteristics with small parameter sets, they are limited in the behavioral repertoire and in generalizing to scenarios outside of those governed by rules beyond their initial rule set. For example, research has shown that rule-based models designed for car following do not generalize to emergency scenarios and crashes [14]. Despite these limitations, rule-based models are still widely used for automated vehicle analyses [15] and thus offer a valid benchmark for new models.

In contrast to rule-based models, data-driven models learn a function mapping observations or features to acceleration behaviors using an algorithm. Recent works have used neural networks [16], hybrid neural network algorithms with physics constraints [17], reinforcement learning [18], and adversarial imitation learning [19] to model car following behavior. These approaches have shown considerable flexibility in replicating human behavior, however, data-driven models still struggle to reproduce well-known traffic phenomena such as stop-and-go oscillation and their generalizability is constrained by the chosen machine learning technique [16, 20]. Furthermore, the complexity of existing data-driven models prohibits interpretability both in the connection between input and output and in their grounding in human psychology. Despite these shortcomings, data-driven models are more generalizable to complex scenarios which are difficult for manual model specification. One important class of data-driven models is Behavior Cloning (BC) known for their simplicity and general effectiveness [21–23]. Neural network-based BC models have been widely adopted for developing and evaluating automated vehicle algorithms and are a common benchmark for evaluating novel data-driven models [24, 25].

The relative strengths of rule-based and data-driven approaches suggest that there is a role for model structure (to aid in interpretability) — especially structure grounded in psychological theory [7] — and learning from data (to aid in flexibility) in car following model development. The incorporation of these two concepts requires a shift to contemporary theories of human cognition. One relevant theory is active inference [26, 27] — a framework developed from Bayesian principles of cognition [28, 29]. The central ideas of active inference are that 1) humans have internal probabilistic generative models of the environment and that 2) humans leverage their model of the environment to make inferences about action courses that reduce surprise in terms of both distance from their desired states of the environment and uncertainty [26, 27]. Importantly, these principles have been translated into a quantitative framework for modeling human behavior and cognition [27, 30]. The quantitative framework includes an explicit representation of agent belief dynamics to facilitate agent decision making and action selection in response to observed perceptual signals. Due to this structure, the model is fundamentally interpretable (i.e., actions can be traced back to beliefs and observations at a given time). On the other hand, the increased complexity and probabilistic nature of the model compared to rule-based frameworks also increase it's flexibility and potentially it's generalizability. Recently, the active inference framework has been extended to driving to depict driving behavior during emergency scenarios with some success [31, 32], however, the application to broader scenarios has been limited.

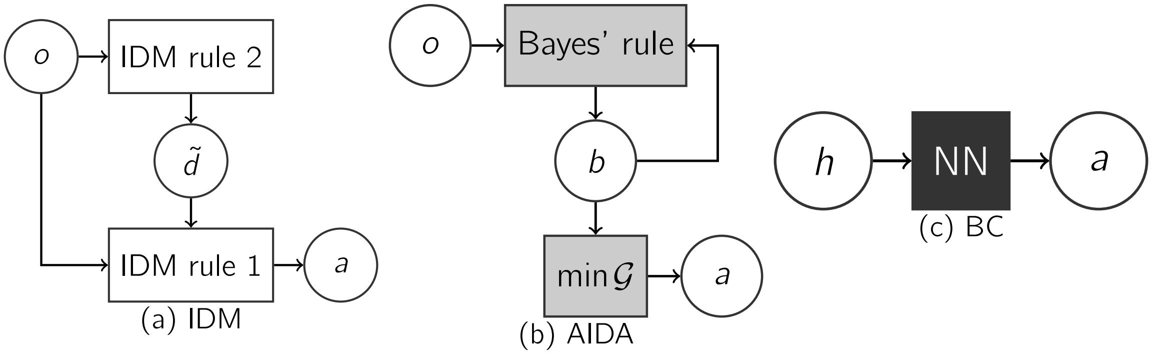
Our goal in this article is to introduce the Active Inference Driving Agent (AIDA), evaluate its flexibility and generalizability relative to rule-based and data-driven benchmarks, and illustrate the interpretability of the model and the resulting insights it provides into car following behavior.

2 MATERIALS AND METHODS

In this section, we introduce a formulation of the benchmarks — IDM and Behavior Cloning — then describe our AIDA formulation. We then describe the dataset used for model fitting and the model comparison approach. To simplify notation, we adopt a unifying view of car following models as longitudinal driving control policies which map input signals observed by drivers to a control signal, i.e., the instantaneous longitudinal acceleration. We denote the driver observations (or features in machine learning terminology) at discrete time step t by o_t and the control signal by a_t. Using this nomenclature, the most generic class of driver control policies can be described as a probabilistic mapping from the entire history of inputs and controls, denoted by $h_t = \{o_{1:t}, a_{1:t-1}\}$ to the next control signal, i.e., $\pi(a_t|h_t)$. However, the control policy may only depend on the most recent observation as $\pi(a_t|o_t)$. The definition of the control policy is the most significant element that differentiates the IDM, Behavior Cloning, and AIDA. These differences are illustrated in the computation graphs in Figure 1 and further described in the subsequent sections.

2.1 Intelligent Driver Model

The IDM [33] implements a control policy based on drivers' instantaneous observation of their own vehicle's speed v, relative speed to the lead vehicle Δv, and distance headway to the lead vehicle d, i.e., $\pi(a_t|o_t = \{v_t, \Delta v_t, d_t\})$. At each time step, the IDM computes a longitudinal



(a) IDM (b) AIDA (c) BC

FIGURE 1: Computation graphs for (a) IDM, (b) AIDA, and (c) neural network BC models. $o =$ instantaneous observation, $a =$ control action, $h =$ complete interaction history, $\tilde{d} =$ desired distance headway, $b =$ instantaneous belief, $\mathcal{G} =$ expected free energy, NN = neural network.

acceleration to regulate the driver's vehicle towards a desired speed $\tilde{v}$ and desired distance headway $\tilde{d}$ using the following control rule:

$$a_t = a_{max}\left[1 - \left(\frac{v_t}{\tilde{v}}\right)^4 - \left(\frac{\tilde{d}}{d_t}\right)^2\right] \tag{1}$$

where the desired distance headway is defined as:

$$\tilde{d} = d_0 + v_t\tau - \frac{v_t \Delta v_t}{2\sqrt{a_{max} b}} \tag{2}$$

The IDM has the following parameters: a_{max} the maximum acceleration rate which can be implemented by the driver, d_0 the minimum allowable distance headway, τ the desired headway time, and b the maximum deceleration rate. While these parameters can be set manually by human designers, they usually depend on the road condition and vary with individual driver characteristics, e.g., the desired velocity and minimum distance headway. Thus, various methods have been proposed to calibrate model parameters from traffic data [34, 35].

A significant limitation of the IDM is that it cannot express certain types of behavior as a result of the control rule defined in (1) and (2). For example, it cannot express behavior due to uncertainty about the lead vehicle behavior and surrounding traffic and is limited to modeling behavior in non-conflict scenarios [14]. Incorporation of such behavior requires significant intervention from the model designers in adapting the control rule, e.g, modifying (1) and (2) to depend on additional inputs or "memory" mechanisms [13].

2.2 Behavior Cloning

BC refers to methods that train neural networks to learn policies from datasets of human car following behavior. The dataset, denoted with $\mathcal{D}$, is usually organized in the form of observation-action trajectories, i.e., $\mathcal{D} = \{o_{1:T}^{(i)}, a_{1:T}^{(i)}\}_{i=1}^{N}$, where N is the total number of trajectories and T is the length of each trajectory. The neural network parameterized policies depend on either the entire history h_t or the most recent observation o_t. Let us denote the policy parameters with θ, BC trains policies to maximize the expected log likelihood of the dataset trajectories:

$$\max_{\theta} \mathcal{L}(\theta) = \mathbb{E}_{o_{1:t}, a_{1:t} \sim \mathcal{D}} \left[\sum_{t=1}^{T} \log \pi_{\theta}(a_t | h_t) \right] \tag{3}$$

BC is simple to implement and more computationally efficient than comparative data-driven machine learning methods like reinforcement learning and online imitation learning. BC also does not require a high fidelity traffic simulation environment for training, which is necessary for reinforcement learning and online imitation learning. In contrast to rule-based policies, BC policies are more flexible and can express a much larger class of behaviors.

However, BC as a representative offline learning method has known disadvantages of being sensitive to the quantity and quality of training data and input features. The covariate-shift between the training dataset and the testing environment and neural network models' difficulty of extrapolating learned mechanisms to unseen inputs often cause BC models to overfit to the training dataset while producing poor control behavior during closed-loop testing (defined in section 2.8.2) [20, 36]. Furthermore, several studies have found that BC can be highly sensitive to input features [16, 19, 37]. Specifically, when a driver's previous control actions are used as input features to the trained policy, it is likely that the policy merely repeats those controls actions in closed-loop testing. This has been interpreted as a form of learning spurious correlations or causal confusion in machine learning, since driver controls at adjacent time steps are usually so similar that predicting previous controls can quickly minimize training error [37]. Because BC does not impose any structure on the policy, examining the failure modes of BC models is as challenging as examining any other black-box neural network models.

Despite these shortcomings, BC, or variations of it, is a widely studied approach in developing automated vehicle algorithms and building simulated agents and environments for training them [24, 25]. It can produce high quality simulated behavior in practice when the training dataset is large and diverse, appropriate features are selected, and the neural network model is large and expressive enough [16, 22, 23] and thus it represents a valid data-driven modeling benchmark.

2.3 Active Inference Driving Agent

An active inference agent is defined by its internal generative model, which we implemented as a Partially Observable Markov Decision Process (POMDP). A POMDP describes a dynamic process in which the state of the environment $s_t \in \mathcal{S}$ evolves with driver actions, $a_t \in \mathcal{A}$, according to a probability distribution, $P(s_{t+1}|s_t, a_t)$, and generates observation signal, $o_{t+1} \in \mathcal{O}$, according to, $P(o_{t+1}|s_{t+1})$. In this work, we assume the observation signals are multivariate continuous variables and the states are discrete to represent probabilistic categorizations of the observation space (i.e., categorical perception [38]). At every time step, the active inference agent first makes inference about the hidden state of the environment upon receiving observations using Bayes' rule:

$$b_t(s_t) = \frac{P(o_t|s_t)P(s_t|b_{t-1}, a_{t-1})}{\sum_{s_t} P(o_t|s_t)P(s_t|b_{t-1}, a_{t-1})} \tag{4}$$

where $b_t(s_t) = P(s_t|h_t)$ denotes the agent's belief about the environment state given the observation-action history h_t and $P(s_t|b_{t-1}, a_{t-1}) = \sum_{s_{t-1}} P(s_t|s_{t-1}, a_{t-1})b(s_{t-1})$ is the prior predictive distribution based on the previous belief.

The active inference agent then selects control actions to minimize a criterion known as the (cumulative) expected free energy (EFE) [39]:

$$\mathcal{G}^*(b_t) = \min_{\pi} \mathbb{E}\left[\sum_{t}^{t+H} EFE(b_t, a_t) + \log \pi(a_t|b_t)\right] \tag{5}$$

where $H \leq \infty$ is a finite planning horizon. The EFE is defined as:

$$EFE(b_t, a_t) \triangleq \mathbb{E}[D_{KL}\left(b_{t+1}||\tilde{P}\right)] + \mathbb{E}[\mathcal{H}(o_{t+1})] \tag{6}$$

where $\tilde{P} := \tilde{P}(s_{t+1})$ defines the agent's preferred state distribution, $D_{KL}(\cdot||\cdot)$ denotes the Kullback-Leibler divergence — measuring the discrepancy between the current belief and the preferred state distribution — and $\mathcal{H}(\cdot)$ denotes Shannon entropy — measuring uncertainty about observations. These terms represent goal-seeking and information-seeking (epistemic) behavior respectively [40]. The first expectation in the EFE is taken with respect to:

$$P(o_{t+1}|b_t, a_t) = \sum_{s_{t+1}} P(o_{t+1}|s_{t+1})P(s_{t+1}|b_t, a_t) \tag{7}$$

and the second expectation in the EFE is taken with respect to $P(s_{t+1}|b_t, a_t)$.

Let $\mathcal{G}^*(b_t, a_t)$ be defined as:

$$\mathcal{G}^*(b_t, a_t) := EFE(b_t, a_t) + \log \pi(a_t|b_t) + \int P(o_{t+1}|b_t, a_t)\mathcal{G}^*(b_{t+1})d_{o_{t+1}} \tag{8}$$

Then the optimal policy has a closed-form expression [41]:

$$\pi(a_t|b_t) = \frac{e^{-\mathcal{G}^*(b_t, a_t)}}{\sum_{\tilde{a} \in \mathcal{A}} e^{-\mathcal{G}^*(b_t, \tilde{a})}} \tag{9}$$

Active inference has two important differences from the traditional notion of POMDP in operations research and reinforcement learning. First, both the generative model and the control objective are internal to the agent, meaning they can differ in substantial ways from the true environment generative model or a canonical notion of desired behavior, e.g., a "good driver" should always be centered in the lane. This has important implications as many human driving behaviors can be explained as inference in subjective or sub-optimal models [42]. Second, active inference makes an explicit distinction between pragmatic and epistemic behavior in its policy objective according to the first and second terms in (6). This distinction supports adaptive behavior in unknown and uncertain environments [40, 43], e.g., traffic environments.

2.4 Dataset

We performed our analysis of the IDM, BC, and AIDA using the INTERACTION dataset [44], a publicly available driving dataset recorded using drones on fixed road segments in the USA, Germany, and China. The dataset provides a lanelet2 format map [45] and a set of time-indexed trajectories of the positions, velocities, and headings of each vehicle in the scene in the map's coordinate system at a sampling frequency of 10 Hz, and the vehicle's length and width for each road segment. The dataset contains a variety of traffic behaviors, including car following, free-flow traffic, and merges.

Due to our emphasis on car following behavior, we selected a subset of the data to include car following data from a two-way, seven-lane highway segment in China with a total distance of 175 m. We focused on vehicles in the middle two west-bound lanes shown in Figure 2. We further filtered the remaining vehicles according to two criteria: 1) there was

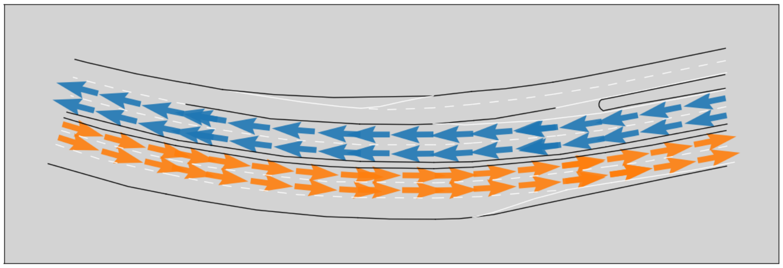


FIGURE 2: Map of the roadway included in data collection. The westbound lanes (blue) were used for training and the eastbound lanes (orange) were used for testing.

a lead vehicle with a maximum distance headway of 60 m, and 2) the ego vehicle was not performing a merge or lane change. We identified merging and lane change behavior using an automated logistic regression-based approach and validated the classifications with a manual review of a subset of trajectories. We also removed all trajectories with length shorter than 5 seconds, leaving a total of 1,254 trajectories with an average length of 14 seconds.

2.4.1 Feature Computation

The input features to the IDM are defined in (1) and (2). For BC and the AIDA, we used d and Δv but excluded v to prevent learning spurious correlations to ego speed or acceleration from past time steps reported in prior studies [16, 22, 37]. Furthermore, we included an additional feature τ^{-1} in BC and AIDA defined as the rate of change of the visual angle of the lead vehicle from the ego driver's seat position divided by the angle itself. τ^{-1} can be considered as a perceptual-control analog of inverse time-to-collision, a feature commonly used in driver modeling [19, 46, 47], with the difference of incorporating the width of the lead vehicle into feature computation and using quantities that can actually be observed by the driver. This is consistent with recent findings on the impact of optical expansion of the lead vehicle's image on driver longitudinal control behavior [48]. Furthermore, the inclusion of this feature puts the information contained in the inputs to BC and the AIDA on a similar level to the IDM, as the IDM implicitly accounts for time-to-collision in its desired distance headway computation in (2).

We computed all features in the Frenet frame (i.e., lane-centric coordinates [49]), by first transforming vehicle positions, velocities, and headings using the current lane center line as the reference path and then computing the features from the transformed positions and velocities. We obtained the drivers' instantaneous longitudinal control inputs (i.e., accelerations) from the dataset by differentiating the Frenet frame longitudinal velocities. For BC and the AIDA, we discretized the continuous control inputs into discrete actions using a Gaussian mixture model of 15 Gaussian components with mean and variance parameters chosen using the Bayesian Information Criteria [50].

2.5 Model implementation

In this section, we describe our approach for parameterizing the IDM, BC, and the AIDA.

IDM. Following [51], we parameterized the IDM by treating the IDM policy as a conditional Gaussian distribution: $\pi(a_t|o_t = \{v_t, \Delta v_t, d_t\}) = \mathcal{N}(a_t|\mu_t, \sigma^2)$ with mean action μ_t and variance σ^2. The mean action μ_t is computed from the IDM rule defined in (1) and (2) by making the desired speed $\tilde{v}$, minimum distance headway d_0, desired headway time τ, maximum acceleration rate a_{max} and maximum deceleration rate b adjustable parameters. The action variance σ^2 is assumed to be independent of the input features and also estimated from data.

BC. We implemented the BC model with two types of neural network policies: standard Multi-Layer Perceptron (MLP) networks and recurrent neural networks (RNN) following [19]. The MLP network takes as input the observation vector (normalized by training set statistics) and outputs a probability distribution over the discrete control actions. The RNN addresses the possibility that driver behavior may be influenced by the full observation history rather than just the most recent observation. For the RNN, we combined a Gated Recurrent Unit (GRU) network [52] with a MLP network, where the GRU network compresses the observation history into a fixed length vector, which is then transformed into the action distribution by the MLP network.

AIDA. We modeled the discrete state transition probabilities $P(s_{t+1}|s_t, a_t)$ and the desired state distribution $\tilde{P}(s_t)$ of the AIDA using categorical distributions parameterized by their logits. We parameterized the observation distributions $P(o_t|s_t)$ using Normalizing Flow, a flexible class of neural network-based density estimator [53, 54]. This provides the AIDA with adequate flexibility in modeling complex and nonlinear observation sequences and associating observed actions with agent beliefs. Normalizing Flow uses invertible neural networks to transform simple distributions, e.g., Gaussian distributions, into complex and correlated distributions while maintaining the tractability of likelihood evaluation and sampling. In this work, we used a single, shared Inverse Autoregressive Flow [55] to transform a set of conditional Gaussians with mean vector $\mu(s_t)$ and covariance matrix $\Sigma(s_t)$. We modeled a distribution over the agent's planning horizon using a Poisson rate parameter and used the QMDP method [56, 57] as a closed-form approximation of the cumulative expected free energy in (8). We approximately computed the entropy of the state-conditioned observation distributions required in the EFE calculation using the entropy of the Normalizing Flow base distributions. In subsequent sections, we refer to the transition and observation parameters with θ_1, the desired state distribution and planning horizon parameters with θ_2, and the combined parameters with $\theta = \{\theta_1, \theta_2\}$.

We provide additional implementation details in Appendix A. Our software implementation is publicly available at [58].

2.6 Parameter Estimation

We estimated the parameters of the IDM, BC, and AIDA by maximizing the expected log likelihood of driver control inputs from the dataset under the corresponding control policy, i.e., (3). This procedure for the AIDA differs slightly from the IDM and BC by requiring a nested step. Between each parameter update in the nested procedure, we first computed the sequence of beliefs given the observation-action history using (4) and the optimal belief-action policy using (9). We then evaluated the log likelihood as a function of the computed beliefs:

$$\max_{\theta} \mathbb{E}_{o_{1:T}, a_{1:T} \sim \mathcal{D}} \left[\sum_{t=1}^{T-1} \log \pi_{\theta}(a_t | b_{t,\theta_1}) \right] \quad \text{s.t.} \quad \pi_{\theta}(a_t | b_t) = \frac{e^{-\mathcal{G}^*_{t,\theta}(b_t, a_t)}}{\sum_{\tilde{a}_t \in \mathcal{A}} e^{-\mathcal{G}^*_{t,\theta}(b_t, \tilde{a}_t)}} \tag{10}$$

While (10) allows us to learn task-relevant beliefs in active inference agents as it depends on both θ_1 and θ_2, the parameters are fundamentally unidentifiable since there are potentially infinite sets of θ with the same likelihood [59–61]. This is because, for example, the estimation algorithm cannot differentiate between drivers who desire small distance headway and drivers who believe the distance headway will increase to desired levels in subsequent time steps. A possible consequence of this is learning an environment model that deviates significantly from the reality, which leads to a large number of crash or inaction as a result of the agent not being able to recognize the actual environment state [62].

In order to constrain the hypothesis space and avoid configurations of θ that are incompatible with real-world constraints, we designed a data-driven prior distribution $P(\theta)$ encoding likely configurations of θ. Specifically, the prior is defined as $P(\theta) = P(\theta_1)P(\theta_2|\theta_1)$, where:

$$P(\theta_1) \propto \exp\left(\lambda \mathbb{E}_{o_{1:T}, a_{1:T} \sim \mathcal{D}} \left[\sum_{t=1}^{T} \log P_{\theta_1}(o_t | h_{t-1}, a_{t-1}) \right] \right) \tag{11}$$

with hyperparameter λ controlling how much the prior distribution prefers model accuracy, measured by expected log likelihood of observations. We let $P(\theta_2|\theta_1)$ be a uniform distribution. In our experiments, we only compute the Maximum A posteriori (MAP) estimate of the Bayesian model by converting the prior into the following loss function added to the objective in (10):

$$\mathcal{L}(\theta_1) = \lambda \mathbb{E}_{o_{1:T}, a_{1:T} \sim \mathcal{D}} \left[\sum_{t=1}^{T} \log P_{\theta_1}(o_t | h_{t-1}, a_{t-1}) \right] \tag{12}$$

To prevent learning unreasonably large observation variance as a result of the observation entropy term in (6), another symptom previously reported in [62], we applied a penalty on the l^2 norm of the observation covariance parameters.

Using these prior loss functions, the AIDA MAP estimate can be written as:

$$\theta^{\text{MAP}} = \arg\max_{\theta} \sum_{t=1}^{T} \mathbb{E}_{o_{1:T}, a_{1:T} \sim \mathcal{D}} \left[\log \pi_{\theta}(a_t | b_{t,\theta_1}) + \lambda_1 \log P_{\theta_1}(o_t | h_{t-1}, a_{t-1}) \right] + \lambda_2 \sum_{s} ||\Sigma_{\theta_1}(s)||^2 \tag{13}$$

2.7 Model selection

We trained each model with 15 random seeds controlling model parameter initialization and dataset mini-batch iteration orders. To select the hyperparameters for the AIDA, we first set $\lambda_2 = 0.1$ since it's sufficient to mitigate overly large covariances. We then trained the model for $\lambda_1 = [0.2, 1, 4]$ and selected $\lambda_1 = 1$ as it best trades off environment model accuracy and agent behavior predictive performance (with criteria described in the next section).

2.8 Model Evaluation and Comparison

We evaluated and compared our models' ability to generate behavior similar to the human drivers in the dataset using both open-loop offline predictions and closed-loop online simulations. In both cases, we evaluated the models on two different held-out testing datasets. The first dataset includes vehicles from the same lanes as the training dataset. This dataset tests whether the models can generalize to unseen vehicles in the same traffic condition. We obtained this dataset by dividing all selected trajectories in the westbound lanes using a 7-3 train-test ratio. The second dataset includes vehicles from the top two eastbound lanes in Figure 2. This dataset tests whether the models can generalize to unseen vehicles in novel traffic conditions, since the traffic in the eastbound lanes have on average higher speed and less density. We refer to these two datasets as *same-lane* and *new-lane*, respectively. We randomly selected 100 trajectories with a minimum length of 10 seconds from the same-lane dataset and 75 trajectories with a minimum length of 5 seconds from the new-lane dataset for testing.

2.8.1 Offline Evaluation

The goal of the offline evaluation was to assess each model's ability to predict a driver's next action based on the observation-action history recorded in the held-out testing dataset. This task evaluates the models' ability to be used as a short-horizon predictor of other vehicles' behavior in an on-board trajectory planner [63]. We measured a model's predictive accuracy using Mean Absolute Error (MAE) of the predicted control inputs (unit=m/s^2) on the held-out testing datasets. For the IDM, we calculated the predicted control inputs by sampling from the Gaussian policy. For BC and the AIDA, we first sampled a discrete action from the action distribution predicted by the models and then sampled from the corresponding Gaussian component in the Gaussian mixture model used to perform action discretization.

2.8.2 Online Evaluation

Rather than predicting instantaneous actions, the goal of the online evaluation was to assess the models' ability to generate trajectories similar to human drivers such that they can be used as simulated agents in automated vehicle training and testing environments [24]. This is fundamentally different from offline predictions because the models need to choose actions based on observation-action history generated by its own actions rather than those stored in the fixed, offline dataset. This can introduce significant covariate shift [20] sometimes resulting in situations outside of the model's training data, which can lead to poor action selection.

We built a single-agent simulator where the ego vehicle's longitudinal acceleration is controlled by the trained models and its lateral acceleration is controlled by a feedback controller for lane-centering. The lead vehicle simply plays back the trajectory recorded in the dataset. Other vehicles do not have any effect on the ego vehicle, given our observation space does not contain other vehicle related features.

Following [25], We measured the similarity between the generated trajectories and the true trajectories using the following metrics:

1. Average deviation error (ADE; unit=m): deviation of the Frenet Frame longitudinal

position from the dataset averaged over all time steps in the trajectory.

2. Lead vehicle collision rate (LVCR; unit=%): percentage of testing trajectories containing collision events with the lead vehicle. A collision is defined as an overlap between the ego and lead vehicles' bounding boxes.

2.8.3 Statistical Evaluation

Following the recommendations in [64, 65] for evaluating learned control policies, we represented the central tendency of a model's offline prediction and online control performance using the interquartile mean (IQM) of the offline MAEs and online ADEs. Note however for collision rate, we compute the regular mean instead of IQM to account for the collision rate lower bound of 0. The IQMs are computed by 1) ranking all tested trajectories by their respective performance metrics and 2) computing the mean of the performance metrics ranked in the middle 50%. To compare the central performance difference between the AIDA and baseline models, we performed two-sided Welch's t-tests with 5 percent rejection level on the MAE-IQM and ADE-IQM values computed from different random seeds with the assumption that the performance distributions between two models may have different variances [64, 65].

3 RESULTS AND DISCUSSION

3.1 Offline Performance Comparison

Figure 3 shows the offline evaluation results for each model with the model type on the x-axis and the IQMs of acceleration prediction MAEs averaged across the testing dataset on the y-axis. The color of the points in the figure represents the testing condition and each point corresponds to a random seed's result. The points are randomly distributed around each x-axis label for clarity. Dispersion on the y-axis indicates sensitivity in the model to initial training conditions. The plot illustrates that the AIDA had the lowest MAE-IQM in the same-lane tests, followed by BC-RNN, BC-MLP, and IDM. The corresponding pairwise Welch's t-test results in Table 1 show that these differences are significant. In the new-lane tests, both the AIDA and neural network BC models significantly outperformed IDM. The AIDA performance has higher variance than BC models, however the difference in the central tendency was not significant. These results show that in the current car following setting, the AIDA and BC generalized better to the new-lane scenario than the IDM, mostly likely due to the IDM rules being unable to adapt to different traffic speed and density than the training dataset. The stronger predictive performance in the AIDA and BC-RNN in the same-lane data can be attributed to the fact that driver acceleration actions depend on the full history of past observations rather than just the most recent observation, which can be modeled by the recurrent structure of the AIDA and BC-RNN. The figure also shows that for the same-lane tests, the AIDA had more variance across the random seeds compared to other models, suggesting that it is the most sensitive to local optima in the training process.

3.2 Online Performance Comparison

Figure 4 shows the IQM of each model's ADEs from data set trajectories in the online evaluations using the same format as the offline evaluation results. In the same-lane testing condition, all models had an ADE-IQM values between 1.8 m and 2.8 m, which is less than the length of a standard sedan (≈ 4.8 m; [66]). Among all models, BC-MLP achieved the

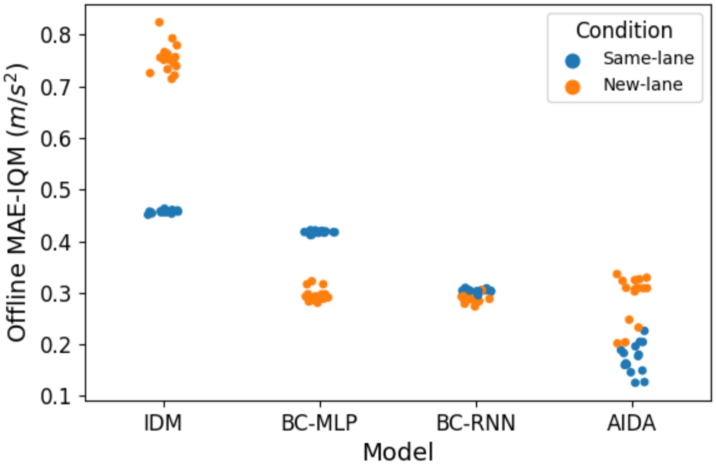


FIGURE 3: Offline evaluation MAE-IQM. Each point corresponds to a random seed used to initialize model training and its color corresponds to the testing condition of either same-lane or new-lane.

TABLE 1: Two-sided Welch's t-test results of offline MAE-IQM against baseline models. Asterisks indicate statistical significance with $\alpha = 0.05$.

Baseline	Comparison	t(df=14)	p-value
IDM	same-lane	t=37.58	p<0.001*
BC-MLP	same-lane	t=32.38	p<0.001*
BC-RNN	same-lane	t=17.31	p<0.001*
IDM	new-lane	t=33.21	p<0.001*
BC-MLP	new-lane	t=0.35	p=0.73
BC-RNN	new-lane	t=-0.12	p=0.90

lowest ADE values for both the same-lane and new-lane conditions, followed by the AIDA, IDM, and BC-RNN. Furthermore, both the AIDA and BC models achieved lower ADE-IQM in the new lane settings compared to the same-lane setting, however the IDM achieved higher ADE-IQM in the new-lane setting. The Welch's t-test results in Table 2 show that AIDA's online test performances are significantly different from all baseline models in both the same-lane and new-lane settings ($P \leq 0.01$). These findings confirm that the AIDA and BC models generalized better to the new-lane setting than the IDM and suggest that the AIDA's average online trajectory-matching ability is significantly better than IDM and BC-RNN, although BC-MLP is significantly better than the AIDA.

Figure 5 shows the lead vehicle collision rates for each random seed and model using the same format as Figure 4. The figure illustrates that in the same-lane condition, the random seeds for BC-MLP, BC-RNN, and the AIDA had more collisions than the IDM (0% collision rate across all seeds). In particular, BC-RNN and the AIDA had substantial differences across random seeds compared to the other models. However, the minimum collision rates for BC-MLP, BC-RNN, and the AIDA were consistent (less than or equal to 1%). In the new-lane condition, the collision rate was 0% for all four models. The higher collision rates in the same-lane data are likely due to the traffic density and complexity, which were higher in the

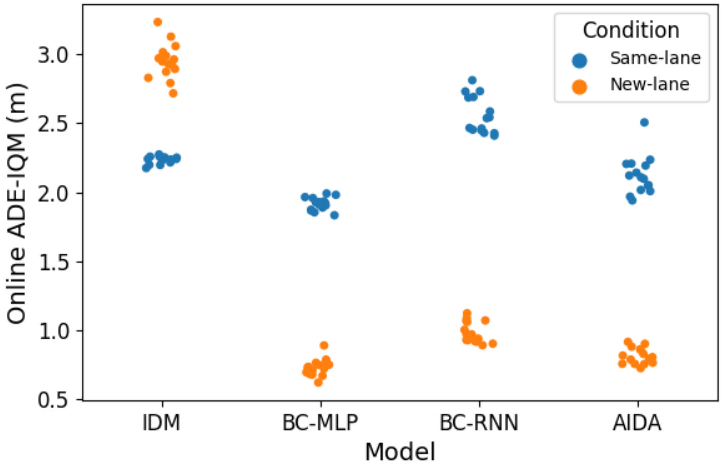


FIGURE 4: Online evaluation ADE-IQM. Each point corresponds to a random seed used to initialize model training and its color corresponds to the testing condition of either same-lane or new-lane.

TABLE 2: Two-sided Welch's t-test results of online ADE-IQM against baseline models. Asterisks indicate statistical significance with $\alpha = 0.05$.

Baseline	Comparison	t(df=14)	p-value
IDM	same-lane	t=3.05	p<0.01*
BC-MLP	same-lane	t=-5.46	p<0.001*
BC-RNN	same-lane	t=8.73	p<0.001*
IDM	new-lane	t=58.18	p<0.001*
BC-MLP	new-lane	t=-3.77	p<0.001*
BC-RNN	new-lane	t = 6.87	p<0.001*

same-lane condition compared to the new-lane condition.

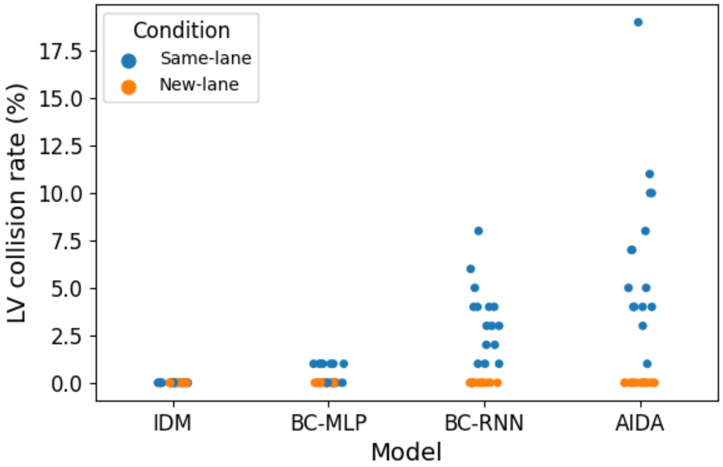


FIGURE 5: Lead vehicle collision rate in online evaluation. Each point corresponds to a random seed used to initialize model training and its color corresponds to the testing condition of either same-lane or new-lane.

3.3 AIDA Interpretability Analysis

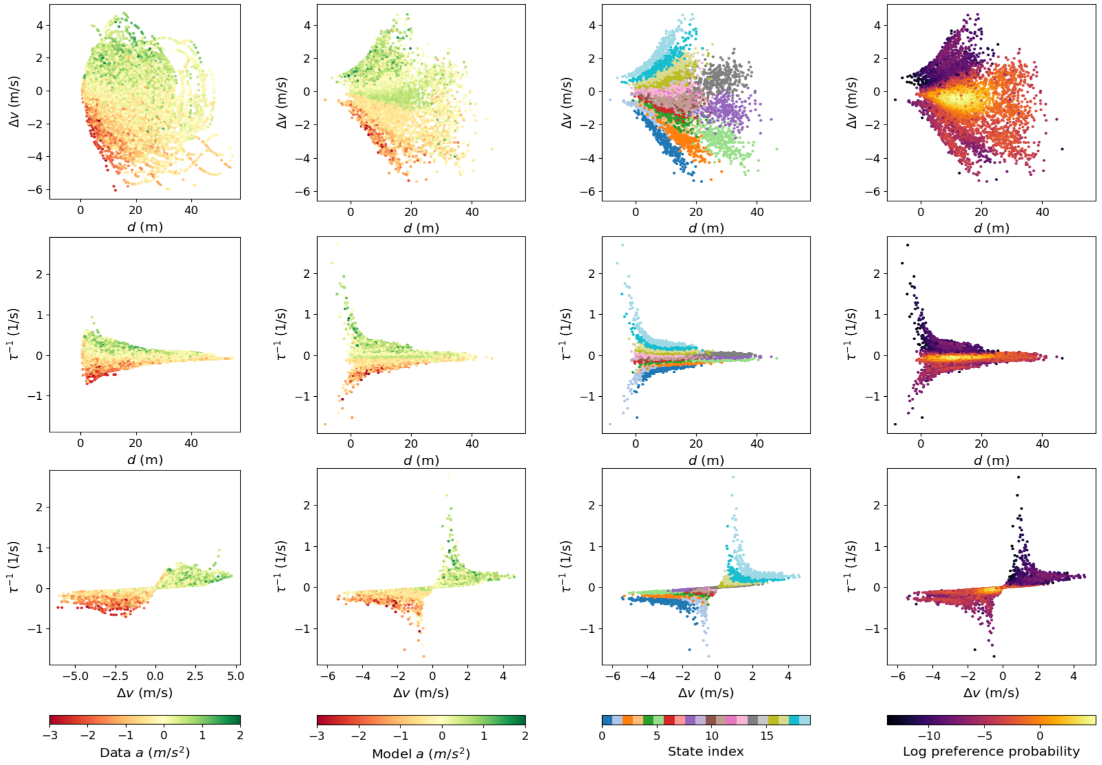
The previous sections suggest that the AIDA can capture driver car following behavior significantly better than the IDM and comparably to data-driven BC models. However, the findings have yet addressed the interpretability of the AIDA. While there is no established metric for model interpretability, Räukur et. al. [8] recommend assessments based on the easiness of comprehending the connection between model input and output and tracing model predictive errors to internal model dynamics. Given that the AIDA's decisions are emitted from a two-step process, i.e., (1) forming beliefs about the environment and (2) selecting control actions that minimize free energy, the model's success at the car following task depends on the two sub-processes both independently and jointly. We examined the AIDA's learned input-output mechanism by visualizing its independent components (i.e., the observation, transition, and preference distributions) and verified them against expectations guided by driving theory [43, 67, 68]. We then examined the joint belief-action process by replaying the AIDA beliefs and diagnosing its predictions of recorded human drivers in the offline setting and its own decisions in the online setting.

3.3.1 Independent Component Interpretability

Initial insights into the model input and output connections can be gained by visualizing the AIDA components, specifically its policy (Figure 6b), observation distribution (shown in Figure 6c), and preference distribution (Figure 6d). These figures show 200 random samples from each state of the AIDA's state-conditioned observation distribution, $P(o|s)$, plotted on each pair of observation modalities. Color is used to highlight relevant quantities of interest. We further used samples drawn from the INTERACTION dataset, plotted in Figure 6a and colored by the recorded accelerations, to facilitate interpreting the the AIDA samples.

Figure 6b illustrates the observation samples by the model's chosen control actions. The top chart shows the samples using distance headway (d; x-axis) by relative velocity to the lead vehicle (Δv; y-axis), the middle chart shows distance headway by τ^{-1}, and the bottom chart shows relative velocity by τ^{-1}. The shape of the sampled points matches the contour of the empirical dataset (Figure 6a), particularly in the middle and bottom visualizations, which suggests that the model's learned observation model aligns with the recorded observations in the dataset. Darker green and red colors correspond to larger acceleration and deceleration magnitudes, respectively, and light yellow color corresponds to near zero control inputs. The color gradient at different regions in Figure 6b is consistent with that of the empirical dataset shown in Figure 6a. This shows that the model learned a similar observation to action mapping as the empirical dataset. The mapping can be interpreted as the tendency to choose negative accelerations when the relative speed and τ^{-1} are negative and the distance headway is small, and positive accelerations in the opposite case. Furthermore, the sensitivity of the red and green color gradients with respect to distance headway shows that the model tends to accelerate whenever there is positive relative velocity, regardless of the distance headway. However, it tends to input smaller deceleration at large distance headway for the same level of relative speed.

Figure 6c shows the observation samples colored by their associated discrete states. The juxtaposition of color clusters in the top panel shows that the AIDA learned to categorize



(a) (b) (c) (d)

FIGURE 6: Visualizations of the dataset and AIDA model components. In panel (a), we plotted observations sampled from the dataset. In panels (b), (c), and (d) we sampled 200 points from the AIDA's state conditioned observation distributions and plotted the sampled points for each pair of observation feature combinations. The points in each panel are colored by: (a) accelerations from the dataset, (b) the AIDA's predicted accelerations upon observing the sampled signals from a uniform prior belief, (c) state assignments (d) log probabilities of the preference distribution.

observations by relative speed and distance headway and its categorization for relative speed is more fine-grained at small distance headways and spans a larger range of values. The middle and bottom panels show that its categorization of relative speed is highly correlated with τ^{-1} as the ordering of colors along the y-axis is approximately the same as in the top panel. The middle and bottom panels show that the AIDA's categorization of high τ_1 magnitude states (blue and cyan clusters) have larger span than that of low τ^{-1} magnitude states. These patterns further establish that the AIDA has learned a representation of the environment consistent with the dataset. At the same time, it can be interpreted as a form of satisficing in that the model represents low urgency large distance headway states with less granularity [69].

Figure 6d shows the observation samples by the log of its preference probability, $\tilde{P}(o) = \sum_s \tilde{P}(s)P(o|s)$, where higher preference probability (i.e., desirability) corresponds to brighter colors (e.g., yellow) and lower desirability corresponds to darker colors (e.g., purple). The figure shows that the highest preference probability corresponds to observations of zero τ^{-1}, zero relative velocity, and a distance headway of 18 m (see the center region of the middle chart, and the yellow circle at the left-center of the top chart). This aligns with the task-difficulty homeostasis hypothesis that drivers prefer states in which the crash risk is manageable [67] and not increasing. It is also consistent with the observed driver behavior in Figure 6a where drivers tend to maintain low accelerations (light yellow points) within the same regions.

Overall, these results show a clear mapping between the AIDA's perceptual (Figure 6c) and control (Figure 6d and 6b) behavior that is both consistent with the observed data and straightforwardly illustrated using samples from the fitted model distributions. This mapping facilitates predictions of the AIDA's reaction to observations without querying the model, which is an important dimension of interpretability in real world model verification [8].

3.3.2 *Joint Model Interpretability*

While the previous analysis illustrates the interpretability of individual model components, the interaction between components introduces additional challenges for overall model interpretability. To address this, we analyzed two same-lane scenarios where the AIDA made sub-optimal decisions in the model testing phase — one from the offline evaluations where the AIDA's predictions had the largest MAE and one from the online evaluations where the AIDA generated a rear-end collision with the lead vehicle. We first visualized the AIDA's beliefs (computed by (4)) and policies (computed by (9)) as the model generated actions and then used those visualizations to demonstrate how the transparent input-output mechanism in the AIDA can be used to mitigate the sub-optimal decisions.

The chosen offline evaluation trajectory is visualized in Figure 7. The left column charts show the data of the three observation features over time. The right column charts show the time-varying ground truth action probabilities over time (top), action probabilities predicted by the AIDA over time (middle), and environment state probabilities $P(s|h)$ inferred by the AIDA over time (bottom). In the right-middle and right-bottom charts, the action and belief state indices are sorted by the mean acceleration and τ^{-1} value of each state to facilitate alignment with the left and top-right charts. We labeled the actions by the corresponding means but not the belief states because they represent multi-dimensional observation categorizations (see Figure 6c). The bottom-right chart shows that the inferred belief patterns closely followed the observed relative speed and τ^{-1} in the left-middle and left-bottom charts with high precision, i.e., close to probability of 1. The predicted action probabilities in the right-middle chart followed the trend of the ground truth actions, however, they exhibited substantially higher uncertainty at most time steps and multi-modality at $t = 1$ s and $t = 12$ s, where one of the predicted modes coincided with the true actions. Given the inferred beliefs were precise, uncertain and multi-model actions were likely caused by inter-driver variability in the dataset, where drivers experienced similar belief states but selected different actions. Alternatively, this uncertainty may be caused by drivers having highly different beliefs after experiencing similar observations, where a simple policy would be sufficient to predict their actions. In either case, the error in AIDA predictions can be attributed to inconsistency between the belief trajectories and action predictions.

The chosen online evaluation trajectory which resulted in a rear-end collision with the lead vehicle is shown in Figure 8 plotted using the same format as Figure 7. The duration of the crash event is highlighted by the red square in the bottom-left chart, where the sign

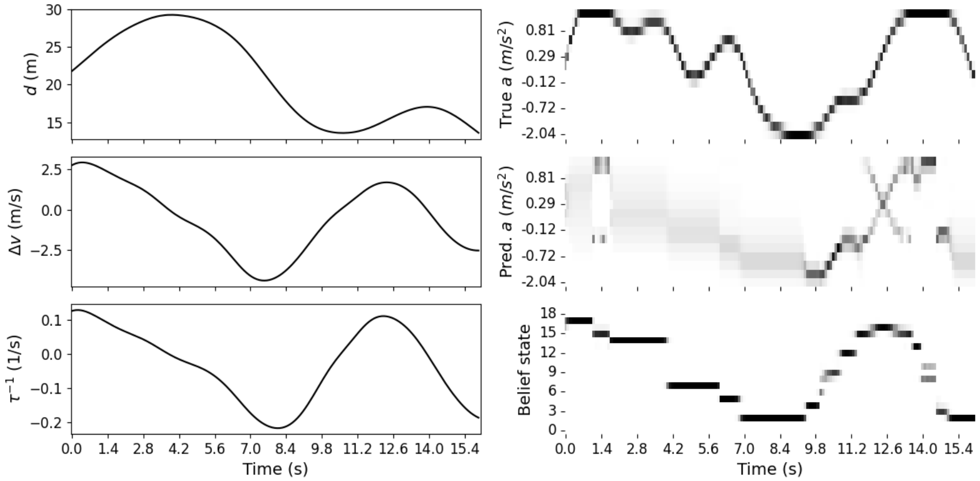


FIGURE 7: Visualizations of a same-lane offline evaluation trajectory where the AIDA had the highest prediction MAE. The charts in the left column show distance headway, relative speed, and τ^{-1} signals observed by the model over time. The binary heat maps in the right column show the ground truth action probabilities (top), action probabilities predicted by the AIDA (middle), and the corresponding belief states (bottom) over time (x-axis), where darker colors correspond to higher probabilities. The belief state and action indices are sorted by the mean τ^{-1} and acceleration value of each state, respectively.

of τ^{-1} values instantly inverted when overlapping bounding boxes between the ego and lead vehicle first occurred and eventually ended. The AIDA initially made the correct and precise decision of braking, however, its predictions for high magnitude actions became substantially less precise prior to the collision ($t > 1$ s; see right middle chart). This led to the model failing to stop fully before colliding with the lead vehicle. The belief pattern shows that the AIDA tracked the initial decreasing values of relative speed and τ^{-1} but did not further respond to increasing magnitude of τ^{-1} 3 seconds prior to the crash (starting at $t = 1.6$ s). These findings show that the model exhibited the correct behavior of being "shocked" by out-of-sample near-crash observations, however, the learned categorical belief representation was not able to extrapolate beyond the data from the crash-free INTERACTION dataset.

The analysis of the near-crash AIDA beliefs suggests that editing the AIDA's learned environment dynamics model (i.e., the transition and observation distributions) to properly recognize near-crash observation signals can likely avoid the current crash. To demonstrate the utility of being able to make precise model-editing decision based on the interpretability analysis, we tested a modification of the AIDA by replacing its learned dynamics model with a physics-based dynamics model assuming constant lead vehicle velocity in the model predictions. Although the physics-based dynamics model does not capture the stochasticity in the lead vehicle behavior, it is sufficient for mitigating the current crash given its ability to accurately predict near-crash observations. We evaluated this new model in the same online testing scenarios as the AIDA, where the control actions were generated from a model-predictive controller (MPC [70]) using the AIDA's preference distribution as the reward function (for detailed implementation seed Appendix A.3). The AIDA-MPC mitigated all crashes when deployed in the same scenarios as the AIDA as our analysis predicted. How-

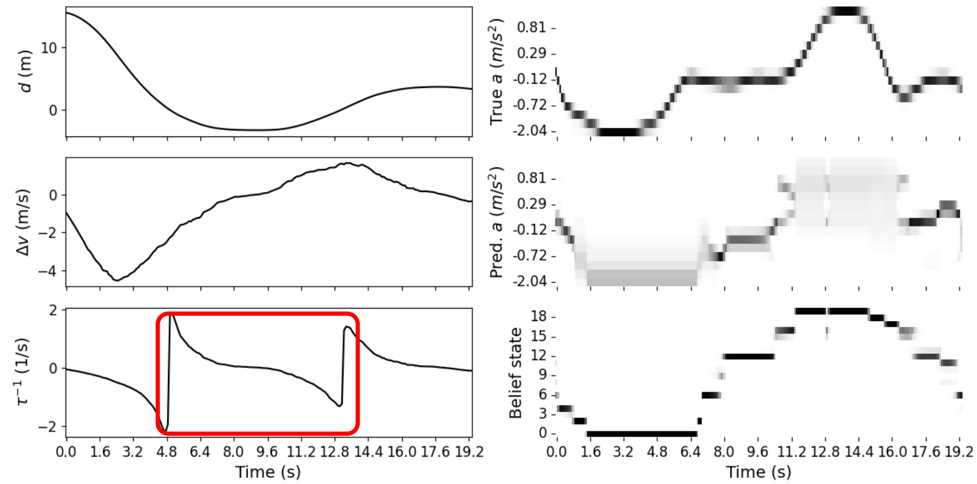


FIGURE 8: Visualizations of a same-lane online evaluation trajectory where the AIDA generated a rear-end collision with the lead vehicle. This figure shares the same format as Figure 7. The red square in the bottom-left chart represents the duration of the rear-end crash event where the vehicle controlled by the AIDA had overlapping bounding box with the lead vehicle.

ever, it generated substantially more high-ADE trajectories than the AIDA, most likely due to the lack of representation of lead vehicle stochasticity.

The analyses in this section show that the decision making structure in the AIDA enables modelers to reason about the training dataset's effect on the learned model behavior. To the best of our knowledge, this analysis is not possible with neural network BC models using existing interpretability tools. We also showed how this understanding can be used to edit parts of the model to achieve desired safety criteria.

4 GENERAL DISCUSSION

In this article, we introduced and evaluated a novel active inference model of driver car following behavior (AIDA). The proposed AIDA significantly outperformed the IDM and neural network BC models in offline predictions in the same-lane condition and outperformed the IDM while performing similarly to BC models in the new-lane condition. Additionally, the AIDA achieved significantly lower average deviation error than the IDM and BC-RNN in the online control settings. However, the results showed that the AIDA was sensitive to initial training conditions, which resulted in higher rates of lead vehicle collisions in the same-lane condition compared to the IDM and BC-MLP. While BC had comparable or better performance than the AIDA in action prediction and control, the AIDA is substantially more interpretable than BC models. In contrast to approximate explanatory methods for BC neural networks, we showed that the AIDA's decision making process can be directly accessed by sampling and visualizing the AIDA distributions. Further, we illustrated how the AIDA's joint belief and action trajectories could be used to understand model errors and correct them. This level of understanding and diagnostic analysis is central to real world model inspection and verification which are essential components of interpretability [8, 71].

These results partially confirm our hypothesis that balancing the relative strengths of rule-based and data-driven models, specifically using the active inference framework, results in better predictions of driver behavior and more nuanced understanding of driver cognitive dynamics during car following. In contrast to fixed rule-based models like the IDM, the AIDA can incorporate additional "rules" in its state and policy priors while maintaining the flexibility provided by its probabilistic representation. In contrast to purely data-driven models, learning in the AIDA is constrained by its probability distributions and structure. This balance preserves interpretability but still allows the model to be flexible to new data. Our findings here suggest that this flexibilty comes at a cost of sensitivity to local optima in the training process as evidenced by the collision rates across random seeds in online evaluations. Further, our findings suggest that the AIDA, like other data-driven approaches, may be limited by the scope of the data used in training (e.g., the crash limitation illustrated in Figure 8).

Our findings here extend prior applications of active inference theory in driving and driver models and illustrate the value of rule-based modeling. Engström and colleagues [43] presented active inference as a general theory of driving behavior with qualitative illustrations, highlighting the need to separate pragmatic (risk) and epistemic (uncertainty) behavior and relaxing the requirement of a strictly accurate environment model among human drivers. Portions of this theory have been enacted in other driver models including [17, 31, 72]. The model in [17] includes the concept of balancing rule-based and data-driven models, but the focus is primarily on physical concepts rather than psychological concepts in the AIDA. The model presented by Pekkanen et al. [72] includes an attention mechanism driven by the uncertainty of desired actions. The desired actions were computed using the IDM and action uncertainty was obtained by propagating state uncertainty computed from a Bayesian filter. The most notable differences between Pekkanen et al.'s model and the AIDA is that their model assumes an accurate environment model and uses the IDM to generate behavior. Our results show that an integrated perception-action system is important to the AIDA's trajectory-matching performance. However, we did not investigate epistemic behavior in the model due to the simplicity of the car following task. The AIDA posed here also extends our prior work [31] to model fine-grained longitudinal control, validate that model against established benchmarks, and provide a more detailed interpretability analysis.

In addition to the contributions to driver modeling, this work extends research on human perception and control modeling. Our simultaneous estimation of human preference, understanding of environment dynamics (i.e., transition probabilities), and perceptual uncertainties (i.e., observation probabilities), and use of data from a complex driving environment differentiate this work from [59, 73, 74]. Our findings here suggest that the AIDA can be extended to complex environments successfully, although it is sensitive to training data and model parameterization. Our use of a data-driven prior distribution, i.e.,(11), to prevent estimating transition and observation parameters that are highly inconsistent with actual traffic dynamics and reduce unidentifiability is also novel and differentiates this work from [57] and [75]. Our visualizations of model preference and beliefs in Figure 6d and Figures 7-8 show that the proposed data-driven prior leads to preference and dynamics estimation consistent with the observed data and driver behavior theories.

Our work is limited by the following aspects. First, we have assumed three driver observation modalities: distance headway, relative speed, and τ^{-1} with respect to the lead vehicle. However, human drivers are known to monitor other surrounding vehicles while driving [6] and to have broader visual sampling [76]. Second, our parameterization of discrete states has limited the expressivity of the model and prevented inductive biases such as the smoothness of physical dynamics from being encoded. The limited dataset coverage, e.g., the lack of crashes, prevented the learned dynamics from generalizing to some out-of-distribution scenarios. The combination of model and data insufficiency led to the difficulty of recognizing near-crash states and resulted in substantially more lead vehicle crashes than BC-MLP and the IDM. Third, since the INTERACTION dataset was collected on highway, there likely exists considerable heterogeneous driving behavior. This is shown in the uncertain and multi-modal predictions in Figure 7 as the model had to explain drivers who took different actions upon observing similar signals. While we anticipate incorporating additional observations and higher state space dimension and application to alternative driving scenarios to be easy under the current model formulation, doing so would impose additional requirements on dataset quality and diversity. We thus recommend future work to consider general methods for incorporating domain knowledge in more expressive generative models to combat dataset limitations and modeling heterogeneity in naturalistic driver behavior. The results here suggest that these extensions may alleviate many of the current model limitations.

5 CONCLUSIONS

We proposed a novel active inference model of driver behavior (AIDA). Using car following data, we showed that the AIDA significantly outperformed the rule-based IDM on all metrics and performed comparably with the data-driven neural network benchmarks. Using an interpretability analysis, we showed that the structure of the AIDA provides superior transparency of its input-output mechanics than the neural network models. Future work should focus on training with data from more diverse driving environments and examining model extensions that can capture heterogeneity across drivers.

6 ACKNOWLEDGEMENTS

Support for this research was provided in part by the U.S. Department of Transportation (DOT), University Transportation Centers Program to the Safety through Disruption University Transportation Center (451453-19C36) and the UK Engineering and Physical Sciences Research Council (EPSRC; EP/S005056/1). Thanks to advisers, J. Engstrom and M. O'Kelly, from Waymo, who helped set the technical direction, identified relevant published research, and advised on the scope and structuring of this publication, independent of the support this research received from USDOT.

A APPENDIX

A.1 BC Implementation

For BC-MLP, we used a two-layer MLP network with ReLU activation and 40 hidden units in each layer. For BC-RNN, we used a two-layer MLP network on top of a single-layer GRU network with ReLU activation and 30 hidden units in each layer. The GRU layer only takes in past observations but not past actions. We found that larger number of hidden units in the BC-RNN model leads to significant overfitting. Both BC-MLP and BC-RNN receive 3 input observations and output probability distributions over 15 discrete actions.

A.2 AIDA Implementation

The AIDA implementation follows value-iteration network and QMDP network [57, 75] to enable end-to-end training in Pytorch [77]. We used a state dimension of 20, action dimension of 15, and a maximum planning horizon of 30 steps (3 seconds). The Normalizing Flow network consisted of a Gaussian mixture base distribution and a two-layer MLP network with ReLU activation and 30 hidden units in each layer. For each mini-batch of observation-action sequences, we first computed the log likelihood of the observations at all time steps and used (4) to compute the belief sequences. We then computed the cumulative EFE in (8) and the resulting optimal policy in (9) for each inferred belief using the QMDP approximation method [56]. We evaluated dataset action likelihood using a weighted average of optimal policies over different horizons:

$$\pi(a|b) = \sum_{H} \pi(a|b,H)P(H) \tag{14}$$

where $P(H)$ is a truncated Poisson distribution up to the maximum planning horizon.

The QMDP method assumes the belief-action value can be approximated as a weighted-average of the state-action value:

$$\mathcal{G}^*(b_t,a_t) = \sum_{s_t} b_t(s_t)\mathcal{G}^*(s_t,a_t) \tag{15}$$

where

$$\mathcal{G}^*(s_t,a_t) = EFE(s_t,a_t) + \log \pi(a_t|b_t) + \sum_{s_{t+1}} P(s_{t+1}|s_t,a_t)\mathcal{G}^*(s_{t+1}) \tag{16}$$

$$EFE(s_t,a_t) = D_{KL}(P(s_{t+1}|s_t,a_t)||\tilde{P}(s_{t+1})) + \mathbb{E}_{P(s_{t+1}|s_t,a_t)}[\mathcal{H}(P(o_{t+1}|s_{t+1}))] \tag{17}$$

and $\forall s \in \mathcal{S}, \mathcal{G}^*(s_{t+H+1}) = 0$.

The combination of QMDP approximation and computing the observation entropy in (17) using the Gaussian base distributions reduced the model's ability to evaluate state uncertainty. However, given the low state uncertainty shown in Figure 7 and Figure 8 (i.e., the nearly deterministic belief states in the lower right charts), these approximations do not significantly impact the current results while providing the benefit of computational tractability.

Another difference between our implementation and the common active inference presentation is that we performed exact Bayesian state inference (i.e., (4)) instead of approximate variational inference (e.g., in [27]). This does not impact the current results since both methods arrive at the same solution in the discrete state setting.

A.3 AIDA-MPC Implementation

In the AIDA-MPC model, we replaced the learned discrete environment dynamics model with a physics-based dynamics model with deterministic state transition and observation functions. The physical-based model had the same three observation modalities: $d, \Delta v, \tau^{-1}$. We defined the state space as $\{d, \Delta v\}$. Assuming constant lead vehicle acceleration, the state transition function is the following linear function:

$$\begin{bmatrix} d' \\ \Delta v' \end{bmatrix} = \begin{bmatrix} 1 & -\delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} d \\ \Delta v \end{bmatrix} + \begin{bmatrix} 0 \\ -\delta t \end{bmatrix} a \tag{18}$$

The state to observation mapping for d and Δv are identity functions. The observation τ^{-1} is computed as $\tau^{-1} \approx \Delta v / d$.

Given this dynamics model, we used the Cross-Entropy Method (CEM [70]) model predictive controller to generate actions treating the AIDA log preference probability over observation as the reward function, i.e., $R(o) = \log \sum_s P(o|s)\tilde{P}(s)$. At each time step, the CEM controller is initialized with a Gaussian distribution over finite horizon action sequences. It then iteratively refines the distribution by sampling N action sequences from the distribution, simulating the action sequences forward using the dynamics model, refitting the Gaussian distribution to the top K samples. Finally, it selects the first step of the mean action sequence of the final Gaussian distribution as the action output. We used a CEM planning horizon of 6 time steps (0.6 seconds), sampled $N = 50$ action sequences, selected the top $K = 5$ sequences, and refined the distribution for 20 iterations.

A.4 Parameter Counts

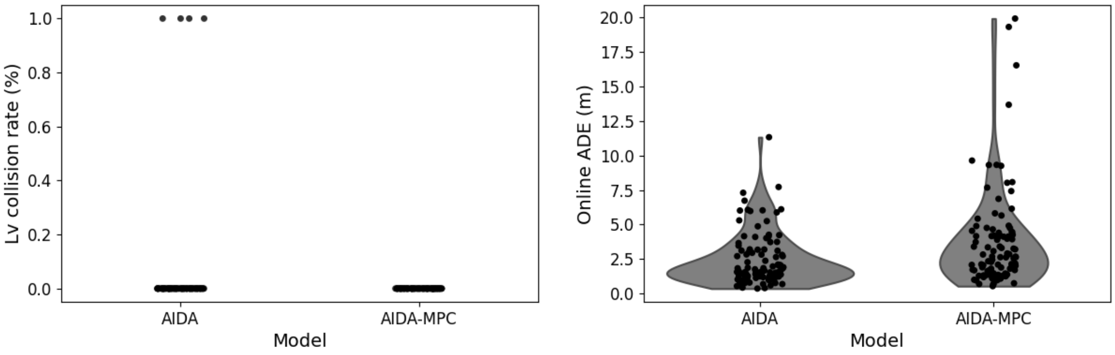
The number of parameters in each model is listed in Table 3.

TABLE 3: Parameter count of all models.

	IDM	BC-MLP	BC-RNN	AIDA
Count	6	4125	6465	7670

A.5 AIDA vs. AIDA-MPC

Figure 9a and 9b show AIDA and AIDA-MPC's lead vehicle collision rate and ADE for each tested trajectory, respectively, where each point corresponds to the result of a trajectory. The shadows in Figure 9b represent the density of each model's ADEs, where wider shadow represents higher density.



(a) (b)

FIGURE 9: Online same-lane evaluation results of AIDA and AIDA-MPC. Each point represents a trajectory in the test set. The AIDA-MPC replaces the AIDA's dynamics model with a physics-based dynamics model and plans by treating the AIDA's preference distribution as a reward function using model-predictive control. (a) Lead vehicle collision rate of each trajectory. (b) ADE of each trajectory. Wider shadows represent higher density of the ADE values.

REFERENCES

1. Scheel, O., L. Bergamini, M. Wolczyk, B. Osiński, and P. Ondruska, Urban driver: Learning to drive from real-world demonstrations using policy gradients. In *Conference on Robot Learning*, PMLR, 2022, pp. 718–728.
2. Scanlon, J. M., K. D. Kusano, T. Daniel, C. Alderson, A. Ogle, and T. Victor, Waymo simulated driving behavior in reconstructed fatal crashes within an autonomous vehicle operating domain. *Accident Analysis & Prevention*, Vol. 163, 2021, p. 106454.
3. Bärgman, J., C.-N. Boda, and M. Dozza, Counterfactual simulations applied to SHRP2 crashes: The effect of driver behavior models on safety benefit estimations of intelligent safety systems. *Accident Analysis & Prevention*, Vol. 102, 2017, pp. 165–180.
4. Roesener, C., J. Hiller, H. Weber, and L. Eckstein, How safe is automated driving? Human driver models for safety performance assessment. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2017, pp. 1–7.
5. Rhinehart, N., R. McAllister, K. Kitani, and S. Levine, Precog: Prediction conditioned on goals in visual multi-agent settings. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2821–2830.
6. Brown, B., The social life of autonomous cars. *Computer*, Vol. 50, No. 2, 2017, pp. 92–96.
7. Markkula, G., Y.-S. Lin, A. R. Srinivasan, J. Billington, M. Leonetti, A. H. Kalantari, Y. Yang, Y. M. Lee, R. Madigan, and N. Merat, Explaining human interactions on the road requires large-scale integration of psychological theory, 2022.
8. Räukur, T., A. Ho, S. Casper, and D. Hadfield-Menell, Toward transparent ai: A survey on interpreting the inner structures of deep neural networks. *arXiv preprint arXiv:2207.13243*, 2022.
9. Alambeigi, H., A. D. McDonald, and S. R. Tankasala, *Crash Themes in Automated*

Vehicles: A Topic Modeling Analysis of the California Department of Motor Vehicles Automated Vehicle Crash Database, 2020.

10. Novakazi, F., M. Johansson, H. Strömberg, and M. Karlsson, Levels of what? Investigating drivers' understanding of different levels of automation in vehicles. *Journal of Cognitive Engineering and Decision Making*, Vol. 15, No. 2-3, 2021, pp. 116–132.
11. Saifuzzaman, M. and Z. Zheng, Incorporating human-factors in car-following models: a review of recent developments and research needs. *Transportation research part C: emerging technologies*, Vol. 48, 2014, pp. 379–403.
12. McDonald, A. D., H. Alambeigi, J. Engström, G. Markkula, T. Vogelpohl, J. Dunne, and N. Yuma, Toward computational simulations of behavior during automated driving takeovers: a review of the empirical and modeling literatures. *Human factors*, Vol. 61, No. 4, 2019, pp. 642–688.
13. Kesting, A., M. Treiber, and D. Helbing, Agents for traffic simulation. *Multi-agent systems: Simulation and applications*, Vol. 5, 2009.
14. Hamdar, S. H. and H. S. Mahmassani, From existing accident-free car-following models to colliding vehicles: exploration and assessment. *Transportation research record*, Vol. 2088, No. 1, 2008, pp. 45–56.
15. Talebpour, A. and H. S. Mahmassani, Influence of connected and autonomous vehicles on traffic flow stability and throughput. *Transportation Research Part C: Emerging Technologies*, Vol. 71, 2016, pp. 143–163.
16. Zhou, M., X. Qu, and X. Li, A recurrent neural network based microscopic car following model to predict traffic oscillation. *Transportation research part C: emerging technologies*, Vol. 84, 2017, pp. 245–264.
17. Mo, Z., R. Shi, and X. Di, A physics-informed deep learning paradigm for car-following models. *Transportation research part C: emerging technologies*, Vol. 130, 2021, p. 103240.
18. Zhu, M., X. Wang, and Y. Wang, Human-like autonomous car-following model with deep reinforcement learning. *Transportation research part C: emerging technologies*, Vol. 97, 2018, pp. 348–368.
19. Bhattacharyya, R., B. Wulfe, D. Phillips, A. Kuefler, J. Morton, R. Senanayake, and M. Kochenderfer, Modeling human driving behavior through generative adversarial imitation learning. *arXiv preprint arXiv:2006.06412*, 2020.
20. Spencer, J., S. Choudhury, A. Venkatraman, B. Ziebart, and J. A. Bagnell, Feedback in imitation learning: The three regimes of covariate shift. *arXiv preprint arXiv:2102.02872*, 2021.
21. Pomerleau, D. A., Alvinn: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, Vol. 1, 1988.
22. Codevilla, F., E. Santana, A. M. López, and A. Gaidon, Exploring the limitations of behavior cloning for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9329–9338.
23. Kumar, A., J. Hong, A. Singh, and S. Levine, Should I Run Offline Reinforcement Learning or Behavioral Cloning? In *International Conference on Learning Representations*, 2021.

24. Igl, M., D. Kim, A. Kuefler, P. Mougin, P. Shah, K. Shiarlis, D. Anguelov, M. Palatucci, B. White, and S. Whiteson, Symphony: Learning Realistic and Diverse Agents for Autonomous Driving Simulation. *arXiv preprint arXiv:2205.03195*, 2022.
25. Suo, S., S. Regalado, S. Casas, and R. Urtasun, Trafficsim: Learning to simulate realistic multi-agent behaviors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10400–10409.
26. Friston, K., The free-energy principle: a unified brain theory? *Nature reviews neuroscience*, Vol. 11, No. 2, 2010, pp. 127–138.
27. Friston, K., T. FitzGerald, F. Rigoli, P. Schwartenbeck, and G. Pezzulo, Active inference: a process theory. *Neural computation*, Vol. 29, No. 1, 2017, pp. 1–49.
28. Doya, K., S. Ishii, A. Pouget, and R. P. Rao, *Bayesian brain: Probabilistic approaches to neural coding*. MIT press, 2007.
29. Gershman, S. J., E. J. Horvitz, and J. B. Tenenbaum, Computational rationality: A converging paradigm for intelligence in brains, minds, and machines. *Science*, Vol. 349, No. 6245, 2015, pp. 273–278.
30. Smith, R., P. Badcock, and K. J. Friston, Recent advances in the application of predictive coding and active inference models within clinical neuroscience. *Psychiatry and Clinical Neurosciences*, Vol. 75, No. 1, 2021, pp. 3–13.
31. Wei, R., A. D. McDonald, A. Garcia, and H. Alambeigi, Modeling Driver Responses to Automation Failures With Active Inference. *IEEE Transactions on Intelligent Transportation Systems*, 2022.
32. Engström, J., S.-Y. Liu, A. Dinparastdjadid, and C. Simoiu, Modeling road user response timing in naturalistic settings: a surprise-based framework. *arXiv preprint arXiv:2208.08651*, 2022.
33. Treiber, M., A. Hennecke, and D. Helbing, Congested traffic states in empirical observations and microscopic simulations. *Physical review E*, Vol. 62, No. 2, 2000, p. 1805.
34. Papathanasopoulou, V. and C. Antoniou, Towards data-driven car-following models. *Transportation Research Part C: Emerging Technologies*, Vol. 55, 2015, pp. 496–509.
35. Treiber, M. and A. Kesting, Microscopic calibration and validation of car-following models–a systematic approach. *Procedia-Social and Behavioral Sciences*, Vol. 80, 2013, pp. 922–939.
36. Ross, S. and D. Bagnell, Efficient reductions for imitation learning. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, JMLR Workshop and Conference Proceedings, 2010, pp. 661–668.
37. De Haan, P., D. Jayaraman, and S. Levine, Causal confusion in imitation learning. *Advances in Neural Information Processing Systems*, Vol. 32, 2019.
38. Beer, R. D., The dynamics of active categorical perception in an evolved model agent. *Adaptive behavior*, Vol. 11, No. 4, 2003, pp. 209–243.
39. Friston, K., L. Da Costa, D. Hafner, C. Hesp, and T. Parr, Sophisticated inference. *Neural Computation*, Vol. 33, No. 3, 2021, pp. 713–763.

40. Tschantz, A., A. K. Seth, and C. L. Buckley, Learning action-oriented models through active inference. *PLoS computational biology*, Vol. 16, No. 4, 2020, p. e1007805.

41. Haarnoja, T., A. Zhou, P. Abbeel, and S. Levine, Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, PMLR, 2018, pp. 1861–1870.

42. Schwartenbeck, P., T. H. FitzGerald, C. Mathys, R. Dolan, F. Wurst, M. Kronbichler, and K. Friston, Optimal inference with suboptimal models: addiction and active Bayesian inference. *Medical hypotheses*, Vol. 84, No. 2, 2015, pp. 109–117.

43. Engström, J., J. Bärgman, D. Nilsson, B. Seppelt, G. Markkula, G. B. Piccinini, and T. Victor, Great expectations: a predictive processing account of automobile driving. *Theoretical issues in ergonomics science*, Vol. 19, No. 2, 2018, pp. 156–194.

44. Zhan, W., L. Sun, D. Wang, H. Shi, A. Clausse, M. Naumann, J. Kummerle, H. Konigshof, C. Stiller, A. de La Fortelle, et al., Interaction dataset: An international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps. *arXiv preprint arXiv:1910.03088*, 2019.

45. Poggenhans, F., J.-H. Pauls, J. Janosovits, S. Orf, M. Naumann, F. Kuhnt, and M. Mayr, Lanelet2: A high-definition map framework for the future of automated driving. In *2018 21st international conference on intelligent transportation systems (ITSC)*, IEEE, 2018, pp. 1672–1679.

46. Svärd, M., G. Markkula, J. Bärgman, and T. Victor, Computational modeling of driver pre-crash brake response, with and without off-road glances: Parameterization using real-world crashes and near-crashes. *Accident Analysis & Prevention*, Vol. 163, 2021, p. 106433.

47. Engström, J., G. Markkula, Q. Xue, and N. Merat, Simulating the effect of cognitive load on braking responses in lead vehicle braking scenarios. *IET Intelligent Transport Systems*, Vol. 12, No. 6, 2018, pp. 427–433.

48. Markkula, G., J. Engström, J. Lodin, J. Bärgman, and T. Victor, A farewell to brake reaction times? Kinematics-dependent brake response in naturalistic rear-end emergencies. *Accident Analysis & Prevention*, Vol. 95, 2016, pp. 209–226.

49. Werling, M., J. Ziegler, S. Kammel, and S. Thrun, Optimal trajectory generation for dynamic street scenarios in a frenet frame. In *2010 IEEE International Conference on Robotics and Automation*, IEEE, 2010, pp. 987–993.

50. Murphy, K. P., *Machine learning: a probabilistic perspective*. MIT press, 2012.

51. Jung, S., R. Senanayake, and M. J. Kochenderfer, A Gray Box Model for Characterizing Driver Behavior. In *SafeAI@ AAAI*, 2022.

52. Chung, J., C. Gulcehre, K. Cho, and Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

53. Rezende, D. and S. Mohamed, Variational inference with normalizing flows. In *International conference on machine learning*, PMLR, 2015, pp. 1530–1538.

54. Papamakarios, G., E. T. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan, Normalizing Flows for Probabilistic Modeling and Inference. *J. Mach. Learn. Res.*, Vol. 22, No. 57, 2021, pp. 1–64.

55. Kingma, D. P., T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling, Improved variational inference with inverse autoregressive flow. *Advances in neural information processing systems*, Vol. 29, 2016.

56. Littman, M. L., A. R. Cassandra, and L. P. Kaelbling, Learning policies for partially observable environments: Scaling up. In *Machine Learning Proceedings 1995*, Elsevier, 1995, pp. 362–370.

57. Karkus, P., D. Hsu, and W. S. Lee, Qmdp-net: Deep learning for planning under partial observability. *Advances in neural information processing systems*, Vol. 30, 2017.

58. Wei, R., *Active Inference Interaction Modeling*. `https://github.com/rw422scarlet/interactive_inference`, 2022.

59. Reddy, S., A. Dragan, and S. Levine, Where do you think you're going?: Inferring beliefs about dynamics from behavior. *Advances in Neural Information Processing Systems*, Vol. 31, 2018.

60. Armstrong, S. and S. Mindermann, Impossibility of deducing preferences and rationality from human policy. *arXiv preprint arXiv:1712.05812*, 2017.

61. Bouchard, G. and B. Triggs, The tradeoff between generative and discriminative classifiers. In *16th IASC International Symposium on Computational Statistics (COMPSTAT'04)*, 2004, pp. 721–728.

62. Wei, R., A. Garcia, A. McDonald, G. Markkula, J. Engström, I. Supeene, and M. O'Kelly, World Model Learning From Demonstrations With Active Inference: Application to Driving Behavior. In *Forthcoming, 3rd International Workshop on Active Inference, Grenoble, France*, 2022.

63. Sadigh, D., S. Sastry, S. A. Seshia, and A. D. Dragan, Planning for autonomous cars that leverage effects on human actions. In *Robotics: Science and systems*, Ann Arbor, MI, USA, 2016, Vol. 2, pp. 1–9.

64. Colas, C., O. Sigaud, and P.-Y. Oudeyer, A hitchhiker's guide to statistical comparisons of reinforcement learning algorithms. *arXiv preprint arXiv:1904.06979*, 2019.

65. Agarwal, R., M. Schwarzer, P. S. Castro, A. C. Courville, and M. Bellemare, Deep reinforcement learning at the edge of the statistical precipice. *Advances in neural information processing systems*, Vol. 34, 2021, pp. 29304–29320.

66. *Sedan vehicle dimensions*. `https://www.mathworks.com/help/driving/ref/sedan.html`, 2022, accessed: 2022-12-15.

67. Fuller, R., Towards a general theory of driver behaviour. *Accident analysis & prevention*, Vol. 37, No. 3, 2005, pp. 461–472.

68. Summala, H., Hierarchical model of behavioural adaptation and traffic accidents. *Traffic and transport psychology. Theory and application*, 1997.

69. Hancock, P., Is car following the real question–are equations the answer? *Transportation research part F: traffic psychology and behaviour*, Vol. 2, No. 4, 1999, pp. 197–199.

70. De Boer, P.-T., D. P. Kroese, S. Mannor, and R. Y. Rubinstein, A tutorial on the cross-entropy method. *Annals of operations research*, Vol. 134, No. 1, 2005, pp. 19–67.

71. Rudin, C., Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, Vol. 1, No. 5, 2019, pp. 206–215.

72. Pekkanen, J., O. Lappi, P. Rinkkala, S. Tuhkanen, R. Frantsi, and H. Summala, A computational model for driver's cognitive state, visual perception and intermittent attention in a distracted car following task. *Royal Society open science*, Vol. 5, No. 9, 2018, p. 180194.

73. Herman, M., T. Gindele, J. Wagner, F. Schmitt, and W. Burgard, Inverse reinforcement learning with simultaneous estimation of rewards and dynamics. In *Artificial Intelligence and Statistics*, PMLR, 2016, pp. 102–110.

74. Baker, C., R. Saxe, and J. Tenenbaum, Bayesian theory of mind: Modeling joint belief-desire attribution. In *Proceedings of the annual meeting of the cognitive science society*, 2011, Vol. 33.

75. Tamar, A., Y. Wu, G. Thomas, S. Levine, and P. Abbeel, Value iteration networks. *Advances in neural information processing systems*, Vol. 29, 2016.

76. Victor, T., M. Dozza, J. Bärgman, C.-N. Boda, J. Engström, C. Flannagan, J. D. Lee, and G. Markkula, *Analysis of naturalistic driving study data: Safer glances, driver inattention, and crash risk*, 2015.

77. Paszke, A., S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, Vol. 32, 2019.