2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Connected and Automated Vehicles in Mixed-Traffic: Learning Human Driver Behavior for Effective On-Ramp Merging

Nishanth Venkatesh[1], Viet-Anh Le[2,1], Aditya Dave[3], *Student Members, IEEE,*
Andreas A. Malikopoulos[1,3], *Senior Member, IEEE*

***Abstract*— Highway merging scenarios featuring mixed traffic conditions pose significant modeling and control challenges for connected and automated vehicles (CAVs) interacting with incoming on-ramp human-driven vehicles (HDVs). In this paper, we present an approach to learn an approximate information state (AIS) model of *CAV-HDV interactions*. Thus, the CAV learns the behavior of an incoming HDV using the AIS model and uses it to generate a control strategy for merging. First, we validate the efficacy of this framework on real-world data by using it to predict the behavior of an HDV in situations with other HDVs extracted from the Next-Generation Simulation repository. Then, we generate simulation data for HDV-CAV interactions in a highway merging scenario using a standard inverse reinforcement learning approach. Without assuming a prior knowledge of the generating model, we show that our AIS model learns to predict the future trajectory of the HDV using only observations. Subsequently, we generate safe merging control policies for a CAV when merging with HDVs that demonstrate a spectrum of driving behaviors, from aggressive to conservative. We establish the effectiveness of the proposed approach by performing numerical simulations.**

I. Introduction

Connected and automated vehicles (CAVs) have the potential to significantly improve transportation networks. Numerous research efforts have focused on how to ensure energy efficiency [1], safety [2], and traveler comfort [3] with 100 % CAV penetration. However, since we expect a the number of CAVs on the roads to rise gradually, CAVs must also safely maneuver alongside human-driven vehicles (HDVs). This poses a challenge in the control of CAVs due to the unpredictability of HDV behavior and thus, we require new approaches to predict human-driving behavior online and dynamically adjust a CAV's trajectory.

In recent years, significant attention has been directed toward game-theoretic models for such interactions between vehicles. Liu *et al*. [4] proposed a decision-making algorithm to address game theoretic interactions in merging scenarios. Chandra and Manocha [5] proposed a robotics-based auction framework for vehicle navigation. A game-theoretic model predictive control with weight adaptation strategies for CAVs in mixed-traffic merging scenarios was proposed in [6].

This research was supported by NSF under Grants CNS-2149520 and CMMI-2219761.

[1]Systems Engineering, Cornell University, Ithaca, NY 14850 USA.
[2]Department of Mechanical Engineering, University of Delaware, Newark, DE 19716 USA.
[3]School of Civil and Environmental Engineering, Cornell University, Ithaca, NY 14850 USA. email: {ns942, vl299, a.dave, amaliko}@cornell.edu

Other approaches have used deep learning to predict HDV behaviors. Altché and Fortelle [7] proposed a a long short-term memory (LSTM) architecture to predict the trajectory of a vehicle on a straight road. Park *et al*. [8] generated the future trajectories of all vehicles surrounding an HDV using an LSTM model. Similarly, Deo and Trivedi used LSTMs for interaction-aware motion prediction of surrounding vehicles on freeways [9]. While most deep learning methods predict the future behavior of HDVs, there is a growing interest in utilizing such predictions to control CAVs. Kherroubi *et al*. [10] trained a neural network model to predict the intentions of human drivers and learn a driving strategy. Guo *et al*. [11] applied Q-learning to control the change of lanes for CAVs when HDVs follow Gipps' car-following model. However, there is still a need for a framework of learning-enabled [12] CAVs that prioritize safety while generalizing across different models of HDV behavior and decentralized information structures [13].

To this end, we draw upon ideas in the literature on partially observed reinforcement learning. Subramanian *et al*. [14] presented a principled framework to learn state-space representations and compute control strategies for systems with hidden states. Such a model, known as an approximate information state (AIS) model, has been extended to robust formulations [15], [16] and also utilized in robotics [17] and medical applications [18]. Building from the notion of an AIS, our main contributions are as follows. We provide a methodology to learn an AIS model for an on-ramp HDV's behavior in highway merging and generate a safety prioritized strategy for a CAV. Here, the CAV's control problem includes the objectives of safety and energy efficiency. Specifically: (1) We construct and train a neural network architecture to predict the trajectory of an HDV without prior knowledge of their dynamics. We achieve this by considering the history of observations to be representative of the HDV's driving behavior and compressing it into an AIS. (2) We develop an iterative model predictive control (MPC) algorithm which uses the learned AIS model to generate a safe and energy-efficient control strategy for the CAV online. (3) We validate the efficacy of our proposed learning framework by successfully predicting vehicle trajectories from the Next Generation Simulation (NGSIM) repository. (4) We validate the efficacy of our learning and control approach across 5000 simulations of HDV-CAV interactions, featuring HDV behaviors ranging from aggressive to conservative.

The remainder of the paper proceeds as follows. In Section II, we present our problem formulation. In Section III, we

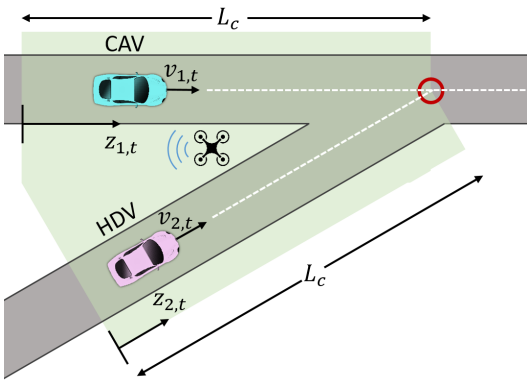


Fig. 1: Mixed traffic merging scenario.

present preliminaries on AISs and our proposed encoder-decoder model to learn *CAV-HDV interactions*. In Section IV, we use the learned model to solve our control problem and present numerical simulation results. Finally, in Section V, we draw concluding remarks.

II. Problem Formulation

In this section, we formulate an MPC problem to compute the control input for a CAV in a mixed-traffic highway merging scenario. As shown in Fig. 1, a CAV indexed by 1, and an HDV indexed by 2, travel on a main road and a ramp, respectively. We define a *control zone* as highlighted by the green region in Fig. 1. Within the control zone, the control inputs to CAV–1 are determined by our proposed method, whereas, outside the control zone, they can be determined by a standard car-following model [11]. We consider that the control zone begins at a distance $L_c \in \mathbb{R}_{>0}$ upstream of the *conflict point* on each road, i.e., the location where the paths of CAV–1 and HDV–2 intersect. The conflict point is where a lateral collision may occur (a red circle in Fig. 1). Our goal is to control CAV–1 to merge safely and effectively, given the presence of HDV–2 in this merging scenario.

We measure the longitudinal positions of CAV–1 and HDV–2 from the beginning of the control zone at any $t \in \mathbb{N}$, and denote them by $z_{1,t} \in \mathbb{R}_{\geq 0}$ and $z_{2,t} \in \mathbb{R}_{\geq 0}$, respectively. Let $v_{1,t} \in \mathbb{R}_{\geq 0}$ and $u_{1,t} \in \mathbb{R}$ be the speed and acceleration of CAV–1, $v_{2,t} \in \mathbb{R}_{\geq 0}$ and $u_{2,t} \in \mathbb{R}$ be the speed and acceleration of HDV–2 at time t. We denote the position and speed of each vehicle $i = 1, 2$ as its state $x_{i,t} = (z_{i,t}\,,\, v_{i,t})$ at time t. Then, starting with the initial state $x_{1,0}$ and $x_{2,0}$ at $t = 0$, the state of each vehicle evolves according to $x_{1,t+1} = f_{1,t}(x_{1,t}, u_{1,t})$ and $x_{2,t+1} = f_{2,t}(x_{2,t}, u_{2,t})$. In this paper, we consider that for all $t \in \mathbb{N}$ the dynamics of CAV–1 follow a double-integrator model:

$$\begin{aligned} z_{1,t+1} &= z_{1,t} + \Delta T v_{1,t} + \frac{1}{2}\Delta T^2 u_{1,t}, \\ v_{1,t+1} &= v_{1,t} + \Delta T u_{1,t}, \end{aligned} \tag{1}$$

where $\Delta T \in \mathbb{R}_{>0}$ is the sampling time between two discrete timesteps. We consider the following constraints for CAV–1

$$0 \leq v_{\min} \leq v_{1,t} \leq v_{\max}, \quad \text{and} \quad u_{\min} \leq u_{1,t} \leq u_{\max}, \tag{2}$$

where $u_{\min},\ u_{\max} \in \mathbb{R}$ are the minimum deceleration and maximum acceleration, respectively, and $v_{\min},\ v_{\max} \in \mathbb{R}_{>0}$ are the minimum and maximum speed limits, respectively. Both the vehicles are restricted to only move in the forward direction, and hence, their speeds $v_{i,t}$ for $i = 1, 2$ are always non-negative. However, we do not impose an upper bound $v_{2,t}$ because it can be violated by HDV–2. In the next subsection, we formulate the control problem for CAV–1.

A. Model Predictive Control Formulation

In this subsection, we formulate an MPC problem to control CAV–1 in the merging scenario. Let $H \in \mathbb{N}$ be the time length of the control horizon, $t \in \mathbb{N}$ be the current time step, and $\mathcal{I}_t = \{t, \ldots, t + H - 1\}$ be the control horizon at t. The objective for CAV–1 at any time k is a linear combination of: *(i)* the control input, *(ii)* the deviation from the maximum allowed speed to reduce the travel time, and *(iii)* a logarithmic penalty for collision avoidance [6], i.e.,

$$\begin{aligned} l(x_{1,k+1}, u_{1,k}) = \omega_1 u_{1,k}^2 + \omega_2 (v_{1,k+1} - v_{\max})^2 - \omega_3 \log(\\ (z_{1,k+1} - z_c + \rho v_{1,k+1})^2 + (\hat{z}_{2,k+1} - z_c + \rho \hat{v}_{2,k+1})^2), \end{aligned} \tag{3}$$

where ω_1, ω_2, and ω_3 are positive weights, z_c is the position of the conflict point, and $\rho \in \mathbb{R}_{>0}$ accounts for the reaction delay of HDV–2. An increase in ρ signifies greater conservatism by considering HDV–2 with longer delays in reaction. In (3), $\hat{z}_{2,k+1}$ and $\hat{v}_{2,k+1}$ denote the predicted future positions and speeds of HDV–2 at time step $k+1$. Then, the MPC problem for CAV–1 for each $t \in \mathbb{N}$ is

$$\underset{\{u_{1,k}\}_{k \in \mathcal{I}_t}}{\text{minimize}} \sum_{k \in \mathcal{I}_t} l(x_{1,k+1}, u_{1,k}), \tag{4a}$$

$$\text{subject to: } (1), (2). \tag{4b}$$

We seek the control inputs $u_{1,j},\ \forall j \in \mathcal{I}_t$ to minimize (4) at each $t \in \mathbb{N}$ without prior knowledge of the dynamics of HDV–2. Thus, CAV–1 must learn a model to predict the future trajectory of HDV–2 using real time data. Thus, we impose the following assumption.

Assumption 1. A coordinator, as shown in Fig. 1, exists to collect perfect and transmit observations on the states and control actions of HDV–2 in real-time.

Remark 1. The future control actions and the trajectory for HDV–2 are not pre-determined because the dynamics of HDV–2 are unknown, and the impact of of CAV–1 on the actions of HDV–2 is also unknown. We model the joint dynamics of both CAV–1 and HDV–2 as an unknown *partially observable Markov decision process* (POMDP), and learn an AIS model for the same.

III. Learning Framework

A. Preliminaries

In this subsection, we present the mathematical formulation of POMDPs and the framework of AISs from [14] that we will use to predict the future trajectory of HDV–2.

POMDP Fundamentals: We define a POMDP as a tuple $(\mathcal{X}, \mathcal{U}, \mathcal{Y}, T, O, c, \gamma)$, where the set of feasible states is $\mathcal{X}$, the

set of feasible control actions is $\mathcal{U}$, the set of feasible observations is $\mathcal{Y}$, the function $T : \mathcal{X} \times \mathcal{U} \times \mathcal{X} \to [0, 1]$ yields the transition probability $T(x_t, u_{t-1}, x_{t+1}) = p(x_{t+1}|x_t, u_{t-1})$ for all $x_t, x_{t+1} \in \mathcal{X}$ and $u_t \in \mathcal{U}$, the function $O : \mathcal{X} \times \mathcal{Y} \to [0, 1]$ yields the observation probability $O(x_{t+1}, y_{t+1}) = p(y_{t+1}|x_{t+1})$ for all $x_{t+1} \in \mathcal{X}$ and $y_{t+1} \in \mathcal{Y}$, the function $c : \mathcal{X} \times \mathcal{U} \to \mathbb{R}$ yields the cost $c(x_t, u_t)$ for all $x_t \in \mathcal{X}$ and $u_t \in \mathcal{U}$, and $\gamma \in [0, 1)$ is the discount factor. The history of observations and control actions up to a given time t is the memory $m_t = (m_{t-1}, y_t, u_{t-1}) \in \mathcal{M}_t$.

AIS: An AIS is a compression of the memory which can be used to compute an approximately optimal control strategy for a POMDP using a dynamic programming decomposition. Note that an AIS model can be learned purely from observation data without complete knowledge of underlying dynamics. Consider a measurable space $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$, where $\mathcal{B}(\mathcal{X})$ is the sigma algebra on $\mathcal{X}$. We denote an integral probability metric between any two probability distributions $\mu, \nu \in \Delta(\mathcal{X})$ by $d(\cdot, \cdot)$. Examples of such metrics are Wasserstein metric and maximum mean discrepancy metric. Then, for a POMDP with a horizon $T \in \mathbb{N}$, a time-invariant AIS model is defined by a tuple $(\mathcal{S}, \hat{p}, \hat{c}, \sigma_t : t = 0, \dots, T)$, where $\mathcal{S}$ is a Banach space of feasible information states, $\hat{p} : \mathcal{S} \times \mathcal{U} \to \Delta(\mathcal{S})$ is a Markovian probability kernel, $\hat{c} : \mathcal{S} \times \mathcal{U} \to \mathbb{R}$ is an approximate cost function, and $\sigma_t : \mathcal{M}_t \to \mathcal{S}$ is a memory compression function. This constitutes an (ε, δ)-approximate model of the POMDP if there exist $\varepsilon, \delta \in \mathbb{R}_{>0}$ such that for all realizations of memory $m_t \in \mathcal{M}_t$ and realizations of control action $u_t \in \mathcal{U}$, the following properties hold:

(**AP1**) Sufficient to approximate the cost $c_t = c(x_t, u_t)$:

$$|\mathbb{E}[c_t(x_t, u_t) \,|\, m_t, u_t] - \hat{c}(\sigma_t(m_t), u_t)| \leq \varepsilon. \tag{5}$$

(**AP2a**) The AIS evolves in a deterministic manner, i.e., there exists a measurable function $\psi : \mathcal{S} \times \mathcal{Y} \times \mathcal{U} \to \mathcal{S}$ such that

$$\sigma_t(m_t) = \psi(\sigma_t(m_{t-1}), y_t, u_{t-1}). \tag{6}$$

(**AP2b**) Sufficient to approximately predict future observations, i.e., there exists a probability kernel $\hat{p}^{\mathrm{y}} : \mathcal{S} \times \mathcal{U} \to \Delta(\mathcal{Y})$ such that for any Borel subset B of $\mathcal{Y}$, the memory-based distribution $\mu_t(B) = p(y_{t+1} \in B|m_t, u_t)$ and the predicted distribution $\phi(B) = \hat{p}^{\mathrm{y}}(y_{t+1} \in B|\sigma_t(m_t), u_t)$:

$$d(\mu_t, \phi) \leq \delta. \tag{7}$$

An (ε, δ)-AIS model forms a perfectly observed Markov decision process with the state at each t given by $s_t \in \mathcal{S}$. Thus, we can use it to compute an approximately optimal control strategy by formulating a dynamic programming decomposition of the problem. Such an approximate strategy has a bounded performance loss which improves linearly with a decrease in ε and δ [14]. Thus, an AIS constitutes a principled compression of the memory learned by enforcing (**AP1**) and (**AP2b**) while satisfying (**AP2a**).

In light of these advantages, we adopt this framework in our CAV-HDV merging scenario to predict the trajectory of HDV–2 and, subsequently, compute a control strategy for CAV–1. In this framework, at each $t \in \mathbb{N}$, the state of CAV–1 is $x_{1,t} = (z_{1,t}, v_{1,t})$, the control action is $u_{1,t}$ and the observation received by CAV–1 is $y_t = (z_{1,t}, v_{1,t}, u_{1,t-1}, z_{2,t}, v_{2,t}, u_{2,t-1})$. We deploy an encoder-decoder neural network architecture to represent an AIS model which learns a representation of the unknown impact of CAV–1's actions on the actions of HDV–2. This is illustrated in Fig. 2, where we denote the encoder by $\psi : \mathcal{S} \times \mathcal{Y} \times \mathcal{U} \to \mathcal{S}$ and the decoder by $\phi : \mathcal{S} \times \mathcal{U} \to \mathcal{Y}^3$, where $\mathcal{Y} \subset \mathbb{R}^6$. We seek to learn this AIS model purely from observation data. For this, we compare the distributions predicted by the decoder against data that constitutes sampled points from the ground truth. Next, we describe each component of our neural network architecture.

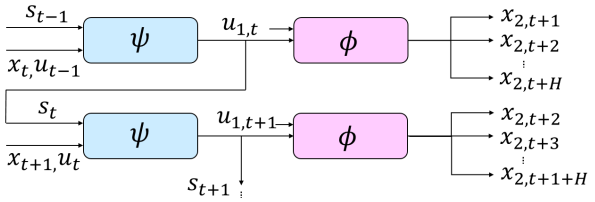


Fig. 2: A visualization of the encoder-decoder architecture.

B. Encoder-Decoder Architecture

1) The encoder ψ aims to learn a mapping which enforces (**AP2a**) i.e., its aim is to compress the memory m_t into an AIS $s_t = \sigma_t(m_t)$ at each $t \in \mathbb{N}$. Note that its input includes the previous AIS, latest observation, and previous control action given as a tuple (s_{t-1}, y_t, u_{t-1}). We illustrate this in Fig. 2. The encoder comprises of two fully connected layers followed by a recurrent neural network (RNN). The hidden state of the RNN at each $t \in \mathbb{N}$ is treated as the output of the encoder s_t. The linear layers receive the inputs (y_t, u_{t-1}), and the RNN ensures that the encoder receives the previous AIS s_{t-1}. Thus, the AIS is updated at each $t \in \mathbb{N}$ as

$$s_t = \psi(s_{t-1}, y_t, u_{t-1}). \tag{8}$$

2) The decoder ϕ aims to learn a mapping to enforce (**AP1**) and (**AP2b**), i.e., to predict a distribution on future observations and the expected cost based on the AIS and the control action. The decoder comprises of three fully connected layers. At each $t \in \mathbb{N}$, the decoder takes as an input the AIS s_t and control action $u_{1,t}$ of CAV–1. The decoder generates a distribution of future states of HDV–2 over a horizon $\mathcal{I}_t$, $(x_{2,t+1}, x_{2,t+2}, \dots, x_{2,t+H})$ given by

$$\phi(s_t, u_{1,t}) = \hat{p}^{\mathrm{y}}(x_{2,t+1}, \dots, x_{2,t+H} \mid s_t, u_{1,t}). \tag{9}$$

We enforce (**AP2b**) for the entire control horizon $\mathcal{I}_t$ to serve two purposes: (i) to enable the decoder to predict the future states of HDV–2, and (ii) to allow a planning depth across the control horizon $\mathcal{I}_t$ during the implementation of the MPC controller. We do not include an expected cost term in the output of the decoder because the cost (3) at each time is deterministic. Thus, if the decoder learns to best predict the trajectories, it automatically ensures both (**AP1**) and (**AP2b**).

C. Training

In this subsection, we describe how we train both the encoder ψ and the decoder ϕ by considering the left-hand side in (7) from (**AP2b**) as a training loss. Let the underlying distribution on the ground truth be denoted by p and the predicted distribution of the decoder be $\hat{p}^{\mathrm{y}}$. The training loss is given by $d(\hat{p}^{\mathrm{y}}, p)^2$, where $d(\cdot,\cdot)$ is the distance-based maximum mean discrepancy (MMD) metric for two probability distributions [14, Proposition 32]. We consider that $\hat{p}^{\mathrm{y}}$ is a multivariate normal distribution on the possible future realizations of the states $(x_{2,t+1}, \ldots, x_{2,t+H})$ of HDV–2 over the horizon $\mathcal{I}_t$. Then, we fix the standard deviation of $\hat{p}^{\mathrm{y}}$ to 1 and consider that outputs of the decoder represent the mean value vector of the distribution. At each $t \in \mathbb{N}$, we have access to a sampled observation $(x^{\mathrm{ref}}_{2,t+1}, \ldots, x^{\mathrm{ref}}_{2,t+H})$ from the underlying distribution p. To minimize the loss $d(\hat{p}^{\mathrm{y}}, p)^2$, we need to construct an estimator of the gradient of this training loss using only sampled points. From [14, Proposition 33], an unbiased estimate of the gradient $\nabla d(\hat{p}^{\mathrm{y}}, p)^2$ from the mean of $\hat{p}^{\mathrm{y}}$ and sampled points from p is the gradient of $(x_{2,t+i} - 2 \cdot x^{ref}_{2,t+i})^\top x_{2,t+i}$ for all $i = 1, \ldots, H$. This estimate is a surrogate loss for the training.

IV. Simulation and Results

We train and validate the use of AIS models for three datasets for highway on-ramp merging scenarios. For the first dataset, we use trajectory data from the NGSIM repository for I-80 [19] to extract the positions, velocities, and accelerations of the on-ramp vehicle along with the two consecutive interacting highway vehicles. Using this data, we validate the predictive ability of our AIS model in real-life merging scenarios and we present the results in Subsection IV-A.

Next, we use an *inverse reinforcement learning* (IRL) technique described in [6] to replicate human drivers in simulations and generate two more datasets for the scenario presented in Section II. With the IRL technique, we assume that the control actions of an HDV minimize an objective function based on different driving styles, ranging from conservative to aggressive. In the second dataset, we simulate CAV–1 with a safe control strategy from [6] against HDVs with different levels of aggression. This ensures that only 1% of the included situations demonstrate unsafe merging. In the third dataset, the control actions of both HDV–2 and CAV–1 are generated using the IRL technique with randomized weights to simulate an exploratory control strategy. The details of training and validation of predictions are presented in Subsection IV-A. Furthermore, we utilize the prediction of the AIS model to generate a control strategy for the CAV using iterative MPC in Subsection IV-B.

A. Network Architecture and Prediction Results

1) The NGSIM dataset model takes in as an input to the encoder ψ, a 9 dimensional vector made up of 2 dimensional positions and speeds for the three vehicles. The linear layers of the encoder are of dimensions $(9, 8)$ and $(8, 16)$ each with a ReLU activation, followed by a recurrent gated unit (GRU) layer with a hidden state of size 24. The hidden state, which is the AIS along with the most recent actions of each vehicle are provided as input to the decoder ϕ. The decoder comprises three fully connected layers with dimensions $(24 + 3, 32)$, $(32, 64)$, and $(64, 2)$, respectively, and with ReLU activation for the first two layers. The decoder predicts the trajectory of the on-ramp vehicle for a horizon of $H = 1$ time step. A comparison of the predicted and the actual trajectories for three on-ramp vehicles is plotted in Fig. 4. Each column in this plot corresponds to a single vehicle, where the plots in the first row are a comparison of the position in the direction perpendicular to the highway (labeled as lateral position), and in the second row, we plot the position along the direction of the highway (labeled as longitudinal position). The AIS model performs well in predicting the longitudinal positions with root mean square error (RMSE) of 0.35 m for 4(a), 0.39 m for 4(b), and 0.54 m for 4(c). It is also capable of closely predicting the longitudinal positions with RMSE of 2.62 m for 4(a), 1.92 m for 4(b), and 1.73 m for 4(c).

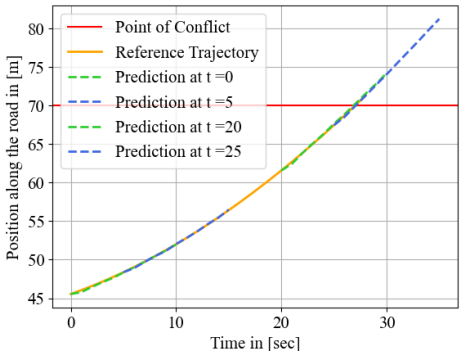
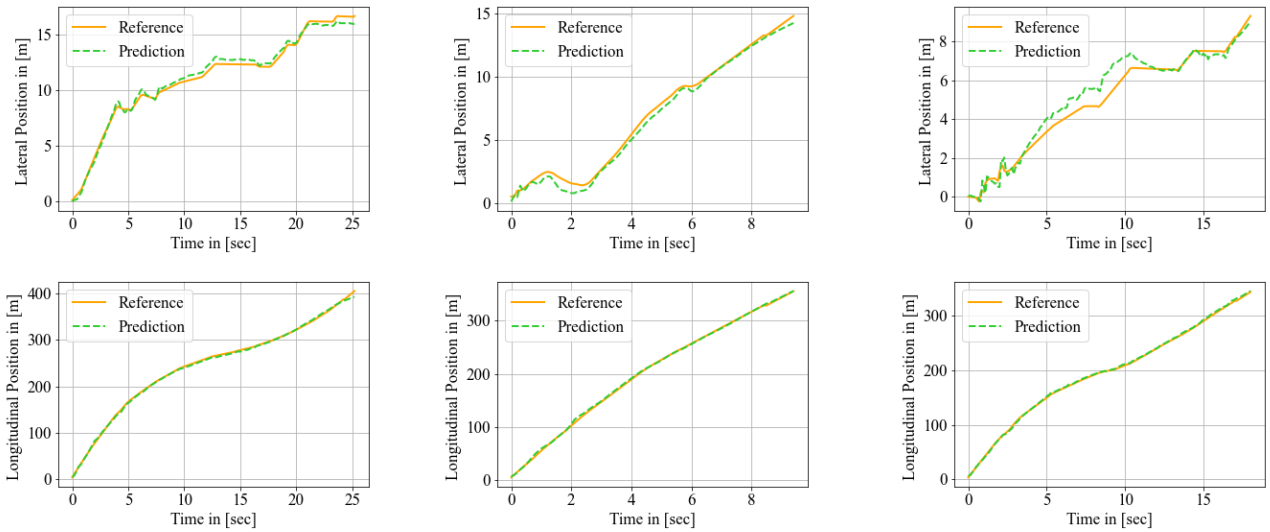


Fig. 3: Predicted v/s actual trajectories for generated data.

2) The IRL dataset models follow a similar architecture to the NGSIM based model. We use identical networks across the two generated datasets. At each time t, input to the encoder ψ is the most recent 2 dimensional state and the action of both CAV–1 and HDV–2, which is a 6 dimensional vector. The encoder structure is given by $(6, 8)$, $(8, 16)$ with ReLU activation followed by a GRU with a 4 dimensional hidden state representing the AIS. Since we consider only two vehicles in section II, a 4 dimensional AIS is sufficient.The most recent action of CAV–1 along with the AIS is the input to the decoder ϕ. The decoder comprises three layers of dimensions $(4 + 1, 2)$, $(2, 4)$ with ReLU activation, and $(4, H)$, where H is the length of the horizon. We compare the predicted and refernce trajectories in Fig. 3 at different time instances $t = 0, 5, 20, 25$ sec. The RMSE between all predicted and reference trajectories is 0.17 m. For these datasets, we set the length of the control zone $L_c = 70$ m and indicate the conflict point using a red horizontal line.

B. Iterative MPC Implementation

To solve the MPC problem for the control input $u_{1,t}$, we can consider the AIS predictions as a constraint. However, the complexity of the neural network model creates com-



(a) (b) (c)

Fig. 4: Trajectory prediction v/s the reference trajectory from the NGSIM dataset for three on-ramp cars (a), (b), and (c).

Algorithm 1 Iterative MPC Implementation

Require: t, $H \in \mathbb{N}$, $j_{\max} \in \mathbb{N}$, $u_{1,t}^{(0)} = 0$

1: Predict the AIS by $s_t = \psi(s_{t-1}, x_t, u_{t-1})$
2: **for** $j = 1, 2, \ldots, j_{\max}$ **do**
3: Predict states and control actions of HDV–2 by $y_{t+1:t+H}^{(j)} = \phi(s_t, u_{1,t}^{(j-1)})$
4: Solve (4) to obtain $u_{1,t:t+H-1}^{(j)}$
5: **return** $u_{1,t}^{(j_{\max})}$

putational intractability. Therefore, we propose an iterative MPC implementation in Algorithm 1 to sequentially compute the neural network prediction and solve the MPC problem. At each time step, we initialize $u_{1,t}^{(0)} = 0$ and compute the prediction of the AIS by $s_t = \psi(s_{t-1}, x_t, u_{t-1})$ where the inputs include the previous AIS s_{t-1}, current states x_t, and previous control inputs u_{t-1}. Next, at each iteration j, we predict states and actions of HDV–2 by $y_{t+1:t+H}^{(j)} = \phi(s_t, u_{1,t}^{(j-1)})$ that use the AIS s_t and the control input $u_{1,t}^{(j-1)}$ of the MPC in the last iteration. Then we solve the MPC problem given the prediction $y_{t+1:t+H}^{(j)}$ to obtain the new inputs $u_{1,t}^{(j)}$. This procedure is repeated until a maximum number of iterations $j_{\max}$.

Remark 2. While Algorithm 1 performs well for finite iterations in simulation, a formal proof of convergence needs to be established in future work to ensure stability.

C. Simulation Results

We conduct simulations in Python programming language in which CasADi [20] and the built-in IPOPT solver [21] are used for formulating and solving the MPC problem, respectively. The parameters of MPC are: $\Delta T = 0.2\,\mathrm{s}$, $H = 10$, $v_{\min} = 0.0\,\mathrm{m/s}$, $v_{\max} = 14.0\,\mathrm{m/s}$, $u_{\min} = -3.0\,\mathrm{m/s^2}$, $u_{\max} = 2.0\,\mathrm{m/s^2}$, $\omega_1 = 1.0$, $\omega_2 = 10.0$, $\omega_3 = 10^3$, $\rho = 1.0\,\mathrm{s}$.

TABLE I: Performance comparison safe v/s exploratory.

Reaction Time Delay (ρ)	Safe	Exploratory
0.6	4984 (99.6%)	4578 (91.5%)
0.8	4996 (99.9%)	4793 (95.8%)
1.0	5000 (100%)	4971 (99.4%)

Figures 5 and 6 show the trajectories, speeds, and control inputs of CAV–1 and HDV–2 in two simulations with two human driving styles, i.e., an aggressive and a conservative human driver. As shown from the figures, in both simulations, safe distance is guaranteed, but CAV–1 has different merging maneuvers depending on the behavior of HDV–2. If the human driver accelerates to cross first, then CAV–1 slows down to yield, while if the human driver is more conservative, CAV–1 finds it safe to cross before HDV–2.

To evaluate the performance of the MPC with two learned prediction models, we conduct 5000 simulations with different initial conditions of the vehicles and different IRL models for HDV–2 and compare the level of safety under three different values for ρ. Recall from Subsection II-A that CAV–1's conservatism increases with ρ. Hence, we observe that the number of unsafe situations decreases with an increase in ρ. The results are indicated in Table I and show that MPC with the model trained using the safe-strategy dataset achieves a higher level of safety compared to the model trained using the exploratory-strategy dataset. We believe that this might be due to the model for the exploratory dataset requiring further training.

V. Concluding Remarks

In this paper, we presented an approach to learn an AIS model of CAV-HDV interactions for a CAV to maneuver safely during highway merging. We validated the capability of our AIS model by training with and predicting real-life merging scenarios involving human-driven vehicles for the NGSIM repository. Then, we showed that using data

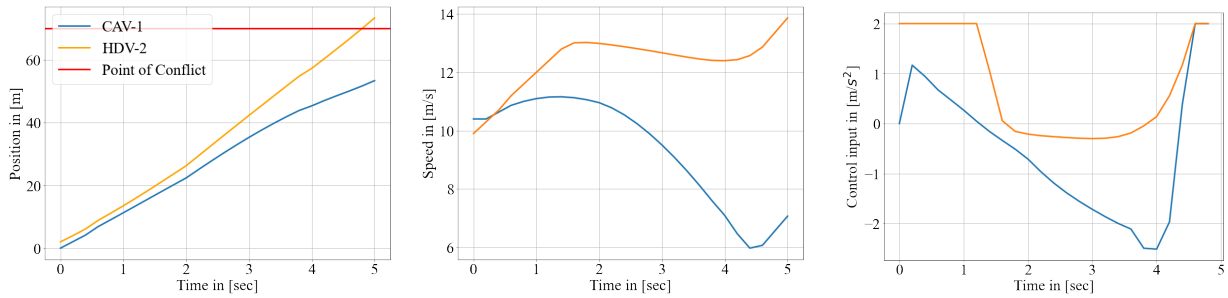


Fig. 5: Trajectories, speeds, and distances of the vehicles in the simulation with an aggressive human driver.

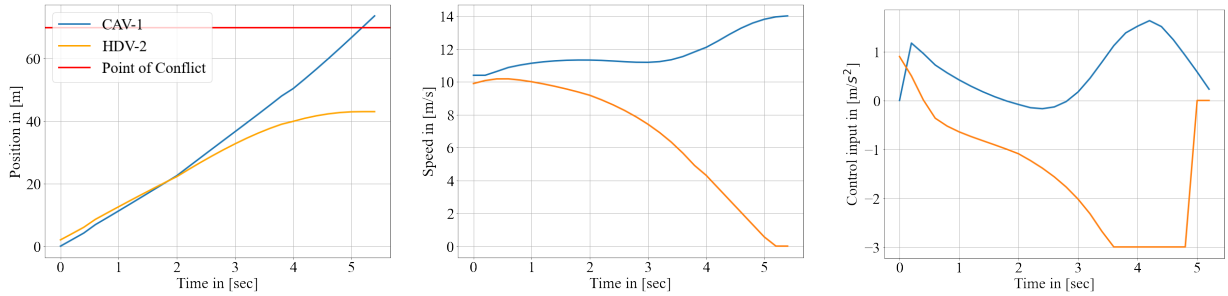


Fig. 6: Trajectories, speeds, and distances of the vehicles in the simulation with a conservative human driver.

generated from an IRL model for a mixed traffic scenario with a CAV and an HDV, an AIS model can be learned to predict future trajectories of the HDV. We proposed an iterative MPC algorithm to generate control actions for the CAV utilizing these predictions and showed using numerical simulations that they are safe against a spectrum of driving behaviors of the HDV. Future research should focus on extending the framework for continuous-time control and deriving safety guarantees for the control algorithm.

References

[1] L. Zhao and A. A. Malikopoulos, "Enhanced mobility with connectivity and automation: A review of shared autonomous vehicle systems," *IEEE Intelligent Transportation Systems Magazine*, vol. 14, no. 1, pp. 87–102, 2022.

[2] A. A. Malikopoulos, L. E. Beaver, and I. V. Chremos, "Optimal time trajectory and coordination for connected and automated vehicles," *Automatica*, vol. 125, no. 109469, 2021.

[3] W. Gao, A. Odekunle, Y. Chen, and Z. Jiang, "Predictive cruise control of connected and autonomous vehicles via reinforcement learning," *IET Control Theory & Applications*, vol. 13, pp. 2849–2855, Nov. 2019.

[4] K. Liu, N. Li, H. E. Tseng, I. Kolmanovsky, A. Girard, and D. Filev, "Cooperation-Aware Decision Making for Autonomous Vehicles in Merge Scenarios," in *2021 60th IEEE Conference on Decision and Control (CDC)*, pp. 5006–5012, IEEE, 2021.

[5] R. Chandra and D. Manocha, "Gameplan: Game-theoretic multi-agent planning with human drivers at intersections, roundabouts, and merging," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2676–2683, 2022.

[6] V.-A. Le and A. A. Malikopoulos, "Optimal weight adaptation of model predictive control for connected and automated vehicles in mixed traffic with bayesian optimization," in *2023 American Control Conference (ACC)*, pp. 1183–1188, IEEE, 2023.

[7] F. Altché and A. de La Fortelle, "An lstm network for highway trajectory prediction," in *2017 IEEE 20th international conference on intelligent transportation systems (ITSC)*, pp. 353–359, IEEE, 2017.

[8] S. H. Park, B. Kim, C. M. Kang, C. C. Chung, and J. W. Choi, "Sequence-to-sequence prediction of vehicle trajectory via lstm encoder-decoder architecture," in *2018 IEEE intelligent vehicles symposium (IV)*, pp. 1672–1678, IEEE, 2018.

[9] N. Deo and M. M. Trivedi, "Multi-modal trajectory prediction of surrounding vehicles with maneuver based lstms," in *2018 IEEE intelligent vehicles symposium (IV)*, pp. 1179–1184, IEEE, 2018.

[10] Z. el abidine Kherroubi, S. Aknine, and R. Bacha, "Novel decision-making strategy for connected and autonomous vehicles in highway on-ramp merging," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 12490–12502, 2021.

[11] J. Guo, S. Cheng, and Y. Liu, "Merging and diverging impact on mixed traffic of regular and autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 3, pp. 1639–1649, 2020.

[12] A. A. Malikopoulos, "Separation of learning and control for cyber-physical systems," *Automatica*, vol. 151, no. 110912, 2023.

[13] A. A. Malikopoulos, "On team decision problems with nonclassical information structures," *IEEE Transactions on Automatic Control*, vol. 68, no. 7, pp. 3915–3930, 2023.

[14] J. Subramanian, A. Sinha, R. Seraj, and A. Mahajan, "Approximate information state for approximate planning and reinforcement learning in partially observed systems," *Journal of Machine Learning Research*, vol. 23, no. 12, pp. 1–83, 2022.

[15] A. Dave, N. Venkatesh, and A. A. Malikopoulos, "Approximate information states for worst-case control of uncertain systems," in *Proceedings of the 61th IEEE Conference on Decision and Control (CDC)*, pp. 4945–4950, 2022.

[16] A. Dave, N. Venkatesh, and A. A. Malikopoulos, "Approximate Information States for Worst-Case Control and Learning in Uncertain Systems," *arXiv:2301.05089 (in review)*, 2023.

[17] L. Yang, K. Zhang, A. Amice, Y. Li, and R. Tedrake, "Discrete approximate information states in partially observable environments," in *2022 American Control Conference (ACC)*, pp. 1406–1413, IEEE, 2022.

[18] M. Fatemi, T. W. Killian, J. Subramanian, and M. Ghassemi, "Medical dead-ends and learning to identify high-risk states and treatments," *Advances in Neural Information Processing Systems*, vol. 34, pp. 4856–4870, 2021.

[19] U. S. D. of Transportation Federal Highway Administration, "Next generation simulation (ngsim) vehicle trajectories and supporting data," 2016.

[20] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi: a software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.

[21] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, no. 1, pp. 25–57, 2006.