Adaptive Autopilot: Constrained DRL for Diverse Driving Behaviors

Adaptive Autopilot: Constrained DRL for Diverse Driving Behaviors

Dinesh Cyril Selvaraj*, Christian Vitale†, Tania Panayiotou†, Panayiotis Kolios‡,
Carla Fabiana Chiasserini* and Georgios Ellinas†
*CARS@Polito, Politecnico di Torino, Torino, Italy
†KIOS CoE and Dept. of Electrical and Computer Eng., University of Cyprus, Nicosia, Cyprus
‡KIOS CoE and Dept. of Computer Science, University of Cyprus, Nicosia, Cyprus

***Abstract*—In pursuit of autonomous vehicles, achieving human-like driving behavior is vital. This study introduces adaptive autopilot (AA), a unique framework utilizing constrained-deep reinforcement learning (C-DRL). AA aims to safely emulate human driving to reduce the necessity for driver intervention. Focusing on the car-following scenario, the process involves: (1) extracting data from the highD natural driving study, categorizing it into three driving styles using a rule-based classifier; (2) employing deep neural network (DNN) regressors to predict human-like acceleration across styles; (3) using C-DRL, specifically the soft actor-critic Lagrangian technique, to learn human-like safe driving policies. Results indicate effectiveness in each step, with the rule-based classifier distinguishing driving styles, the regressor model accurately predicting acceleration, outperforming traditional car-following models, and C-DRL agents learning optimal policies for human-like driving across styles.**

***Index Terms*—ITS, Adaptive cruise control, Constrained reinforcement learning, Connected vehicles**

1. Introduction

In recent years, the automotive industry has experienced a digital transformation, enhancing vehicles with sensing devices, electronic control units, and advanced driver assistance algorithms, including features like blind-spot detection and adaptive cruise control (ACC) [1]. This evolution aims to improve safety, traffic efficiency, and the overall travel experience [2]. However, consumer adoption relies on trust in automated systems and considerations on legal issues [3].

The acceptance of these systems is also influenced by their ability to emulate human-like driving styles [4], [5]. Toward this, distinct driver categories, like aggressive drivers prioritizing smaller gaps with abrupt maneuvers and conservative drivers favoring larger gaps with smoother behavior [6], require tailored controllers. Current car-following controllers [7]–[10], despite attempts to differentiate among driving styles, depend on predefined parameters, lacking real-world adaptability. The remedy lies in data-driven controllers utilizing real-world data, emulating diverse driving styles, and, theoretically, having the potential to reduce disengagement rates [4]. In this context, machine learning (ML) plays a pivotal role in developing models capable of making informed decisions by analyzing complex and multi-variate data. Among ML paradigms, reinforcement learning (RL) is well-suited for this intricate task [11], [12], as RL agents learn by interacting with the environment through a trial-and-error mechanism, aiming to maximize cumulative rewards. However, traditional RL agents often overlook safety constraints critical for real-world applications like autonomous driving. C-DRL addresses this limitation as, unlike traditional RL, it incorporates constraints through cost functions, ensuring safe driving by minimizing them during the learning process [13].

Building on C-DRL, our work introduces the adaptive autopilot (AA) framework. This framework employs a C-DRL approach to effectively accommodate diverse driving styles by integrating rewards based on a human-like acceleration predictor, alongside constraints to enforce a minimum headway among vehicles. The three main steps of the framework are: (i) categorizing real-world driving data from the highD dataset [14] into aggressive, normal, and conservative styles, (ii) training deep neural network-based regressors to predict human-like vehicle acceleration tailored to each driving behavior, and (iii) implementing the C-DRL framework to take human-like safe actions. The trained agents, corresponding to each driving style and based on the soft actor-critic Lagrangian algorithm [15], are validated using real-world driving data from the highD dataset. Results demonstrate the framework's ability to drive the vehicle in line with corresponding human drivers under different styles, with the headway trend highlighting the prioritization of safety constraints. To summarize, the main contributions of this work are as follows:

- *(i)* Real-world driving data is classified into aggressive, normal, and conservative styles using a rule-based approach. Separate neural network-based regressors are then trained for each style to predict human-like vehicle accelerations.
- *(ii)* A novel C-DRL framework is introduced, adapting vehicle acceleration to different driving styles by (safely) mimicking human drivers. This is achieved through minimizing the difference between C-DRL and predicted human actions at each step. Further, a headway-based safety constraint is imposed during training, where multiple real-world driving traces are used to enhance generalization.
- *(iii)* Performance results demonstrate the proposed framework's ability to adapt to diverse driving styles while adhering to safety constraints.

In the remainder of the paper, Sec. 2 describes related research, Sec. 3 introduces the AA framework, Sec. 4 discusses the obtained results, and Sec. 5 presents concluding remarks.

2. Related Work

While commercial ACC systems were introduced in the early 2000s to enhance safety and driving experience [16], they still offer limited customization options with few user-defined parameters like desired gap and velocity. The rigidity of these systems hampers their ability to accurately replicate human driving styles, leading to reduced trust and increased instances of driver intervention, thereby affecting safety benefits. Various research directions [12], [17], [18] have been explored to address these limitations and enhance ACC systems.

One research direction involves car-following (CF) models to provide optimal control actions in response to lead vehicle movements. Relevant models include the Gipps model [10], which prioritizes a safe inter-vehicle distance, incorporating human factors like reaction time and comfort. The intelligent driver model (IDM) [8] considers desired velocity and inter-vehicle distance, using different parameter values for various driving styles [7]. However, these CF models struggle to accurately represent real-world driving behavior due to oversimplification, and their parameters are calibrated for traffic scenarios and safety rather than human-like driving behavior [17]. Our framework, compared to IDM, employs a data-driven approach demonstrating safe and human-like acceleration behavior across different styles.

Another research direction explores data-driven models, optimizing vehicle control using real-world mobility traces. For example, [18] employs a particle swarm optimization with bi-directional long short-term memory (PSO–Bi–LSTM) to enhance IDM model parameters and predict human driving behavior. IDM's learned fixed parameters limit however its ability to accurately model driving behavior. Other works use traditional and recurrent neural networks for acceleration/velocity predictions [19], [20]. Such neural networks face challenges in personalized driver behavior modeling due to training data influences. Similarly, DRL has been utilized [12], [21] for improved car-following behavior, emphasizing safety, traffic efficiency, and comfort. Nevertheless, these DRL approaches focus solely on generic driving behaviors, lacking consideration for a human in their training process to achieve human-like driving behavior.

Finally, the offline human-in-the-loop RL paradigm gains popularity for enhancing RL frameworks' adaptability by incorporating the human factor [22]. This approach does not require real-time human intervention but leverages human experience to shape reward functions. For instance, [23] uses Shanghai naturalistic driving study data to mimic human-like driving behavior by designing reward functions to reduce errors between simulated and empirical values in spacing and velocity. It outperforms traditional neural network models in capturing driver behavior, although safety concerns arise as aggressive human behaviors are replicated without considering safety. Additionally, [24] employs behavior cloning, an imitation learning method, to achieve human-like driving behavior. A major drawback of imitation learning is the accumulation of errors over time, leading to adverse control actions.

To the best of our knowledge, our work is the first to present a comprehensive human-in-the-loop C-DRL framework designed to adapt vehicle driving behavior across diverse driving styles along with safety constraints.

3. Adaptive Autopilot Framework

In this work, the focus is on achieving human-like driving behavior through an Adaptive Autopilot controller designed to accommodate various driving styles while ensuring safe conditions, especially in car-following scenarios. The proposed AA framework addresses three interconnected problems: (i) identifying and classifying the driver's style into aggressive, normal, or conservative using a rule-based approach based on headway, lead vehicle relative velocity, and acceleration (Sec. 3-A); (ii) training a neural network-based regressor to predict human-like control actions, particularly acceleration rates, of the same driving style of the driver (Sec. 3-B); (iii) implementing a C-DRL framework for the controller, considering vehicle states as input and ensuring safety while minimizing the difference between the control action and human-like acceleration predicted by the regressor (Sec. 3-C).

A. Rule-based Classifier

Inspired by [7], we categorize driving styles into aggressive, normal, and conservative. Such classification typically utilizes indicators related to longitudinal movements, including speed, acceleration, headway, relative velocity, as well as steering input and lateral acceleration [6]. Nevertheless, given the focus on car-following scenarios, only longitudinal control-related indicators are employed for style categorization.

Designing a model to accurately classify a driver's entire data trace into a unique driving style is challenging due to potential variations within a driver's behavior. For instance, aggressive drivers may exhibit normal or conservative driving at times. In this work, each control action of the driver is tagged with a specific driving behavior. Subsequently, the ratio of each tagged behavior across the entire trace is calculated to categorize drivers as aggressive, normal, or conservative. Driver actions are tagged with a specific driving behavior based on a rule-based approach, which utilizes headway trends as a key factor in differentiating driving styles. As suggested by [7], aggressive drivers aim to maintain a headway of 1 second or below. Normal drivers aim for headways of around 1.5 seconds, while conservative drivers aim for a headway of 1.8 seconds and above. Based on longitudinal indicators, the classifier's objective is to tag the driver's intention, analyzing how the driver's action will change the headway and toward which of the three goal headways it will lead in the long term.

Considering these aspects, Fig. 1 outlines the hierarchical rule-based classification approach employed in this work, where the leaf nodes represent the assigned driving style. Specifically, the classifier considers information related to both lead and ego vehicles at a given time t: $X(t)=\{\vartheta(t), \nu(t), \ddot{x}_{ego}(t), \ddot{x}_{lead}(t)\}$, representing headway, relative velocity, ego vehicle acceleration, and lead vehicle acceleration, respectively, to classify the behavior:

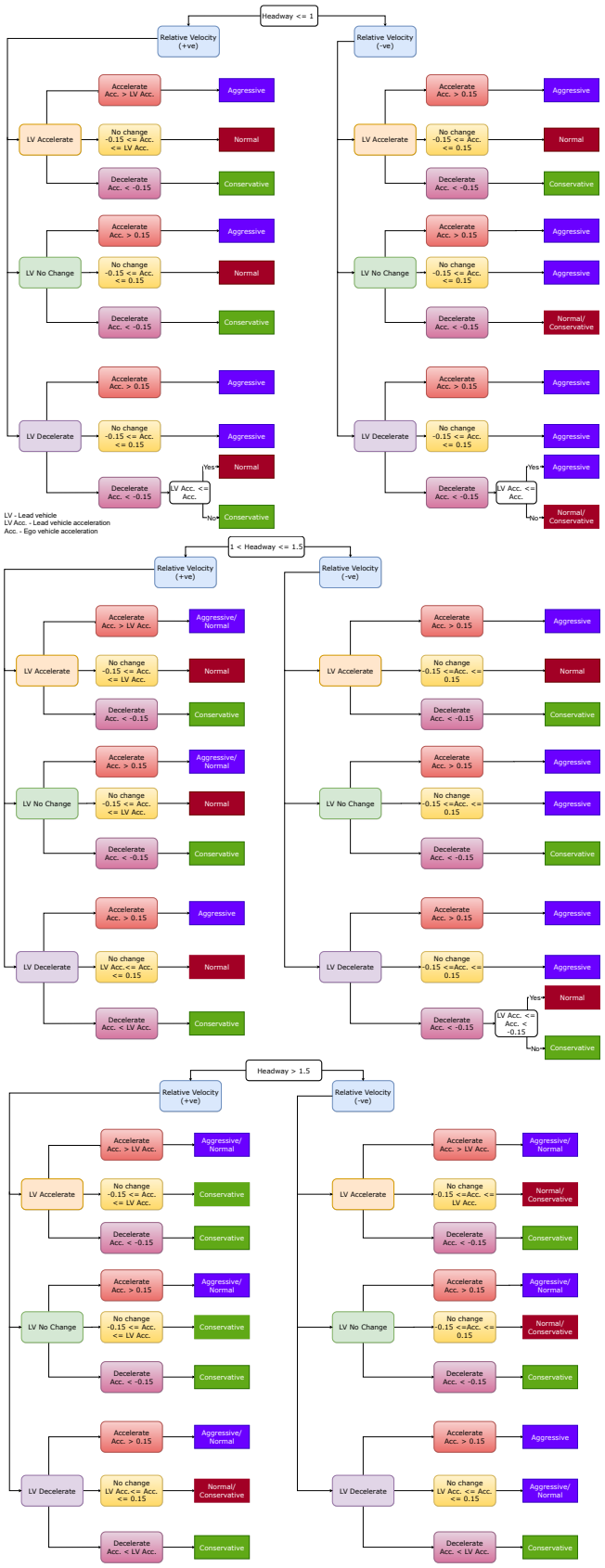

Figure 1. A hierarchical rule-based classifier that labels the driving data into three driving style categories: Aggressive, Normal, and Conservative.

$y(t)=\{Aggressive, Normal, Conservative\}$. Headway and relative velocity are formulated as:

$$\vartheta(t)=\frac{\Delta x_{lead}(t)}{\dot{x}_{ego}(t)} \tag{1}$$

$$\nu(t)=\dot{x}_{lead}(t)-\dot{x}_{ego}(t), \tag{2}$$

where $\Delta x_{lead}(t)=x_{lead}(t)-x_{ego}(t)$ is the relative distance between lead and ego vehicles, and $\dot{x}_{lead}(t)$ and $\dot{x}_{ego}(t)$ represent (resp.) the lead and ego vehicle's speed.

Among the input features, headway serves as the primary criterion for splitting the data into three leaf nodes. Subsequently, for each leaf node, relative velocity becomes a crucial factor in anticipating potential headway changes in subsequent time steps, acting as a secondary criterion for data segmentation. Considering the current headways and relative velocities, the driver's action, specifically the applied acceleration, is categorized into one of the three driving styles. Acceleration differences among the lead and ego vehicles enable an understanding of future changes in relative velocity before they manifest in the data points. This ability facilitates the identification of future achieved headways. Matching future achieved headways with desired headways (as per [7]) provides a straightforward mean to categorize driving actions.

B. Human-like Action Predictor

The human-like action predictor operates as a regressor model, using relevant input data to forecast the vehicle's acceleration. Its purpose is to learn a non-linear function that approximates the relationship between input data and the next vehicle's acceleration. To circumvent relying on past human actions, which might be unavailable in autopilot scenarios as the one of the AA framework, the regressor incorporates both historical and current data related to the lead vehicle, while utilizing only the present ego vehicle data as input. The input dataset consists of the following set of observations:

$$X(t)=\{\ddot{x}_{lead}(t-2\Delta t), \ddot{x}_{lead}(t-\Delta t), \ddot{x}_{lead}(t), \dot{x}_{lead}(t-2\Delta t), \dot{x}_{lead}(t-\Delta t), \dot{x}_{lead}(t), \dot{x}_{ego}(t), \vartheta(t)\}, \tag{3}$$

to obtain prediction $\hat{y}(t)=\ddot{x}_{preg}(t)$ corresponding to the ego vehicle acceleration $y(t)=\ddot{x}_{ego}(t)$, where t, $t-\Delta t$, and $t-2\Delta t$ represent the present and two past time instants, respectively.

In this work, a DNN-based regressor, a deep learning technique, is utilized to predict human-like acceleration values. The highD dataset, [14], serves as the training dataset, following the segmentation into the three driving styles mentioned above, achieved through the rule-based classifier outlined in Sec. 3-A. Hence, a separate model is obtained for each driving style. Throughout the training process, the model is optimized to minimize the mean absolute error (MAE) loss function:

$$L_{mae}=\frac{1}{N}\sum_{i=1}^{N}|y_i-\hat{y}_i|, \tag{4}$$

where N is the number of observations used for MAE loss minimization, $\hat{y}_i$ is the predicted value of the i^{th} observation, and y_i is the actual value of the i^{th} observation.

C. C-DRL Framework

We now introduce our C-DRL framework, inspired by a previously proposed algorithm [15]. The C-DRL framework utilizes pertinent vehicle data as input to decide the vehicle's longitudinal control action, focusing specifically on vehicle acceleration. The control action applied guides the vehicle, earning rewards based on its ability to emulate the desired human-like driving behavior. Moreover, the framework integrates safety constraints to guarantee that the applied control actions maintain a safe distance between vehicles. Fig. 2 provides an overview of the proposed AA framework.

Background: Constrained RL extends traditional RL by introducing constraints on the actions taken by the agent. C-RL is formalized as a constrained Markov decision process (CMDP) [13], an extension of the standard MDP framework. In this form, C-RL is characterized by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \mathcal{C}, b, \gamma)$, representing the state space, action space, transition probabilities, reward, cost function, safety threshold, and discount factor, respectively. The goal of C-RL is to solve the CMDP by learning an optimal policy $\pi : \mathcal{S} \rightarrow \mathcal{A}$ that maximizes the expected cumulative discounted reward while satisfying the constraints. The problem addressed by C-RL is:

$$\begin{aligned} &\max_{\pi:(\mathbf{s}(t),a(t))\sim\rho_\pi} \mathbb{E}\left[\sum_t \gamma^t \mathcal{R}(\mathbf{s}(t), a(t))\right] \\ &\text{subject to:} \quad \mathbb{E}\left[\sum_t \gamma^t \mathcal{C}(\mathbf{s}(t), a(t))\right] \leq b \end{aligned} \tag{5}$$

where ρ_π denotes the trajectory distribution following policy π, and $\mathcal{R}(\mathbf{s}(t), a(t))$ and $\mathcal{C}(\mathbf{s}(t), a(t))$ represent (resp.) the reward and cost functions associated with state $(\mathbf{s}(t))$ and action $(a(t))$ at a specific time step t. As modeling the state transition probabilities for intricate problems can be challenging, in C-RL a model-free approach is typically used, with the relationship between action and reward/cost implicitly learned by interacting with the environment.

In constrained optimization problems, such as Eq. (5), C-RL can utilize an equivalent formulation with Lagrangian multipliers (λ) for optimization. The Lagrangian's saddle point is determined through iterative gradient ascent steps for the policy function π and gradient descent on the Lagrangian multipliers λ [15], [25], [26]. Notably, the gradient step related to λ emphasizes the loss function associated with the constraint. If the constraint is violated, the gradient update increases the multiplier's value, prioritizing the constraint over the reward function, and vice versa. C-DRL advances upon C-RL by incorporating deep neural network-based function approximators to model the policy function $\pi(\mathbf{s}|\theta)$, where θ denotes the neural network parameters. This augmentation significantly improves the C-RL framework's capability to navigate intricate, high-dimensional real-world environments. In this study, we specifically adopt the Soft Actor-Critic Lagrangian (SAC-Lagrangian) technique [15] as the C-DRL methodology to achieve the desired outcome.

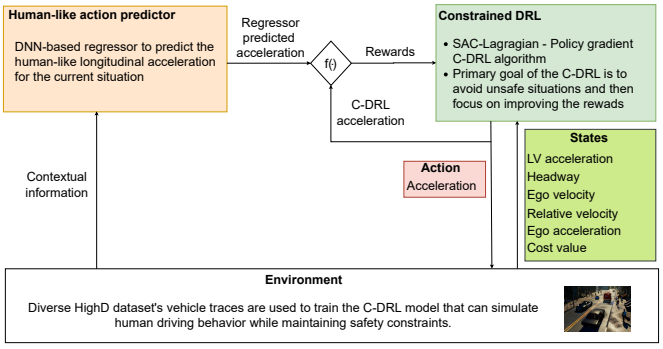

Figure 2. An overview of the proposed AA methodology.

States and Action space: The C-DRL state, $\mathbf{s}(t) \in \mathcal{S}$, represents the vehicle state at any time t and is given by:

$$\mathbf{s}(t) = \{\ddot{x}_{lead}(t), \vartheta(t), \dot{x}_{ego}(t), \nu(t), a(t-\Delta t), \psi(t)\} \tag{6}$$

where $a(t-\Delta t)$ denotes the control action taken by the C-DRL framework at time $t-\Delta t$ and $\psi(t)$ represents the value obtained from the indicator cost function associated with the safety constraint. The cost function is formulated as:

$$\psi(t) = I(\vartheta(t) > \omega), \tag{7}$$

where ω represents the safety threshold for the distance between the vehicles. As mentioned earlier, the action space, $a(t) \in \mathcal{A}$, corresponds to the vehicle's acceleration (a continuous variable bounded within the range $[-4, 4]\,\mathrm{ms}^{-2}$). Additionally, consecutive acceleration values are restricted to vary by no more than $\pm 0.24\,\mathrm{ms}^{-2}$ [27].

Reward components: The reward signal is a scalar value provided by the environment after each action, offering insights into the agent's performance concerning the framework objectives. Our reward function comprises two key components: (i) human similarity reward, which assesses the disparity between C-DRL control actions and human-like actions predicted by the regressor model; and (ii) comfort, ensuring smooth acceleration changes between time steps. The trends of these reward components are illustrated in Fig. 3. Formally, the reward is expressed as:

$$r(\mathbf{s}(t), a(t)) = r_h(\mathbf{s}(t), a(t)) + r_c(\mathbf{s}(t), a(t)), \tag{8}$$

where $r_h(\mathbf{s}(t), a(t))$ and $r_c(\mathbf{s}(t), a(t))$ represent human similarity and comfort rewards at time step t, respectively. The reward components are further detailed below.

Human similarity reward component: This reward component assesses the similarity between the driving behavior of the vehicle and that of a human. It quantifies the disparity between the predicted acceleration values by the DNN regressor and the ones applied by C-DRL, encouraging the agent to minimize this difference. Specifically, the function offers a reward that is maximum (+1) for zero error and decreases significantly as the difference between predictions increases. The reward formulation incorporates a $\tanh$ function for this purpose:

$$r_h(\mathbf{s}(t), a(t)) = 2 \cdot F_h + 1, \quad \text{with} \tag{9}$$

$$F_h = \tanh(-2 \cdot \xi(t)), \tag{10}$$

$$\xi(t) = |a(t) - \ddot{x}_{preg}(t)|\,. \tag{11}$$

Comfort reward component: Sudden acceleration changes can lead to discomfort for passengers. To address this, the comfort reward component considers the rate of change of acceleration with time, known as jerk ($j(t)$). The reward function is designed to decrease gradually as the absolute jerk value increases, ranging from a maximum reward value of 0 to a minimum of -1. This desired reward trend is crafted using a curve-fitting function, specifically a 4PL model, as illustrated in Fig. 3 (right), depicting the comfort reward trend.

Figure 3. Human similarity (left) and comfort (right) reward trend.

Simulation environment: A straightforward car-following simulation environment is created to replicate vehicle movements, enabling the C-DRL agent to learn the desired behavior. Utilizing the highD dataset, the simulation environment incorporates movements from the dataset for the lead vehicle, while simulating the ego vehicle's motions using a linear motion model. The C-DRL agent's predicted acceleration serves as the control action to drive the vehicle, with a set sampling interval of $\Delta t{=}80\,\text{ms}$. The ego vehicle movements follow:

$$\begin{aligned} \dot{x}_{ego}(t{+}\Delta t){=}\dot{x}_{ego}(t){+}a(t)\Delta t, \\ x_{ego}(t{+}\Delta t){=}x_{ego}(t){+}\dot{x}_{ego}(t)\Delta t{+}0.5a(t)\Delta t^2\,. \end{aligned} \tag{12}$$

Learning process: In C-DRL, the exploration-exploitation process is crucial, involving a balance between trying new actions and exploiting actions with high rewards. Initial training stages necessitate thorough exploration of the action space to discover those maximizing cumulative rewards. However, if action values are restricted, hindering exploration, the agent may miss identifying actions leading to higher rewards. To mitigate this, we adopt a curriculum learning approach [28], [29], gradually increasing difficulty during training. Initial episodes allow unrestricted changes in subsequent actions, with limits introduced once the agent learns the desired behavior. Training utilizes multiple driver traces from the highD dataset for each driving style to ensure generalization, and it continues until satisfactory and stable rewards are achieved.

4. Performance Evaluation

In this section, we introduce the dataset we used for training and evaluating our proposed framework, and present the performance of our solution.

A. Dataset

This work utilizes the highD dataset, comprising vehicle trajectories recorded via a drone on German highways at six locations, each covering 420 meters [14]. The dataset consists of $110,500$ vehicle trajectories across 60 recordings, with an average recording length of 17 minutes, encompassing free-driving, car-following, and lane-changing events. To focus on the car-following scenario, vehicle traces were filtered based on criteria including duration (minimum $10\,\text{s}$ of data), absence of lane changes, consistent lead vehicle, minimum speed ($6\,\text{ms}^{-1}$), and vehicle type classification. The sampling frequency used in this work is $\Delta t{=}80\,\text{ms}$. Among the 60 recordings, 32 were selected for pre-processing, resulting in approximately 2.6 million rows of data, balancing accuracy in training regressor models with computational efficiency.

B. Performance Results: Rule-based Classifier

Here we showcase the performance of the rule-based classifier on the highD dataset. Based on the rules presented in Sec. 3-A, the data points are classified into three categories: Aggressive, Normal, and Conservative. Fig. 4 depicts the key characteristics, in terms of longitudinal acceleration and time headway, of each category using this rule-based setup. Overall, aggressive, normal, and conservative driving behaviors comprise 924k, 1.4M, and 863k data points, respectively, with some data double-tagged because certain behaviors coincide with more than one driving style.

The performance results of the rule-based classifier are consistent with expectations, showing large differences between driving styles. Looking at the probability density function (PDF) (Fig. 4 (top)) of the applied longitudinal acceleration, conservative drivers tend to brake to increase the distance from the lead vehicle. Specifically, the PDF mode is at $-0.2\,\text{ms}^{-2}$ and 85% of the conservative actions represent a braking action. On the contrary, aggressive drivers aim to close the gap with the lead vehicle as much as possible (the PDF mode is at $0.2\,\text{ms}^{-2}$ and 73% of the aggressive actions represent acceleration). Normal driving follows a hybrid pattern, with the PDF mode at around $0\,\text{ms}^{-2}$.

Further, although classification happens based on the driver's intention to change its headway, the headway PDF plots (Fig. 4 (bottom)) show that the mode behavior of a specific driving style corresponds to the ones envisioned in [7] (i.e., the PDF mode of aggressive drivers just below the 1-s mark, of normal drivers between 1 and 1.5 s, and of conservative drivers just before the 2-s mark).

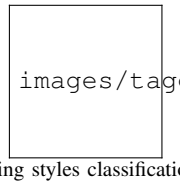

Figure 4. Analysis of driving styles classification using the highD dataset.

Table I
DNN REGRESSOR'S HYPERPARAMETER VALUES

	Driving style		
	Aggressive	Normal	Conservative
Number of hidden units (3 layers)	256, 128, 64	256, 256, 128	256, 128, 64
Dropout rate between hidden layers		0.2, 0.15, 0.1	
Learning rate		0.0001	
Batch size	32	64	64

C. Performance Results: Acceleration Prediction

This section discusses the performance of the regressor models corresponding to the three driving behaviors. As mentioned earlier, we employed a traditional DNN regressor to train the models for predicting vehicle acceleration based on the input data. Using the categorized data obtained from the rule-based classifier, each driver behavior dataset was divided into training (65%), validation (15%), and testing (20%) sets. Given the varied nature of the input, we employed a standardization technique to scale the input features, with zero mean and standard deviation equal to one to help the training. Also, to mitigate overfitting, an early stopping technique was employed to stop training if the error improvement was less than 0.001 in the validation set for five epochs. It should be noted that this section presents the best configurations for each model after extensive hyperparameter trials (Tab. I).

MAE was used to compare the results obtained during inference. In Tab. II, the results obtained by the proposed DNN regressor model are compared with the well-known car-following algorithm, IDM, with parameters suggested in [7]. Additionally, the IDM model was enhanced to match as closely as possible the highD dataset. The fixed parameters used in [7] were modified so as to obtain the best possible fitting with the dataset data points, i.e., the IDM fixed parameters were selected so that the MAE was minimized. The results obtained demonstrate that the proposed regressor models outperform the car-following algorithms for all driving styles. Further, analyzing the CDF of the MAE (not presented in the paper due to space limitations), the optimal performance of the DNN predictor is also showcased by the fact that the absolute error of the prediction, i.e., $|\ddot{x}_{ego}(t) - \ddot{x}_{pred}(t)|$, is less than $0.21\,\mathrm{ms}^{-2}$ in 80% of the data points for all driving styles.

Table II
MEAN ABSOLUTE PREDICTION ERRORS

Driving style	DNN	IDM	IDM-GA
Aggressive	0.1356	2.0357	0.3936
Normal	0.1413	2.4309	0.4584
Conservative	0.1415	4.3752	0.6151

To assess the predictor's performance on individual driver traces, Fig. 5 displays three traces representing aggressive (top), normal (middle), and conservative (bottom) driving behaviors. To evaluate their long-term performance, for both the DNN predictor and the benchmarks, the vehicles in the traces are moved by applying the predicted accelerations using the motion model in Eq. 12. That is, while the lead vehicle traces correspond to those in the highD dataset, the ego vehicle trace disregards the driver's applied accelerations but incorporates the acceleration predicted by the DNN and benchmarks.

Despite some deviations, the DNN-based predicted acceleration closely matches the actual driver behavior and outperforms existing benchmarks. The slight discrepancies observed can be attributed to the DNN predictor leveraging data from thousands of drivers to learn how to predict acceleration in specific situations, while individual driver styles may vary slightly even within the same driving behavior. Although space constraints prevent us from presenting it, applying the wrong driving style's DNN regressor in Fig. 5 would yield significantly different results, with vehicle headways consistently diverging from the true values experienced by the drivers.

Table III
C-DRL HYPERPARAMETER VALUES

	SAC-Lagragian
Number of hidden layers (actor, critic)	3, 2
Number of hidden units (actor)	128, 256, 128
Number of hidden units per layer (critic)	128
Learning rate	0.0003
Replay buffer size	1,000,000
Mini-batch size	128
Discount factor	0.99
Number of random exploration episodes	100
Number of transitions between updates	5
Constraint threshold	0.1

D. Performance Results: Human-like Driving

This section discusses the results of the C-DRL models, aiming to mimic human behavior safely. One model was trained for each driving style, with hyperparameter values similar to those in SAC-Lagrangian [15] (except for the differences listed in Tab. III). Diverse traces were used for each driving style to ensure learning generalized behavior. For each agent, a driving trace was randomly selected from the ones chosen for training for each episode.

The safety objective is to maintain a minimum headway of $\omega{=}1\,\mathrm{s}$ between two vehicles. Hence, the final accelerations decided by the C-DRL agent must ensure that the two vehicles are never closer than this minimum headway. To achieve this objective, during training, the agent primarily focuses on finding a policy that minimizes the cost function representing the safety constraint. After ensuring the safety constraint is met, the agent tries to maximize the reward function, aiming to find a comfortable acceleration profile that mimics human-like driving behavior. Fig. 6 presents the evolution during training of the rewards and the weights assigned to the cost (λ) and reward ($1{-}\lambda$) functions for the three agents. Specifically, the first row shows the reward trend of the evaluation episodes during training (executed every 100 training episodes), which is used to assess the training progress. As depicted, the reward grows and stabilizes as training progresses. The normal and

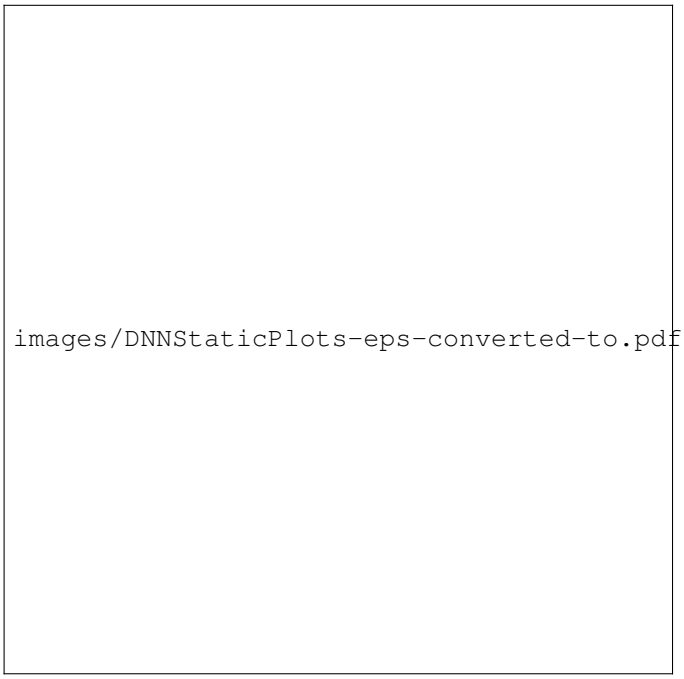

Figure 5. Driver-wise regressor model predictions for the three driving styles.

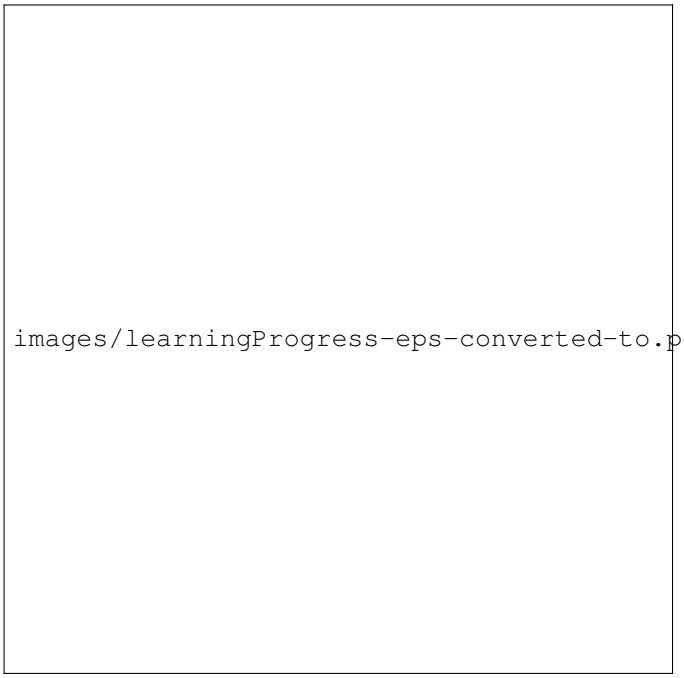

Figure 6. C-DRL training: reward trend (top) and reward vs constraint importance (bottom) for Aggressive, Normal, and Conservative driving.

conservative agents could achieve higher rewards than the aggressive agent because the aggressive agent had to prioritize the safety constraint cost function before maximizing rewards (second row of Fig. 6). The conservative and normal driving behavior agents give more importance to the rewards, as the corresponding agents would not breach the safety constraint (according to their driving style). However, for aggressive drivers, who would naturally drive the headway below the 1-sec mark, the optimal cost function weight λ is not equal

Figure 7. Inference phase: C-DRL - Aggressive driving style.

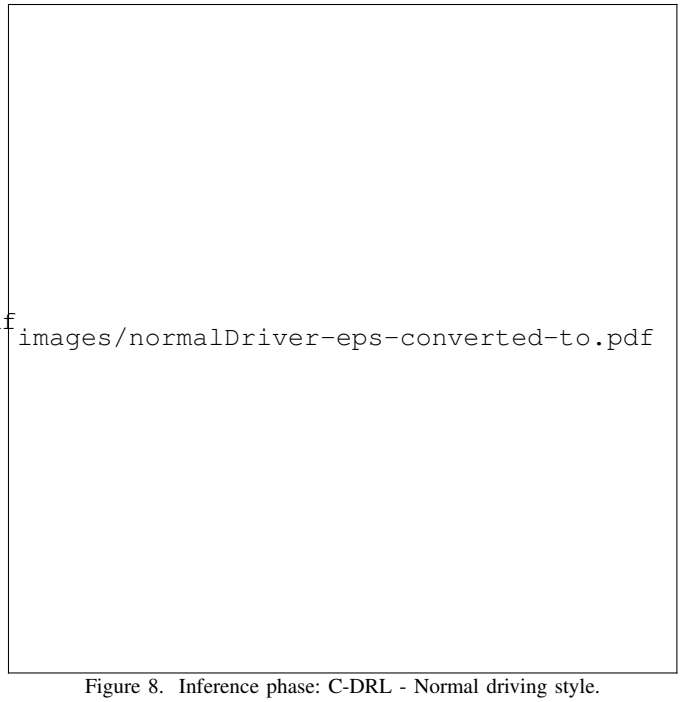

Figure 8. Inference phase: C-DRL - Normal driving style.

to zero. This indicates that maximum reward maximization would fail to respect the safety constraint, which is undesirable. Hence, the final agent trades reward maximization for enhanced safety. To test the performance of each agent, we selected the best-performing model from the evaluation episodes for the inference phase.

During inference, the agents are tested on driving traces that were not used during training. Figures 7–9 illustrate the agent's performance across the three driving styles, each for

a specific trace (with similar results obtained for all tested traces). Fig. 7 demonstrates that the aggressive agent could safely drive the vehicle, maintaining the headway around the 1-s mark without violating the safety threshold and also imitating the acceleration predicted by the regressor model whenever possible. When the headway drops below the safety threshold, the agent starts braking smoothly to maintain a safe distance from the lead vehicle. Once the agent has successfully satisfied the safety constraint, its focus shifts to mimicking the driver's behavior, as depicted by the acceleration trend (Fig. 7 (top right)), closely following the DNN regressor model predictions. This is also confirmed by the human similarity reward trend (Fig. 7 (bottom left)). To emphasize the importance of the C-DRL approach, we compared the proposed framework with a non-constrained DRL technique (referred to as Ego_DRL), where we excluded the cost indicator function during training, confirming that without the safety constraint, the aggressive driving style could lead to unsafe headway, potentially resulting in dangerous situations.

In the normal and conservative driving styles, where the safety constraint's role is not crucial, the agents effectively mimicked human driving behavior by following the DNN regressor-predicted accelerations (Figures 8 and 9). To quantitatively evaluate the results, we calculated the root mean square error between the regressor-predicted human-like acceleration and the C-DRL predicted acceleration, resulting in error values of 0.282, 0.043, and 0.013 for aggressive, normal, and conservative driving behavior, respectively. As anticipated, aggressive driving behavior yields a higher magnitude of error due to safety constraints, while errors for normal and conservative driving behaviors remain minimal. Additionally, it is noteworthy that, although slight discrepancies exist between the DNN regressor and the actual human-applied acceleration, the overall headway profiles generated by the C-DRL agents consistently align closely with those observed in the dataset.

5. Conclusion

We presented an adaptive autopilot framework utilizing C-DRL to drive vehicles similarly to human drivers, adapting to diverse driving styles. The adaptive autopilot framework tackles three interconnected sub-problems: identifying driving styles using real-world data through a rule-based approach, predicting human-like acceleration across different driving styles using a DNN regressor model, and proposing a C-DRL approach to drive vehicles while considering safety constraints and mimicking human-like behavior. Results indicate the regressor model can safely mimic human-like driving behavior effectively and outperforms state-of-the-art IDM models in predicting acceleration. Hence, the comfortable experience provided by the proposed adaptive autopilot has the potential to enhance the satisfaction of human drivers, leading to a reduced disengagement rate of the autopilot driving system. Future work includes leveraging semi-supervised learning for enhanced driving style categorization and extending the framework to realistic vehicle dynamics-based simulation environments for handling complex scenarios like cut-ins and lane changes.

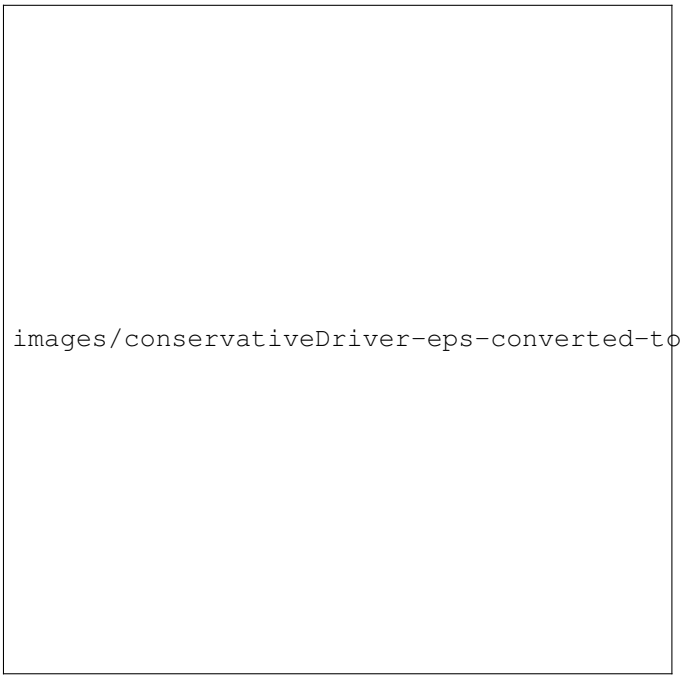

Figure 9. Inference phase: C-DRL - Conservative driving style.

Acknowledgments

This work was supported by the EU's H2020 research and innovation programme under grant agreement No. 739551 (KIOS CoE - TEAMING) and under grant agreement No. 101069688 (CONNECT project), from the Republic of Cyprus through the Deputy Ministry of Research, Innovation and Digital Policy.

References

[1] E. Yurtsever et al., "A survey of autonomous driving: Common practices and emerging technologies," *IEEE Access*, vol. 8, 2020.

[2] L. Yu and R. Wang, "Researches on adaptive cruise control system: A state of the art review," *Proc. of the Inst. of Mech. Eng., Part D: Journal of Automobile Eng.*, vol. 236, no. 2-3, pp. 211–240, 2022.

[3] M. Kyriakidis et al., "Public opinion on automated driving: Results of an international questionnaire among 5000 respondents," *Transp. Res. Part F Traffic Psych. Behav.*, vol. 32, pp. 127–140, 2015.

[4] Z. Ma and Y. Zhang, "Drivers trust, acceptance, and takeover behaviors in fully automated vehicles: Effects of automated driving styles and driver's driving styles," *Accid. Anal. Prev.*, vol. 159, 2021.

[5] Z. Ma and Y. Zhang, "Investigating the effects of automated driving styles and driver's driving styles on driver trust, acceptance, and take over behaviors," in *Proc. Hum. Factors Ergon. Soc. Annu. Meet.*, 2020.

[6] F. Sagberg et al., "A review of research on driving styles and road safety," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 57, no. 7, pp. 1248–1275, 2015.

[7] A. Kesting et al., "Agents for traffic simulation," *arXiv:0805.0300 [physics.soc-ph]*, 2008.

[8] M. Treiber and A. Kesting, "Modeling human aspects of driving behavior," in *Traffic Flow Dynamics: Data, Models and Simulation*. Springer Berlin Heidelberg, 2013, pp. 205–224.

[9] S. Krauss, "Microscopic modeling of traffic flow: investigation of collision free vehicle dynamics," *PhD Thesis*, Univ. of Cologne, 1998.

[10] P. Gipps, "A behavioural car-following model for computer simulation," *Transp. Res. Part B Method.*, vol. 15, no. 2, 1981.

[11] B. Kiran et al., "Deep reinforcement learning for autonomous driving: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, 2022.

[12] D. C. Selvaraj et al., "An ML-aided reinforcement learning approach for challenging vehicle maneuvers," *IEEE Trans. Intell. Veh.*, vol. 8, no. 2, pp. 1686–1698, 2023.

[13] E. Altman, *Constrained Markov Decision Processes: Stochastic Modeling*. Routledge, 2021.

[14] R. Krajewski et al., "The highD dataset: A drone dataset of naturalistic vehicle trajectories on German highways for validation of highly automated driving systems," in *Proc. IEEE ITSC*, 2018.

[15] J. Roy et al., "Direct behavior specification via constrained reinforcement learning," *arXiv:2112.12228 [cs.LG]*, 2021.

[16] T. Watanabe et al., "Development of an intelligent cruise control system," in *Steps Forward. Intell. Transp. Syst. World Congr.*, 1995.

[17] T. Zhang et al., "Car-following models: A multidisciplinary review," *arXiv:2304.07143 [eess.SY]*, 2023.

[18] V. Papathanasopoulou and C. Antoniou, "Towards data-driven car-following models," *Transp. Res. Part C Emerg. Techno.*, vol. 55, 2015.

[19] A. Khodayari et al., "A modified car-following model based on a neural network model of the human driver effects," *IEEE Trans. Syst. Man. Cybern. - Part A: Syst. Hum.*, vol. 42, no. 6, 2012.

[20] X. Wang et al., "Capturing car-following behaviors by deep learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 3, pp. 910–920, 2018.

[21] M. Zhu et al., "Safe, efficient, and comfortable velocity control based on reinforcement learning for autonomous driving," *Transp. Res. Part C Emerg. Techno.*, vol. 117, p. 102 662, 2020.

[22] H. Liang et al., "Human-in-the-loop reinforcement learning," in *Proc. IEEE CAC*, 2017, pp. 4511–4518.

[23] M. Zhu et al., "Human-like autonomous car-following model with deep reinforcement learning," *Transp Res Part C Emerg Tech*, vol. 97, 2018.

[24] Y. Tian et al., "Learning to drive like human beings: A method based on deep reinforcement learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 6357–6367, 2022.

[25] S. Ha et al., "Learning to walk in the real world with minimal human effort," *arXiv:2002.08550 [cs.RO]*, 2020.

[26] Q. Yang et al., "WCSAC: Worst-case soft actor critic for safety-constrained reinforcement learning," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 10 639–10 646.

[27] D. Salles et al., "Extending the intelligent driver model in SUMO and verifying the drive off trajectories with aerial measurements," in *SUMO Conf. Proc.*, 2022, pp. 1–25.

[28] G. Hacohen and D. Weinshall, "On the power of curriculum learning in training deep networks," *arXiv:1904.03626 [cs.LG]*, 2019.

[29] D. Markudova et al., "ReCoCo: Reinforcement learning-based congestion control for real-time applications," in *Proc. IEEE HPSR*, 2023.