Prediction-aware and Reinforcement Learning based Altruistic Cooperative Driving

Rodolfo Valiente[1], Mahdi Razzaghpour[1], Behrad Toghi[1], Ghayoor Shah[1], Yaser P. Fallah[1]

Abstract— Autonomous vehicle (AV) navigation in the presence of Human-driven vehicles (HVs) is challenging, as HVs continuously update their policies in response to AVs. In order to navigate safely in the presence of complex AV-HV social interactions, the AVs must learn to predict these changes. Humans are capable of navigating such challenging social interaction settings because of their intrinsic knowledge about other agents' behaviors and use that to forecast what might happen in the future. Inspired by humans, we provide our AVs the capability of anticipating future states and leveraging prediction in a cooperative reinforcement learning (RL) decision-making framework, to improve safety and robustness. In this paper, we propose an integration of two essential and earlier-presented components of AVs: social navigation and prediction. We formulate the AV's decision-making process as a RL problem and seek to obtain optimal policies that produce socially beneficial results utilizing a prediction-aware planning and social-aware optimization RL framework. We also propose a Hybrid Predictive Network (HPN) that anticipates future observations. The HPN is used in a multi-step prediction chain to compute a window of predicted future observations to be used by the value function network (VFN). Finally, a safe VFN is trained to optimize a social utility using a sequence of previous and predicted observations, and a safety prioritizer is used to leverage the interpretable kinematic predictions to mask the unsafe actions, constraining the RL policy. We compare our prediction-aware AV to state-of-the-art solutions and demonstrate performance improvements in terms of efficiency and safety in multiple simulated scenarios.

Index Terms—Altruistic Cooperative Driving, Prediction-aware, Reinforcement Learning.

I. Introduction

THE adoption of connected and autonomous vehicles (CAVs) is expected to improve safety and efficiency, decrease traffic accidents, and increase mobility [1]. A necessary step toward the widespread integration of autonomous vehicles (AV) in society is allowing coexistence of safe AVs and human-driven vehicles (HVs). Nevertheless, coordination and cooperation with HVs are still challenging for AVs, particularly in complex social interactions [1]–[3]. In order to experience those benefits and allow real adoption of AVs on the road, AVs should not only perceive and understand the current environment state but also proactively predict their future states and learn to coordinate and influence other agents. The innate capability to anticipate agents' behaviors and use this knowledge to forecast potential future outcomes allows humans to navigate through complex scenarios; therefore, prediction capabilities are a crucial component in creating secure AVs that can be integrated into society. [4], [5].

[1] Connected & Autonomous Vehicle Research Lab (CAVREL), University of Central Florida, Orlando, FL, USA. rvalienter90@knights.ucf.edu

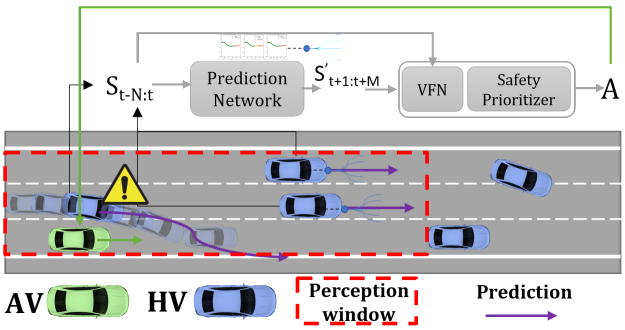

Fig. 1: It is crucial for AV being capable of anticipating future situations of other vehicles using spatial and temporal information. Decision-making by AVs can be enhanced by anticipating the intentions of other agents, which is crucial in complex scenarios and safety-critical situations. The figure depicts an AV (in green) and HVs (in blue) with corresponding predictions within the AVs perception window.

CAVs use Vehicle to Vehicle (V2V) communication to acquire precise situational awareness [6]–[10]. We highlight the advantage of CAVs to improve AVs' robustness and safety in two main directions, first, to overcome the limitations of local sensors, and second, to allow coordination among AVs. An effective and reliable means of communication among agents can facilitate AV-HV coordination. AVs and HVs equipped with such reliable vehicular communication can coordinate, improving safety and efficiency [6]. However, even in the presence of perfect communication, HV-AV interactions are still challenging as the behavior and intentions of HVs are still unknown, despite the vehicles' ability to perceive or share information in the communication network. Anticipating agents' behaviors and actions is an important part of real AVs, particularly in mixed autonomy environments. Due to the significance, prediction and HV behavior modeling is an active area of research [4], [5], [11]–[13].

However, while extensive research has been done in forecasting vehicles trajectories for classical AV stack [4], [5], [11], [12]; using prediction in the decision-making process has received less attention. Particularly in the domain of cooperative driving, social interactions, and multi-agent reinforcement learning (MARL), it is important to provide AVs with such capabilities, as presented in Figure 1, in which AV decision-making may benefit from anticipating the future states of other vehicles. Existing literature proposes probabilistic HV modeling [14] learned from human driving data or rule-based or hand-engineering methods to guide the AVs [15]. These systems frequently have difficulty communicating or negotiat-

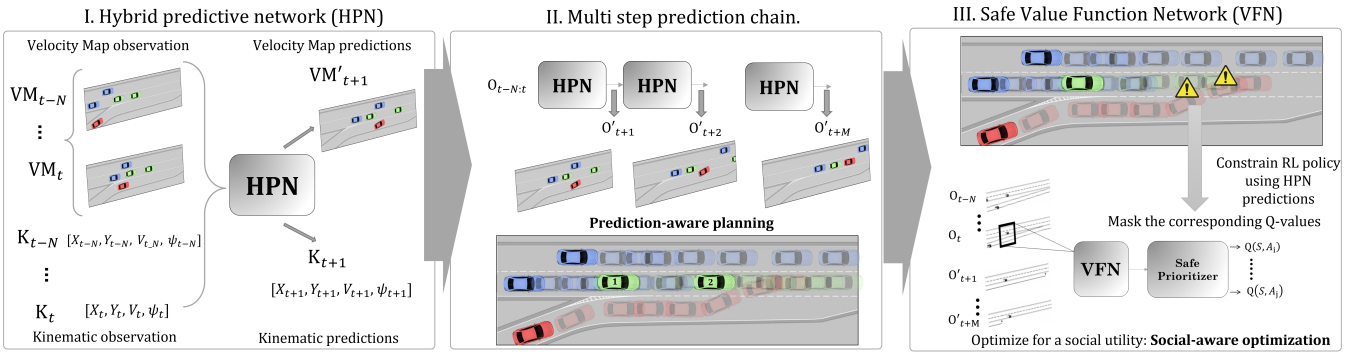

Fig. 2: An overview of our prediction-aware and social-aware cooperative driving approach for Multi-agent Cooperative Reinforcement Learning to improve the safety of CAVs. The proposed decentralized RL architecture employs an HPN, which gives AVs the ability to anticipate future states, which is used by the VFN to optimize for social utility and by the safety prioritizer to avoid high collision potential actions.

ing with other vehicles in complex scenarios. Other approaches use RL [16]–[18] and consider social interactions [17], [18] of AVs-HVs, proposing AVs that can learn from experience and influence HVs, while optimizing a social utility function that benefits all vehicles on the road. However, these approaches do not consider the evolution of the environment into the future and lack the capability of anticipating future states that can be used for decision-making.

Towards this end, we study how CAVs can leverage prediction and social awareness in RL decision-making, to improve safety and efficiency. Therefore, we propose the integration of those important components for AVs, i.e., prediction and social navigation. Figure 2 presents an overview of our approach. First, we propose a Hybrid Predictive Network (HPN) that intends to provide AVs the ability to predict other agents' potential future, as illustrated in Figure 2-I. Second, we use the HPN in a multi-step prediction chain that delivers a window of predicted observations to the value function network (VFN) (as illustrated in Figure 2-II). Finally, the safe VFN relies on a decentralized cooperative RL architecture that optimizes for a social utility and uses expressive velocity map predictions as part of the input states and interpretable kinematic predictions for a safety prioritizer. The safety prioritizer uses the kinematic predictions from the multi-step HPN to constrain the RL policy to ensure safety decision-making by masking the Q-states that produce unsafe results, as shown in Figure 2-III. We evaluate our prediction-aware and social-aware AV with related approaches under a variety of settings and show that, given the ability to forecast future observations, the AVs can use our proposed approach to improve safety, effectiveness, and overall traffic flow.

Our contributions are as follows:

- We present a prediction-aware, social-aware decentralized cooperative RL framework and formalize the altruistic cooperative driving problem as a partially observable stochastic game (POSG).
- We present a hybrid predictive network that provides AVs the ability to anticipate future observations and use it in a multi-step prediction chain that delivers multiple future observations to the value function network.
- We describe a robust safety prioritizer that uses interpretable kinematic predictions from the HPN to minimize future high-risk actions, constraining the RL policy to ensure safe decision-making, enhancing awareness of the imminent hazards, reducing collisions, and accelerating the learning process.

II. Related Work

A. Reinforcement Learning and Social Navigation

Current research in social navigation has demonstrated the importance of AVs as social actors and the benefits of AV-HV coordination [19]. A method for modeling and forecasting human behavior in situations involving multi-human interactions in highly multi-modal situations is proposed in [20]. HV models are learned from demonstration using inverse RL in [21] and [22]. Similarly, a centralized stochastic game model approach is presented in [23]. The authors in [24] and [25] proposed a shared reward function to enable cooperative trajectory planning for robots and humans. Sadigh et al. presents a strategy based on imitation learning, allowing AVs to influence HVs [26].

At the traffic level, the importance of coordination and the benefit of using AVs to guide traffic have also been investigated. Wu et al. [27] examines AVs' ability to stabilize a system of HVs and presents conditions under which enforcing safety constraints on the AVs while stabilizing the traffic improves the overall traffic performance. Similar works have highlighted the potential of influencing HVs and how AVs can guide the traffic flow [27], [28].

Recent works focus on optimizing traffic networks in mixed autonomy to reduce traffic congestion and improve safety and a model of vehicle flow is presented in [29] in which the planner optimizes for a social goal while improving traffic efficiency. The vehicle routing problem is studied in [30], which proposes an innovative learning-augmented local search system using a transformer architecture to mitigate the problem. In contrast to previous works, we do not rely on human cooperation, and secondly, our AVs incorporate prediction and planning to improve decision-making in cooperative driving.

B. Safety in Autonomous Vehicles

Safety is a priority for real-world adoption of AVs [31], [32]. As a result of the complexity of the driving task, safety concerns have been raised, such as: How can we prioritize AV safety in the face of uncertainty? and How can we train RL agents that prioritize AV safety?

Although priority should be given to safety, it often comes at the expense of effectiveness. Consider the following scenario: you decide to pass a car that is moving slowly in front of you. Overtaking a slower car poses a risk because the driver may abruptly change lanes and cause an accident. The only way to ensure safety is to avoid overtaking. Following this and similar arguments, it becomes clear that the only condition that completely guarantees safety is avoiding driving. As a result, the goals of efficiency and safety are frequently conflicting.

AVs based on RL can raise safety concerns as they can select unsafe actions due to function approximation [33]. To improve safety, authors in [34] propose a rule-based system for evaluating controller decisions and masking collision-causing actions. Cameron et al. investigate how humans can supervise agents to achieve an acceptable level of safety [35]. Others use a pure reward-shaping strategy, however, in this case, safety is not prioritized and the AVs are susceptible to select dangerous actions [33], [36]. To overcome this problem, authors in [33] present the concept of safe RL, which aims to increase safety in unobserved environment conditions. A short-horizon safety supervisor is proposed in [36] to replace unsafe actions with safer ones. A Q-masking strategy is presented in [37] to prevent collisions by deleting actions that might lead to a crash. Authors in [38] propose a safety supervisor that considerably decreases crashes [38].

To improve safety we employ a safety prioritizer that uses kinematic predictions from the multi-step prediction network to look ahead and avoid imminent collisions by masking high-risk actions in the short-horizon.

C. Behavior Modeling and Prediction

Human behavior is difficult to predict, and human decision-making is governed by inherently unobservable cognitive processes. The current works on driver behavior and social navigation approaches agents' coordination by either modeling driver behavior [14], [20], [39], [40] or simplifying and making assumptions about the nature of agent interactions [41], [42]. Other works on driver behavior modeling consider data mining [43], driver attributes [44], graph theory [45], or game theory [2].

In vehicular safety applications, the use of kinematic equations by CAVs for the prediction of neighboring vehicles' positions and trajectories in short time horizons is a common approach. Dynamic and kinematics-based solutions are used in [46]. These methods usually consider the vehicles to be rigid point masses and assume that the longitudinal velocity, acceleration, or other motion moments are constant, which are frequently accepted assumptions for prediction by conventional vehicle manufacturers. Amongst these kinematic models, the constant speed (CS) or acceleration (CA) model has more popularity for the prediction of road participants' position and speed in the cooperative vehicle safety application domain [47]–[51].

Numerous methods have defined trajectory prediction as a regression problem, and potent methods like Inverse Reinforcement Learning [52], Recurrent Neural Networks [53], [54] and Gaussian Process Regression and Gaussian Mixture Models [55]–[58], have been successfully applied in different applications. Authors in [59] present a LSTM model to forecast vehicle's trajectories. Other works leverage non-parametric Bayesian approaches to predict fundamental patterns of observable time series. Particularly, within the non-parametric bayesian inference approaches, Gaussian Process (GP) has demonstrated a significant performance [51]. Because of recent advances in deep generative models, generative approaches have been widely used [60], [61]. The majority of works in this domain use Autoencoders, such as Conditional Variational Autoencoder, Recurrent Variational Autoencoder, or Generative Adversarial Network [60], [61].

Differently, we propose a combined approach to predict velocity map images directly using a predictive autoencoder architecture and interpretable kinematics predictions using a GP solution. We use these predictions to improve decision-making, thereby integrating social navigation with prediction, which are crucial elements for AV navigation.

III. Preliminaries and Problem Formulation

A. Autoencoder

An Autoencoder (AE) is a neural network that is trained in an unsupervised approach to minimize the reconstruction error. The AE learns important features that allow reconstructing the original input and its architecture is generally divided into two main components: an encoder and a decoder. Following a similar approach, an AE can be trained for a prediction task assuming that we have the state at time t and the corresponding state at time $t+1$. Formally, the encoder maps the input x to a latent feature representation z denoted by $z = f_{w_e}(x)$. The decoder uses the latent representation z to obtain a reconstruction y of the input x, denoted by $y = f_{w_d}(z)$. The reconstruction error, i.e., the difference between x and the reconstruction y is used as the objective function.

B. Gaussian process

Gaussian processes (GP) are frequently used to predict future trajectory time series by regressing the observed time-series realizations from history, capturing the distinct patterns as they emerge in the data, making GP a useful tool for detecting patterns in time series [14], [62]. When using GP to forecast future trajectories, the set of m observed values is represented by an m-dimensional multivariate Gaussian random vector, described by an $m \times m$ covariance matrix and a m mean vector. This covariance matrix, often known as the GP kernel, is the foundation upon which GP detects and anticipates the underlying behavior of time series based on their recorded history. The fundamental GP components can be expressed mathematically as follows:

$$f(t) \sim \mathcal{GP}\left(m(t), k(t, t')\right); \quad (1a)$$
$$\{X_i\}_{i=1,2,\dots,m} = \{f(t_i)\}_{i=1,2,\dots,m} \sim \mathcal{N}(\mu, \Sigma); \quad (1b)$$
$$\mu = \left(m(t_1), \dots, m(t_m)\right)^T; \quad (1c)$$
$$\Sigma_{ij} = \kappa(t_i, t_j) \;\forall i, j \in \{1, 2, \dots, m\} \quad (1d)$$

where X_i, $f(t)$, $m(\cdot)$, and $\kappa(\cdot,\cdot)$ are the samples of the vehicles' state, observed or to be predicted at the time t_i, the unknown underlying function that the vehicles' states are sampled from, the mean and the covariance functions, respectively.

In this work, we leverage GP to improve kinematics prediction, and instead of working directly with the position time series, our GP inference algorithm treats the vehicles' heading and longitudinal speed as two independent time series that are regressed using GPs, and then using the predicted heading and longitudinal speed, the vehicles' positions are calculated. A model built using a non-parametric Bayesian inference framework dynamically adapts its complexity to the observed data, preventing overly complicated models yet catching unexpected patterns in the data as they emerge.

C. Partially Observable Stochastic Games (POSG)

In this section we present the notation for our RL based altruistic cooperative driving problem, defined as a POSG denoted by $\langle I, S, P, \{A_i\}_{i\in\{1,\dots,N\}}, \{O_i\}_{i\in\{1,\dots,N\}}, \{R_i\}_{i\in\{1,\dots,N\}}, \gamma \rangle$. At a given time t each agent $i \in I$ perceives the environment and receives a partial observation $o_i : S \to O_i$, considering the observation o_i and its policy $\pi_i : O_i \times A_i \to [0, 1]$, the agent takes an action $a_i \in A_i$ and transits to the state $s' \in S$ based on the transition probability $P(s'|s, a)$ and receives a local reward $r_i \in R$. Each agent i seeks to find an optimal policy $\pi^* : S \to A$, that maximizes the sum of future rewards $r_i \in R$, i.e., $\pi^*(s) = \arg\max_a Q(s, a)$, where, $Q(s, a) = \mathbb{E}[\sum_{k=0}^{\infty} \gamma^k R_k(s, a)|s_0 = s, a_0 = a]$, in which, $\gamma \in [0, 1)$ is the discount factor. The optimal action-value function can then be obtained using the Bellman optimality equation, $Q^*(s, a) = \mathbb{E}[R(s, a) + \gamma \max_{a'} Q^*(s', a')|s_0 = s, a_0 = a]$.

D. Deep Q-Network

Deep Q-network (DQN) and Double Deep Q-Network (DDQN) have been widely used in RL problems. DDQN regularly samples data from a buffer in order to calculate an estimate of the Bellman error, which is denoted by the following formula:

$$L(w) = \mathbb{E}_{s,a,r,s' \sim RM}[(\text{Target} - \tilde{Q}(s, a; w))^2] \quad (2)$$

$$\text{Target} = R(s, a) + \gamma \tilde{Q}(s', \arg\max_{a'} \tilde{Q}(s', a'; w); w^-) \quad (3)$$

Following this, the DDQN algorithm learns an approximate action-value function ($\tilde{Q}(\cdot)$) by performing gradient descent steps as $w_{i+1} = w_i - \alpha_i \hat{\nabla}_w L(w)$, on the loss L. Here, w represents the online network weights and w^- represents the target network weights (updated at frequency $\text{Target}_{\text{update}}$). The experience replay buffer (RM) is used to generate training samples (s, a, r, s'), which are randomly drawn to protect from correlated observations and non-stationary data distribution.

E. Driving Scenarios and Behaviors

We study the performance of our framework on multiple HV behaviors and scenarios. We design a set of scenar-ios, F such as straight highway, highway exiting, highway merging, intersection, and roundabout scenarios, defined as $f_h, f_e, f_m, f_i, f_r \in F$ correspondingly. Using these scenarios, we train AVs that are social-aware by using an altruistic reward that embedded Social Value Orientation (SVO) in the AVs. Properly, we describe social preferences (altruism or egoism) by the AV's SVO angular phase ϕ [17], [63]. To simulate diverse behaviors we compute the appropriate parameter values that simulate the desired behaviors. We compute the HV driver parameters (P) and based on the parameters (P) generate a set of behaviors B, i.e., conservative, moderate and aggressive, $b_c, b_m, b_a \in B$ used within the simulator [45], [64]. A mixed behavior scenario is obtained by sampling from the behaviors in B.

F. Problem Formulation

In this work, we focus on prediction-aware planning for altruistic cooperative driving. We assume our scenarios contain a set of AVs $i_i \in I$ and HVs $h_k \in H$, with diverse SVO. We assume that AVs are connected and perceive a partial observation of the environment $\tilde{o}_t \in \tilde{O}$, perceiving a subset of vehicles $C = \tilde{H} \cup \tilde{I}$, i.e., a subset of HVs $\tilde{H} \subseteq H$ and AVs $\tilde{I} \subseteq I$. We study the following problem: How AVs can leverage prediction in decision-making to learn optimal cooperative policies $\pi(s)$ in a mixed-autonomy environment under different HVs behaviors $b \in B$ and scenarios $f \in F$.

The RL-based altruistic cooperative driving problem is formalized as a POSG as described previously, attempting to obtain optimal policies that produce socially advantageous outcomes. To formalize our prediction problem, let us represent our state at time t for the vehicle (car) c, $c \in C$, as s_t^c and let $s_t = s_t^{1,\dots,|C|}$ represent the state for all the vehicles within the perception range. We assume that our state s_t consists of a stack of N past observations and M future hypotheses, accounting for temporal and prediction information, i.e., $s_t = [\tilde{o}_{t-N:t}, \tilde{o}'_{t+1:t+M}]$ for all the vehicles within the local observation. The prediction system takes as input the previous observations $\tilde{o}_{t-N:t}$ and aims to produce $\tilde{o}^c_{t+1:t+M}$. We note that this is the general notation, and in our framework, s_t is not just the vehicle trajectory, but a combination of vehicle kinematic trajectory and a velocity map, the details of which are presented in the following sections.

The previous ($\tilde{o}_{t-N:t}$) and anticipated observations ($\tilde{o}^c_{t+1:t+M}$) are used to learn an optimal policy at a given state s_t, $\pi : S \to A$. The goal of this work is to train prediction-aware and social-aware AVs that can drive safely in a mixed-autonomy scenario.

IV. Prediction-aware altruistic cooperative driving

The POSG becomes significantly more complex in the presence of HVs since their behavior is difficult to predict and change over time. Therefore, predicting HV behavior is crucial for AVs' in a mixed-autonomy environment. On the basis of this insight, we develop a framework that combines prediction and planning. We propose a predictive network that provides predictions to the planner, and the planner learns to use those predictions for decision-making. The prediction networks give the AV the capability to anticipate the future, and the VFN embeds the predictions and learns the inter-agent relations while optimizing for a social utility.

Our approach uses the HPN that provides possible future observations. Then the HPN is used in a multi-step prediction chain that produces multiple possible future observations to the VFN. Finally, the VFN is trained to optimize a social utility within the RL framework. The VFN outputs Q-values, that are masked by a safety prioritizer, constraining the RL policy to a safe action space. The outline of our framework is presented in Figure 2.

The two main sub-systems are the HPN and the VFN, where HPN is a predictive autoencoder and VFN is a 3D convolutional neural network (CNN). We hypothesize that the combination of prediction (HPN) and decision-making (VFN) improves the AVs' ability to learn to navigate complex scenarios. The input of the system is the hybrid spatio-temporal state representation, i.e., VelocityMaps and kinematic state and the output are the action-values, and after the unsafe actions are masked, the action with the highest Q-values is selected ($a = \max_{a' \in \mathcal{A}_{safe}} Q(s; a'; w)$ at the given state $s \in S$). To encourage the required safe social behavior in the AVs, we design a suitable reward function.

A. Action and State Space

Action Space: This study aims to investigate interactions between agents and between AVs and HVs. As a result, we decide to choose the action-space as a collection of discrete meta-actions $a_i \in A_i$ at an abstract level, and the abstract actions are transformed into control signals. We specifically choose a set of actions (a_i) as follows:

$$a_i \in A_i = \begin{bmatrix} \text{Lane Left} \\ \text{Idle} \\ \text{Lane Right} \\ \text{Accelerate} \\ \text{Decelerate} \end{bmatrix} \tag{4}$$

State Space: The AVs at every time step t receive a local observation of the environment $\mho_t \in \Theta_t$. As temporal information is crucial for the driving task we incorporate N consecutive observations. We use VelocityMaps (VM) and Kinematic (K) information, at time step t, each combination of VM and K is an observation from the environment as,

$$\mho_t \in \Theta_t = \begin{matrix} VM_t \\ K_t = X_t; Y_t; V_t; \Phi_t \end{matrix} \tag{5}$$

The kinematic information is included to explicitly incorporate the movement data, which helps the training process, and also serves to obtain accurate Kinematic prediction for the safety prioritizer. Additionally, as anticipating futures states is also important for decision-making in complex scenarios, the prediction chain generates a sequence of M hypotheses from the observations that provide information on how the environment could probably evolve into the future. We combine N consecutive past observations and M hypotheses from the prediction network to construct a more useful state. Therefore, our state consists of a stack of N past observations and M futures hypothesis, accounting for temporal and prediction information, i.e., $s_t = [\mho_{t-N:t}; \mho'_{t+1:t+M}]$.

The VM information incorporates the relative vehicle's speed in pixel values [63]. The K_t is a matrix in which the rows are the number of vehicles included in the observation and the columns contain the kinematics information for each vehicle. The kinematics information for each vehicle c, i.e., k_t^c, is a 4-dimensional vector encoding the vehicle's position, velocity and heading, and it is formed by the kinematics of the vehicle $x; y; v; \phi$, where $(x; y)$ represents the vehicle position, v is the longitudinal speed, ϕ is the heading, and $(\underline{x}; \underline{y})$ are computed as $\underline{x} = v \cos\phi; \underline{y} = v \sin\phi$. The K_t embeds the kinematics of surrounding vehicles, and it includes the kinematics information k_t^c for all the $c \in C$ vehicles, in addition to the ego vehicle, i.e., $K_t = \left[k_t^{ego}; k_t^1; k_t^2; \ldots; k_t^{|C|}\right]^{>}$. Each row r of K_t matrix contains the kinematics information for the vehicle c, $k_t^c = [x_t^c; y_t^c; v_t^c; \phi_t^c]$.

B. Reward Function

Inspired by the work in sympathy and cooperation for autonomous driving [17] [63], we design a reward function that takes into account the traffic metrics, social and individual rewards of AVs and HVs. Therefore, we separate the reward function for each agent $I_i \in I$ in two terms: an egoistic reward R^{ego}, and a social reward R^{social}, as

$$R_i(s; a) = R^{ego} + R^{social}; \tag{6a}$$

$$R^{ego} = \cos\phi_i r_i(s; a); \tag{6b}$$

$$R^{social} = \sin\phi_i \sum_j r_{i;j}^{AV}(s; a) + \sin\phi_i \sum_k r_{i;k}^{HV}(s; a) \tag{6c}$$

in which $i \in I$, $j \in (\mathcal{I} \setminus \{I_i\})$, $k \in \mathcal{H}$. The ego vehicle's reward is defined by r_i and the angle ϕ allows the adjustment of the level of the egoistic and social components.

The R^{social} term considers the social utility of the k HVs and j AVs for the agent i, i.e., $r_{i;k}^{HV}$ and $r_{i;j}^{AV}$ defined as $r_{i;k}^{HV} = \frac{1}{d_{i;k}^{\lambda}} \sum_m \omega_m x_m$ and $r_{i;j}^{AV} = \frac{1}{d_{i;j}^{\lambda}} \sum_m \omega_m x_m$, respectively, where m is the set of traffic metrics that have been taken into account in the utility of the vehicle (crashes, speed, and distance traveled), x_m represents the m metric, and w_m represents the weights (metrics importance). The terms $d_{i;k}/d_{i;j}$ represent the gap between the ego-AV and the associated HV/AV and λ is a hyperparameter that sets the importance of neighboring vehicles.

C. Architecture.

The proposed prediction-aware planning architecture is presented in Figure 2 and consists of the Hybrid predictive

network (HPN, Figure 2-I), the multi-step prediction chain (Figure 2-II), the value function network (VFN, Figure 2-III), and the safety prioritizer. The HPN (as shown in Figure 3) serves as a predictive autoencoder network. It takes as input the history of observations at time t, i.e., $o_{t\ N:t}$ and produces a predicted observation at time $t+1$, i.e., o^c_{t+1}. The prediction chain is a multi-step prediction chain that uses the HPN in a chain to produce a set of M hypotheses. It takes a history of observations at time t, i.e., $o_{t\ N:t}$ and produces a set of M predicted observations, i.e., $o^c_{t+1:t+M}$ for the VFN. Prediction-aware planning is made possible by combining prediction (HPN) and decision-making (VFN), which improve driving performance in challenging situations. The details of the architecture are presented in the following sections.

1) Hybrid predictive network (HPN): The HPN (Figure 3) is a prediction autoencoder network, it uses the sequence of N observations at time t, i.e., $o_{t\ N:t}$ and outputs a predicted observation at time $t+1$, i.e., o^c_{t+1}. The HPN consists of a symmetric encoder-decoder architecture. The encoder consists of 3 convolutional layers with 3x3 filters, with 32, 64, and 64 feature maps. The encoder takes as input the history of observations ($o_{t\ N:t}$), where each observation consists of a velocity map image (VM_t) and the Kinematic matrix (K_t). The VM from $t\ N:t$ are passed through the 3-convolutional layers and the K vectors from $t\ N:t$ are passed through 2-FC (fully connected) layers with 128 hidden units, whose final layer contains the same number of hidden units as in the convolution network (CNN) output. The outputs (VM features and K features representations) are combined using element-wise addition operation. The decoder consists of a symmetric version of the encoder, i.e., a deconvolutional network with 3-convolutional layers and 2-FC layers. The convolutional layers produce a prediction for the next VM^0_{t+1} and the FC layers produce the prediction for the next K^0_{t+1}. The CNN encoder is designed to extract important spatial information of the input VM image. The predictive autoencoder is trained by minimizing the Mean Squared Error (MSE) between the prediction o^0_{t+1} and the target o_{t+1}.

Although the AE provides kinematic predictions, we found that an indirect hybrid GP prediction approach to correct the kinematic predictions provides better results. Our findings are based on previous works that show how a GP-based prediction system is powerful for accurate kinematic predictions and often performs better than other models like AE and LSTM models [14] [56] [32].

Therefore, while we use the predictive AE to predict the next VM^0_{t+1} image and K^0_{t+1} state, we correct the kinematic state K^0_{t+1} using a GP approach to improve kinematics prediction. We particularly find that accurate kinematics prediction are important for the safety prioritizer that uses the predictions to constrain the RL policies to a safer space. In this work, instead of directly using the position time series ($x_{t\ N:t}; y_{t\ N:t}$) for each vehicle, our GP inference algorithm treats the vehicles' heading ($_{t\ N:t}$) and speed ($v_{t\ N:t}$) as two independent time series that are regressed using GPs, and then calculates the vehicles' position ($x^0_{t+1:t+M}; y^0_{t+1:t+M}$) using the predicted heading and speed. We call this approach GP-indirect prediction and present more details in the following sections.

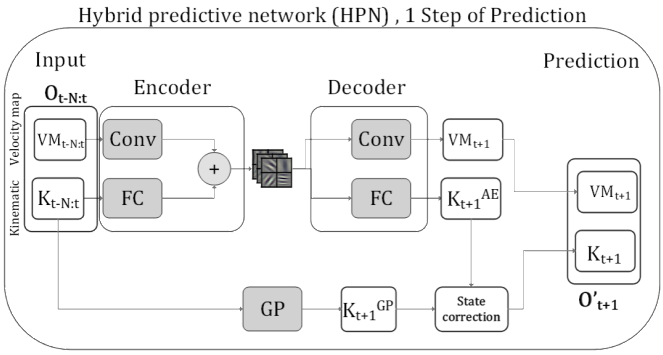

Fig. 3: Architecture of the hybrid predictive network (HPN) for one prediction step. Using HPN in a chain will give AVs the ability to predict future observations.

For each vehicle c, the GP prediction algorithm takes as input the history of the 4-dimensional kinematic vector ($x_{t\ N:t}; y_{t\ N:t}; v_{t\ N:t};\ _{t\ N:t},$), uses the heading ($_{t\ N:t}$) and speed ($v_{t\ N:t}$) time series to predict their future values ($^0_{t+1:t+M}; v^0_{t+1:t+M}$). Then after modelling speed and heading, the future position of the vehicle, i.e., $x^0_{t+1:t+M}; y^0_{t+1:t+M}$ is computed as follows,

$$f V_i g_{i=1;2;:::;m} = f f_{speed}(t_i) g_{i=1;2;:::;m}\ N(v; \overline{v}); \quad (7a)$$

$$_v = \overline{\quad} \quad m_v(t_i); \quad _{v^{i;j}} = k_v(t_i; t_j)\ 8i; j\ 2\ f1; 2; :::; mg; \quad (7b)$$

$$f_{speed}(t)\ \ gp(m_v(t); k_v(t; t^0)) \quad (7c)$$

$$f\ _i g_{i=1;2;:::;m} = f f_{heading}(t_i) g_{i=1;2;:::;m}\ N(\ ; \overline{\quad}); \quad (8a)$$

$$\overline{\quad} = m\ (t_i); \quad _{i;j} = k\ (t_i; t_j)\ 8i; j\ 2\ f1; 2; :::; mg; \quad (8b)$$

$$f_{heading}(t)\ \ gp(m\ (t); k\ (t; t^0)) \quad (8c)$$

$$x^c_{t+1} = x_t + \int_t^{t+1} f_{speed}(t) \cos(f_{heading}(t))\, dt; \quad (9a)$$

$$y^c_{t+1} = y_t + \int_t^{t+1} f_{speed}(t) \sin(f_{heading}(t))\, dt \quad (9b)$$

From the output of the GP model, the 4-dimensional kinematic vector ($k^{GP}_{t+1} = x_{t+1}; y_{t+1}; v_{t+1};\ _{t+1}$) for each vehicle is used to correct the AE kinematic prediction (k^{AE}_{t+1}), and the GP prediction is performed for each vehicle in the K matrix (rows of the matrix) and a new matrix is formed with all the predictions at time $t+1$, i.e., $K^c_{t+1} = k^{ego}_{t+1}; k^1_{t+1}; k^2_{t+1}; :::; k^{jCj}_{t+1}$. The final predicted observation is a combination of the predicted velocity map (VM^0_{t+1}) and the corrected kinematic prediction (K^0_{t+1}) as shown in Figure 3, i.e.,

$$o^0_{t+1} = \quad K_{t+1} = X_{t+1}; Y_{t+1}; V_{t+1};\ _{t+1} \quad (10)$$

Algorithm 1 Multi-step prediction chain.

Input $o_{t\ N:t}$. The sequence of previous observations.
for $t = t$ to $t + M$ do
  Predict $o^{0}_{t+1} = \text{HPN}(o_{t\ N:t})$
  Save prediction o^{C}_{t+1} and use it for the next step
end for
Output $o^{C}_{t+1:t+M}$. The sequence of predicted observations.

2) Multi-step prediction chain: The prediction chain, as presented in Figure 4 is a multi-step prediction process that uses the HPN (Figure 3) in a chain to produce a set of M future hypotheses. It takes a history of observation at time t, i.e., $o_{t\ N:t}$ and produces a set of M predicted observations, i.e., $o^{C}_{t+1:t+M}$, as described in algorithm 1, to compute the input state for the VFN, i.e., $s_t = [o_{t\ N:t}; o^{C}_{t+1:t+M}]$.

3) Safety Prioritizer: In order to improve safety, we propose a safety prioritizer within our VFN. The safety prioritizer penalizes high-risk actions, thereby reducing imminent crashes. If the AVs come into an unexpected situation and based on the output of the VFN, decide to perform a high-risk action, the safety prioritizer will mask the action. The safety prioritizer is comprised of two algorithms, i.e., Algorithm 2 that checks for safe actions and Algorithm 3 that performs action selection.

Algorithm 2 verifies if the selected action a_t is safe based on a safety score for M_{steps} of prediction. Algorithm 2 simulates I_i taking the action a_t and uses the kinematic predictions from HPN, i.e., $k^{C}_{t+1:t+M} = \text{HPN}(k_{t\ N:t})$, for all vehicles in the road $c \in C$ to compute the time-to-collision (ttc) at time t, i.e., ttc_t between I_i and all $c \in (I \cup H) \setminus \{I_i\}$ using $x; y; v;$, at each prediction step is calculated and the minimum ttc is saved, and using the predicted ttc for all the M_{steps} of prediction ($ttc_{t+1:t+M}$), the $safe_{score}$ is computed. The $safe_{score}$ is a weighted average of the $ttc_{t+1:t+M}$, with exponential decay to give more importance to the short-term predictions. Finally, if $safety_{score} < safe_{th}$ or any of the predicted ttc is less than the critical threshold, i.e., $any(ttc_{t+1:t+M}) < critical_{th}$, the action is considered unsafe. The $safe_{th}$ is the safe ttc threshold for possible crash, and $critical_{th}$ is a critical ttc threshold for imminent crash. If the current action is considered unsafe, Algorithm 3 will select another action.

Algorithm 3 presents the selection of the action. It iteratively verifies the actions using Algorithm 2 and selects a safe action that follows the learned policy. The restricted actions will prevent the agent from engaging in risky behavior during training, resulting in a more balanced learning and efficient sampling.

4) Safe Value Function Network (VFN): The VFN estimates the state-action value function. The combination of prediction (HPN) and decision-making (VFN) allows prediction-aware planning and improves the AVs' ability to learn to navigate complex scenarios, and the safety prioritizer further increases safety. The proposed approach utilizes deep reinforcement learning (DRL) to achieve a high-level policy for safe tactical decision-making. As presented, the input consists

Algorithm 2 Action evaluation.

Simulate I_i taking the action a_t
Get Kinematic predictions from HPN, i.e., $k^{C}_{t+1:t+M} = \text{HPN}(k_{t\ N:t})$ for all vehicles in the road $c \in C = (I \cup H) \setminus \{I_i\}$
for $t = t + 1$ to $t + M$ (Compute safety score for M_{steps} predictions) do
  Compute ttc_t between I_i and all $c \in C \setminus \{I_i\}$ using $x; y; v;$ at time t
  Compute $\min(ttc_t)$
  Get next prediction at $t = t + 1$
end for
Compute $safe_{score}$ using the predicted $ttc_{t+1:t+M}$

$$safe_{score} = \frac{\sum_{i=t+1}^{t+M} w_i ttc_i}{\sum_{i=t+1}^{t+M} w_i}$$

if $safe_{score} < safe_{th}$ or $any(ttc_{t+1:t+M}) < critical_{th}$ then
  Return unsafe
else
  Return safe
end if

Algorithm 3 Action selection.

Initialize $A_{safe} = A$
while A_{safe} is not empty do
  if during training then
    Select a_t following the exploration policy on set A_{safe}
  else if during test then
    Select $a_t = \max_{a' \in A_{safe}} Q(s_t; a'; w)$
  end if
  if a_t is safe (Algorithm 2) then
    Return a_t
  else
    Remove a_t from A_{safe}
  end if
end while
Compute the $safety_{score}$ as in Algorithm 2
Return a_t with highest $safety_{score}$ in A

of a stack of N past observations and M future hypotheses, i.e., $s_t = [o_{t\ N:t}; o_{t+1:t+M}]$, and the 3D CNN operates as a feature extractor. The VFN is trained to learn the optimal Q-values that maximize our social reward function, optimizing social utility. During training, agents are trained in a semi-sequential manner, as in [17].

The VFN outputs the Q-values that are masked by a safety prioritizer, constraining the RL policy to a safe action space. Therefore, in our framework, when the agent policy chooses an unsafe action, the safety prioritizer masks the action and selects a safer action, saving the unsafe action (a_t) and the associated state in the RM with a negative reward (r_{unsafe}). By reducing episode restarts due to potential collisions, the safety prioritizer increases sample efficiency and safety.

The proposed prediction-aware planning and the social-aware optimization algorithm is described in Algorithm 4. We first run a batch of sample simulations to pre-fill our replay

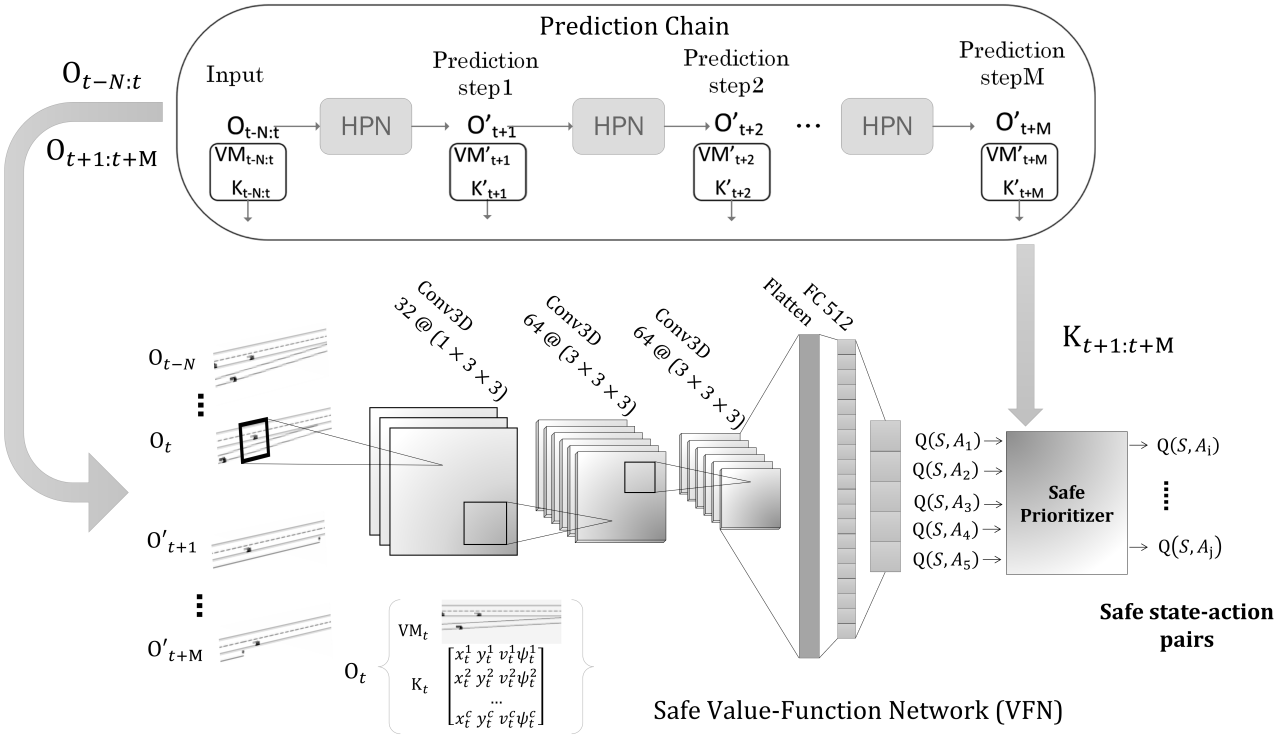

Fig. 4: Multi-step prediction chain and Safe Value Function Network (VFN). The prediction chain is a multi-step prediction chain that uses the HPN in a chain to produce a set of M future hypotheses. The VFN is a 3D CNN that acts as a function approximator, it uses temporal information and prediction to improve decision-making.

buffer before starting the learning phase. To account for the unbalance in training data, the experience replay buffer is re-balanced [63].

V. Experimental Results

This section begins with a description of the simulation environment and the HV model in mixed-autonomy traffic. Before presenting our findings, practical aspects of training and validation are explored. Finally, we present our results showing the importance of prediction-aware planning for cooperative driving.

A. Implementation Details

1) RL environment and Computational Details: We customize the OpenAI Gym Highway environment in [65]. We design five scenarios for our experiments, i.e, a straight highway, highway exiting, highway merging, intersection, and roundabout scenarios ($f_h; f_e; f_m; f_i; f_r \ 2 \ F$). The AVs are trained surrounded by HVs with various behaviors, i.e, conservative, moderate, and aggressive, ($b_c; b_m; b_a \ 2 \ B$). A scenario with mixed behavior is obtained by sampling from the behaviors in B for each HV. The VFN is trained for $N_{episodes} = 15{,}000$ episodes and multiple iterations of the training procedure are carried out to guarantee the convergence of the policies. Table I lists our training and simulation parameters.

TABLE I: Simulation parameters.

Parameter	Value	Parameter	Value
K prediction	GP, RBF kernel	$N_{episode}$	15,000
Prediction Horizon	4s	decay	Linear
History window	2s	RM buffer size	8,000
Latent Dimension	512	Initial exploration $_0$	1.0
Batch size	64	Final exploration	0.05
Learning rate $_0$	0.0005	Optimizer	ADAM
$Target_{update}$	300	Discount factor	0.95

2) Driver Modeling: We model the HV's lateral behavior using the MOBIL model [66] and the longitudinal behavior of HVs is based on the Intelligent Driver Model (IDM) [67]. MOBIL is based on the safety and incentive criteria. For safety, it verifies $a_n > b_{safe}$, where a_n is the deceleration of the following vehicle in the new lane, and b_{safe} is a safe threshold. Then MOBIL verifies the incentive to change lane measured by $a^0_{ego} \ a_{ego} + p \ (a_n \ a_n) + (a_o \ a_o) > a_{th}$, where p is the politeness term and a_{ego} , a_n and a_o are the accelerations of the ego-HV, the new following vehicle, and the old following vehicle, respectively. Finally, based on the MOBIL model, if both criteria are verified, then the HV performs a lane change.

The longitudinal behavior of HV k is modeled by using the IDM model which computes the acceleration $\dot{v}_k$ as $\dot{v}_k = a_{max} \ 1 \ \left(\frac{v_k}{v^0_k}\right)^{\delta} \ \left(\frac{d(v_k; v_k)}{d_k}\right)^2$; in which is the exponent of acceleration, d_k is the gap, v^k_0 is the preferred speed, and v_k is the current speed. Following, the preferred minimum gap is

Algorithm 4 Prediction-aware planning DDQN.

Define and Initialize Replay Memory buffer RM.
Define and Initialize action-value function $Q(:;w)$ and target network $\hat{Q}(:;w)$ with $w = w_{ini}$ and $\hat{w} = w$
Save in the RM the first's E_{ini} episodes.
for e = E_{ini} to $N_{episode}$ do
  Obtain observation history $o_{t\ N:t}$
  Predict M hypothesis $o^{c}_{t+1:t+M}$ (Algorithm 1)
  Compute $s_t = [o_{t\ N:t}; o^{0}_{t+1:t+M}]$
  for t = t_{ini} to T do
    for I_i in I do
      For agents I_j, $j = i$, freeze w
      for $N_{iterations}$ do
        With probability choose a_t randomly,
        else choose $a_t = max_{a' 2 A} Q(s_t; a'; w^{+})$
        Verify action a_t (Algorithm 2)
        if a_t is not safe then
          Store transition $(s_t; a_t; r_{unsafe}; ;)$ in RM
          a_t = Select a safe action (Algorithm 3)
        end if
        Take a_t (a_{safe}), and observe $r_t; o_{t+1}$
        Store transition $(s_t; a_t; r_t; s_{t+1})$ in RM
        Compute w^{+}_{k+1} w^{+}_{k} $r_w \hat{L}(w^{+})$
      end for
      Disseminate weights $w = w^{+}$ for all I_i 2 I
    end for
    Reset $\hat{w}$ w every $Target_{update}$
  end for
end for

computed as $d(v_k; v_k) = d^0 + v_k T^0 + \frac{v_k(v_k - v_k)}{2\sqrt{a \cdot a_{des}}}$ where d is the minimum distance, T is the safe time gap, a_{max} is the acceleration limit, and a_{des} is the deceleration limit.

We use the centrality metrics as in [45], [64] to obtain the parameters P that simulate the driver behaviors for the MOBIL and IDM models. In our scenarios, the computed parameters P that represent aggressive, conservative, and moderate behavior are shown in Table II.

TABLE II: Computed parameters P for different simulated driver behaviors.

Model	Parameter	Aggressive	Moderate	Conservative
MOBIL	p	0	0.3	1
	a_{th}	0 m=s^2	0.1 m=s^2	0.4 m=s^2
	b_{safe}	12.0 m=s^2	6.0 m=s^2	2.0 m=s^2
IDM	T^0	0.5s	1s	3s
	d^0	1 m	2 m	6.0 m
	acc_{max}	7.0 m=s^2	3.0 m=s^2	1.0 m=s^2
	acc_{des}	12.0 m=s^2	7.0 m=s^2	2.0 m=s^2

B. Evaluation Metrics and Hypotheses

The system's performance is evaluated based on safety, effectiveness, and prediction error. We select two indicators that, notwithstanding their correlation, offer distinct perspectives on the effectiveness of our approach. We calculate the proportion of episodes with at least one crash (C(%)) in order to measure safety. The vehicles' average traveled distance (DT(m)) is utilized to measure efficiency. Finally, the prediction error

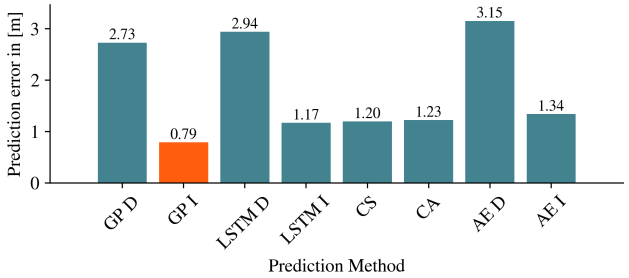

Fig. 5: Kinematic prediction baseline comparison in terms of position error (PE) in meters.

is measured in terms of the prediction reconstruction error (PRE) of the VM predictions and in terms of position error (PE) for the Kinematic prediction K. Based on those evaluation metrics we investigate the following hypotheses:

H1. The GP is a more powerful approach to predicting time series when compared to the AE for Kinematic predictions, therefore, using the GP for kinematic prediction improved the prediction performance when measured by the position error PE. Additionally, temporal information is important for accurate prediction, therefore we expect a higher performance of the VM prediction from our predictive autoencoder when using the observation history, measured by the prediction reconstruction error (PRE).

H2. The ability to forecast future states improve decision-making in AVs, therefore we anticipate a performance improvement of our prediction-aware VFN measured in terms of safety and efficiency when using the HPN.

C. Analysis and Results

1) Learning how to predict, Hybrid predictive network (HPN): Predicting the actions of HVs is a crucial component of AVs' decision-making. We take advantage of this feature and investigate how incorporating prediction into our framework improves safety and efficiency. We look into this insight

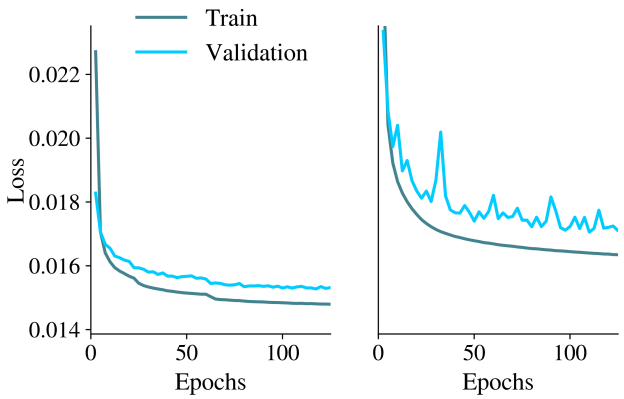

Fig. 6: Training and validation loss of the predictive network using observation history (left), and without history (right).

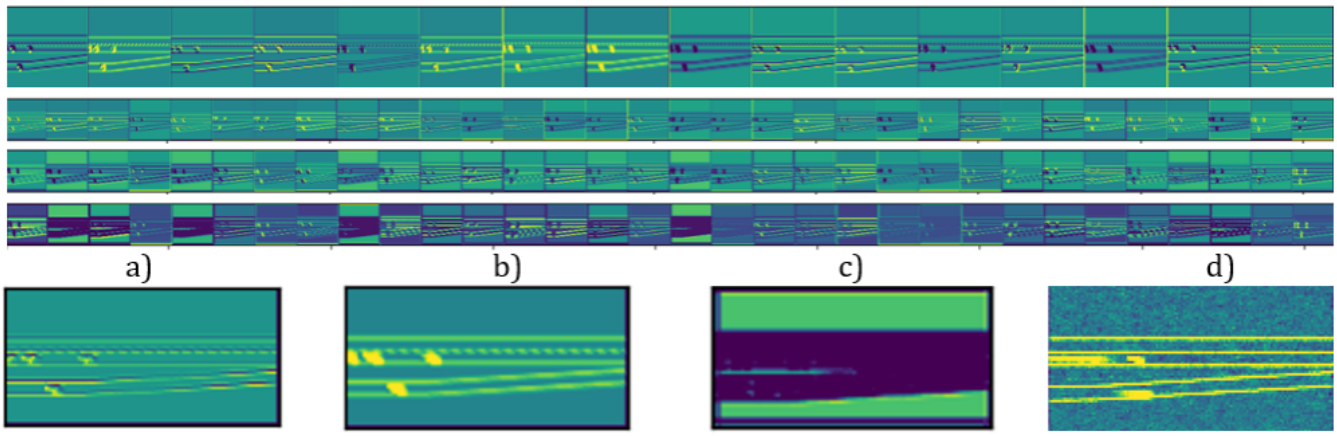

Fig. 7: Internal representations of the features at different layers for a merging scenario (Top), selected Internal representations that have been zoomed in (Bottom).

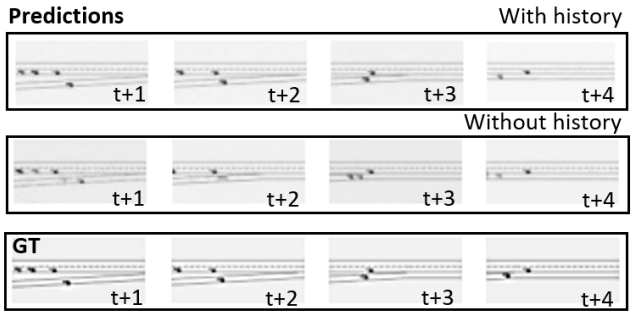

Fig. 8: Prediction chain for merging when using observation history (top), and without history (bottom).

while investigating H1. Particularly, we show that prediction in the image domain allows learning powerful representations, and we present how the HPN learns to predict the V M image and the advantages of using the GP approach to improve Kinematic prediction.

Kinematic prediction. We first investigate our H1 and show how using the GP for kinematic prediction improves the prediction performance. We compare different kinematic prediction baselines and measure their performance using the position error P E in meters. We compare five prediction approaches, i.e, the GP prediction approach, an LSTM network, Constant Speed (CS), Constant Acceleration (CA) based prediction, and the predictive AE kinematic prediction. GP, LSTM, and AE can be used to predict any time series and leveraging that, we consider two approaches for each method: direct and indirect prediction. Therefore, we compare eight baselines the GP direct (GP D), the GP indirect (GP I), LSTM Direct (LSTM D), LSTM Indirect (LSTM I), CS, CA, AE direct (AE D) and AE indirect (AE I).

In a direct prediction approach, $(x; y)$ are regressed by two distinct models learned from the history $(x_{t\ N:t}; y_{t\ N:t})$, producing direct predictions of futures $(x; y)$. Differently, in an indirect prediction approach, the vehicle's heading $(_{t\ N:t})$ and speed $(v_{t\ N:t})$ histories are considered and the models for heading and speed $(; v)$ are learned using the predictive approach. Using the learned models, the predictions for future heading and speed $(; v)$ are computed and utilized to calcu-late future position $(x_{t+1:t+M}\ ; y_{t+1:t+M})$. In our experiments, a compound kernel of linear and RBF is used following previous work [14].

As illustrated in Figure 5 the indirect GP (GP I, in orange) approach outperforms the other baselines in terms of P E, showing how this non-parametric Bayesian scheme allows the incorporation of complex model structures and is a suitable option for our kinematic prediction method, verifying our H1.

Observation history. Together with the Kinematic prediction, the HPN outputs the velocity map image prediction (V M). A prediction reconstruction error (P RE) loss is utilized to calculate the error between the predicted observation o^{pred} and the corresponding o, i.e., $L_2(o_{t+1}; o^0_{t+1}) = \sum_i (o_i \quad o^0_i)^2$. We evaluate the HPN's performance using only the current observation as input and history of N observations, demonstrating the importance of temporal information for accurate prediction measured by the P RE. Figure 6 depicts the training loss results, where the left image is for the HPN that uses the history of N observation and the right image uses just the current observation. As shown in the figure, when using the temporal information the P R E loss is approximately 20% smaller when using history (left) than without history (right). Similarly, Figure 8 shows the qualitative output of the HPN prediction chain using the current observation or a history of observations as input. The results with history (top) show clearer and more accurate visual predictions and confirm why the P R E is lower when using the history.

Figure 7 presents some qualitative results of the internal representations at different layers of the HPN for a merging scenario. In Figure 7 (bottom), a zoomed-in version of some internal representations are illustrated for visualization. As observed, the HPN learns to extract and highlight important information from the input observation, such as (a) lanes, (b) road agents, (c) road segments, and (d) possible hypotheses on how the environment evolves. Despite the fact that the HPN has not been trained for a segmentation task, it learns to segment the road, agents, and lanes, which could be useful information for the prediction and driving tasks.

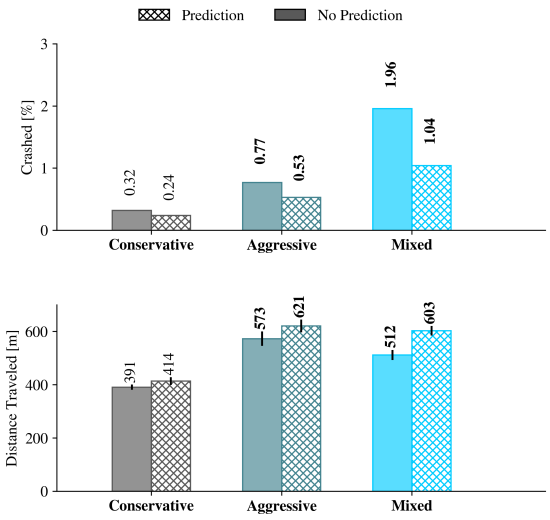

Fig. 9: Performance enhancement in a highway merging scenario resulted from using prediction. Safety is measured in terms of crash percentage (Top, C(%)) and efficiency by average traveled distance (Bottom, DT(m)).

2) Safer VFN leveraging prediction: Using the HPN and prediction chain, the VFN is trained to optimize for a social utility. We evaluate the performance of the VFN when using just the history as input, i.e, $s_t = [o_{t\ N:t}]$ and when additionally utilizing the prediction output of the prediction chain, i.e. $s_t = [o_{t\ N:t}; o_{t+1:t+M}]$. Figure 9 shows performance improvement by using prediction quantified by crash percentage (Top, C(%)) and average distance traveled (Bottom, DT(m)). The results present the AV's performance in the highway merging scenario f_m, in the presence of HVs with conservative, aggressive or mixed behavior. We observe that when train and test are performed in a conservative environment, or in other words, when HV yields and takes safer actions, the gains from prediction capabilities are not as noticeable, whereas, in an aggressive and mixed environment in which the behavior changes, the performance increases are significant. We believe that anticipating the future is especially useful in those scenarios, which is why performance has improved.

We evaluate the effectiveness of our architecture in diverse scenarios, as well as the performance enhancement of leveraging prediction. We argue that by using prediction, we provide the VFN a prior on how the world will evolve which is helpful for decision-making. Table III presents the results in different traffic scenarios, i.e, Exiting (f_e), Merging (f_m), Roundabout (f_r), intersection (f_i), and Highway (f_h) under mixed HV behaviors (b 2 B). We compare our architecture when using prediction (VFN+P), and without prediction (VFN) with other related architectures [17], [18], [68]. Our architectures as shown in Table III outperform the alternative methods, and the improvements are particularly pronounced in the more complex scenarios.

The combination of prediction (HPN) and decision-making (VFN) allows for prediction-aware planning and improves the AVs' ability to learn to navigate complex scenarios, and the safety prioritizer is further improved by leveraging the information provided by the prediction chain, increasing safety and efficiency. The results presented in Figure 9 and Table III verify our H2. Additionally, Figure 10 provides the output of the prediction chain for different traffic scenarios, showing qualitative results that further illustrate the capabilities of the prediction network to forecast the future.

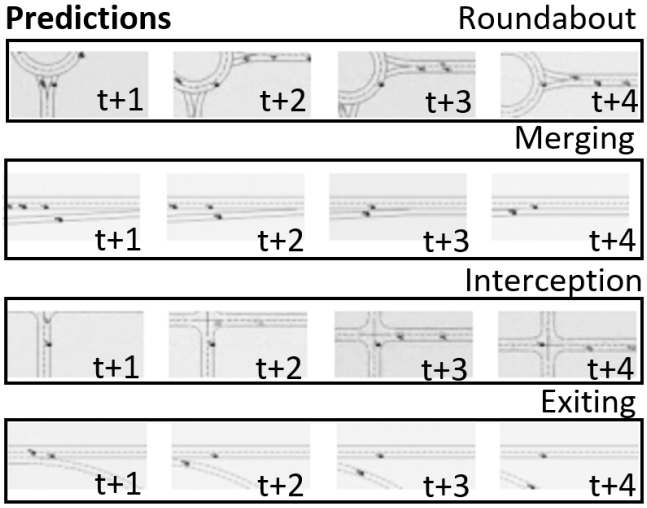

Fig. 10: Qualitative results of the prediction chain output for multiple scenarios

TABLE III: Performance Comparison (Measured by C(%)) of related architectures showing the performance improvement of our predictive VFN, particularly in challenging scenarios such as intersection and roundabout. The results are shown in different scenarios, Exiting (f_e), Merging (f_m), Roundabout (f_r), intersection (f_i), and Highway (f_h).

Approach	f_e	f_m	f_r	f_i	f_h
Conv2D+DQN [68]	24.62	29.12	49.03	54.78	17.21
Conv3D+A2C [17]	9.23	14.99	21.17	36.62	7.43
Conv3D+DQN [18]	3.91	2.59	14.62	24.30	1.31
Safe DQN [64]	2.51	1.95	9.04	18.67	0.44
VFN	2.47	1.96	8.94	17.90	0.39
VFN + P	1.91	1.04	7.01	11.10	0.31

VI. Conclusion

We propose the integration of two crucial components for AVs, social navigation and prediction. The safety and reliability of AVs depend on their predictive capabilities, social awareness, and ability to engage in complex social interactions. For that reason, we propose prediction-aware planning and social-aware optimization in a cooperative RL framework, to allow safe and socially-desirable outcomes. We provide AVs the ability to anticipate the future, allowing them to take informed decisions and proactive actions in AV-HV social interaction scenarios. The safety prioritizer leverages interpretable kinematic predictions from the HPN to restrict the RL policy to assure safe decision-making, reducing future high-risk actions, increasing awareness of the immediate risks, and consequently decreasing crashes. We compare our prediction-aware AV to other solutions and demonstrate how our approach consistently

improves safety and efficiency on the road in multiple scenarios.

Further research needs to be done using real human driver data, and more complex traffic scenarios. We plan to extend this work in this direction and show robust generalization capabilities of our agents. We intend to explore learning meaningful and interpretable representations and predictions to help build intuition on the AV's decision-making process.

References

[1] A. Cosgun, L. Ma, J. Chiu, J. Huang, M. Demir, A. M. Anon, T. Lian, H. Tafish, and S. Al-Stouhi, "Towards full automated drive in urban environments: A demonstration in gomentum station, california," in 2017 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2017, pp. 1811–1818.

[2] W. Schwarting, A. Pierson, J. Alonso-Mora, S. Karaman, and D. Rus, "Social behavior for autonomous vehicles," Proceedings of the National Academy of Sciences, vol. 116, no. 50, pp. 24 972–24 978, 2019.

[3] F. Sagberg, Selpi, G. F. Bianchi Piccinini, and J. Engstrom, "A review of research on driving styles and road safety," Human factors, vol. 57, no. 7, pp. 1248–1275, 2015.

[4] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, "Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data," in European Conference on Computer Vision. Springer, 2020, pp. 683–700.

[5] S. Mozaffari, O. Y. Al-Jarrah, M. Dianati, P. Jennings, and A. Mouzakitis, "Deep learning-based vehicle behavior prediction for autonomous driving applications: A review," IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 1, pp. 33–47, 2020.

[6] S. Aoki, T. Higuchi, and O. Altintas, "Cooperative perception with deep reinforcement learning for connected vehicles," in 2020 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2020, pp. 328–334.

[7] B. Toghi, M. Saifuddin, M. Mughal, and Y. P. Fallah, "Spatio-temporal dynamics of cellular v2x communication in dense vehicular networks," in 2019 IEEE 2nd Connected and Automated Vehicles Symposium (CAVS). IEEE, 2019, pp. 1–5.

[8] G. Shah, R. Valiente, N. Gupta, S. O. Gani, B. Toghi, Y. P. Fallah, and S. D. Gupta, "Real-time hardware-in-the-loop emulation framework for dsrc-based connected vehicle applications," in 2019 IEEE 2nd Connected and Automated Vehicles Symposium (CAVS). IEEE, 2019, pp. 1–6.

[9] G. Shah, M. Saifuddin, Y. P. Fallah, and S. D. Gupta, "Rve-cv2x: A scalable emulation framework for real-time evaluation of cv2x-based connected vehicle applications," in 2020 IEEE Vehicular Networking Conference (VNC). IEEE, 2020, pp. 1–8.

[10] G. Shah, S. Shahram, Y. Fallah, D. Tian, and E. Moradi-Pari, "Enabling a cooperative driver messenger system for lane change assistance application," arXiv preprint arXiv:2207.12574, 2022.

[11] B. Ivanovic and M. Pavone, "The trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs," in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 2375–2384.

[12] ——, "Injecting planning-awareness into prediction and detection evaluation," in 2022 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2022, pp. 821–828.

[13] A. Xie, D. Losey, R. Tolsma, C. Finn, and D. Sadigh, "Learning latent representations to influence multi-agent interaction," in Proceedings of the 4th Conference on Robot Learning (CoRL), November 2020.

[14] H. N. Mahjoub, A. Raftari, R. Valiente, Y. P. Fallah, and S. K. Mahmud, "Representing realistic human driver behaviors using a finite size gaussian process kernel bank," in 2019 IEEE Vehicular Networking Conference (VNC). IEEE, 2019, pp. 1–8.

[15] J. Rios-Torres and A. A. Malikopoulos, "A survey on the coordination of connected and automated vehicles at intersections and merging at highway on-ramps," IEEE Transactions on Intelligent Transportation Systems, vol. 18, no. 5, pp. 1066–1077, 2016.

[16] Y. Lin, J. McPhee, and N. L. Azad, "Anti-jerk on-ramp merging using deep reinforcement learning," in 2020 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2020, pp. 7–14.

[17] B. Toghi, R. Valiente, D. Sadigh, R. Pedarsani, and Y. P. Fallah, "Altruistic maneuver planning for cooperative autonomous vehicles using multi-agent advantage actor-critic," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2021.

[18] ——, "Cooperative autonomous vehicles that sympathize with human drivers," in 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2021.

[19] A. Pokle, R. Martín-Martín, P. Goebel, V. Chow, H. M. Ewald, J. Yang, Z. Wang, A. Sadeghian, D. Sadigh, S. Savarese et al., "Deep local trajectory replanning and control for robot navigation," in 2019 International Conference on Robotics and Automation (ICRA). IEEE, 2019, pp. 5815–5822.

[20] B. Ivanovic, E. Schmerling, K. Leung, and M. Pavone, "Generative modeling of multimodal multi-human behavior. in 2018 ieee," in RSJ International Conference on Intelligent Robots and Systems, 2018, pp. 3088–3095.

[21] M. Kuderer, S. Gulati, and W. Burgard, "Learning driving styles for autonomous vehicles from demonstration," in 2015 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2015, pp. 2641–2646.

[22] D. Sadigh, N. Landolfi, S. S. Sastry, S. A. Seshia, and A. D. Dragan, "Planning for cars that coordinate with people: leveraging effects on human actions for planning and active information gathering over human internal state," Autonomous Robots, vol. 42, no. 7, pp. 1405–1426, 2018.

[23] D. Hadfield-Menell, S. J. Russell, P. Abbeel, and A. Dragan, "Cooperative inverse reinforcement learning," Advances in neural information processing systems, vol. 29, pp. 3909–3917, 2016.

[24] P. Trautman and A. Krause, "Unfreezing the robot: Navigation in dense, interacting crowds," in 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2010, pp. 797–803.

[25] S. Nikolaidis, R. Ramakrishnan, K. Gu, and J. Shah, "Efficient model learning from joint-action demonstrations for human-robot collaborative tasks," in 2015 10th ACM/IEEE International Conference on Human-Robot Interaction (HRI). IEEE, 2015, pp. 189–196.

[26] D. Sadigh, S. Sastry, S. A. Seshia, and A. D. Dragan, "Planning for autonomous cars that leverage effects on human actions." in Robotics: Science and Systems, vol. 2. Ann Arbor, MI, USA, 2016.

[27] C. Wu, A. M. Bayen, and A. Mehta, "Stabilizing traffic with autonomous vehicles," in 2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2018, pp. 6012–6018.

[28] D. A. Lazar, E. Bıyık, D. Sadigh, and R. Pedarsani, "Learning how to dynamically route autonomous vehicles on shared roads," arXiv preprint arXiv:1909.03664, 2019.

[29] E. Bıyık, D. A. Lazar, R. Pedarsani, and D. Sadigh, "Incentivizing efficient equilibria in traffic networks with mixed autonomy," IEEE Transactions on Control of Network Systems, vol. 8, no. 4, pp. 1717–1729, 2021.

[30] S. Li, Z. Yan, and C. Wu, "Learning to delegate for large-scale vehicle routing," Advances in Neural Information Processing Systems, vol. 34, 2021.

[31] K. Srinivasan, B. Eysenbach, S. Ha, J. Tan, and C. Finn, "Learning to be safe: Deep rl with a safety critic," 2020. [Online]. Available: https://arxiv.org/abs/2010.14603

[32] M. Razzaghpour, S. Mosharafian, A. Raftari, J. Mohammadpour Velni, and Y. P. Fallah, "Impact of information flow topology on safety of tightly-coupled connected and automated vehicle platoons utilizing stochastic control," in (ECC 2022).

[33] Z. Li, U. Kalabić, and T. Chu, "Safe reinforcement learning: Learning with supervision using a constraint-admissible set," in 2018 Annual American Control Conference (ACC). IEEE, 2018, pp. 6390–6395.

[34] J. Wang, Q. Zhang, D. Zhao, and Y. Chen, "Lane change decision-making through deep reinforcement learning with rule-based constraints," in 2019 International Joint Conference on Neural Networks (IJCNN). IEEE, 2019, pp. 1–6.

[35] C. Hickert, S. Li, and C. Wu, "Cooperation for scalable supervision of autonomy in mixed traffic," arXiv e-prints, pp. arXiv–2112, 2021.

[36] S. Nageshrao, H. E. Tseng, and D. Filev, "Autonomous highway driving using deep reinforcement learning," in 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC). IEEE, 2019, pp. 2326–2331.

[37] A. Mohammadhasani, H. Mehrivash, A. Lynch, and Z. Shu, "Reinforcement learning based safe decision making for highway autonomous driving," arXiv preprint arXiv:2105.06517, 2021.

[38] D. Chen, Z. Li, Y. Wang, L. Jiang, and Y. Wang, "Deep multi-agent reinforcement learning for highway on-ramp merging in mixed traffic," arXiv preprint arXiv:2105.05701, 2021.

[39] K. Brown, K. Driggs-Campbell, and M. J. Kochenderfer, "A taxonomy and review of algorithms for modeling and predicting human driver behavior. arxiv e-prints, article," arXiv preprint arXiv:2006.08832, 2020.

[40] A. Jami, M. Razzaghpour, H. Alnuweiri, and Y. P. Fallah, "Augmented driver behavior models for high-fidelity simulation

study of crash detection algorithms," 2022. [Online]. Available: https://arxiv.org/abs/2208.05540
[41] M. Lauer and M. Riedmiller, "An algorithm for distributed reinforcement learning in cooperative multi-agent systems," in In Proceedings of the Seventeenth International Conference on Machine Learning. Citeseer, 2000.
[42] S. Omidshafiei, J. Pazis, C. Amato, J. P. How, and J. Vian, "Deep decentralized multi-task multi-agent reinforcement learning under partial observability," in International Conference on Machine Learning. PMLR, 2017, pp. 2681–2690.
[43] Z. Constantinescu, C. Marinoiu, and M. Vladoiu, "Driving style analysis using data mining techniques," International Journal of Computers Communications & Control, vol. 5, no. 5, pp. 654–663, 2010.
[44] K. H. Beck, B. Ali, and S. B. Daughters, "Distress tolerance as a predictor of risky and aggressive driving," Traffic injury prevention, vol. 15, no. 4, pp. 349–354, 2014.
[45] R. Chandra, U. Bhattacharya, T. Mittal, A. Bera, and D. Manocha, "Cmetric: A driving behavior measure using centrality functions," in 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2020, pp. 2035–2042.
[46] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," Physical review E, vol. 51, no. 5, p. 4282, 1995.
[47] R. Parker and S. Valaee, "Cooperative vehicle position estimation," in IEEE International Conference on Communications, 2007.
[48] S. Baek, C. Liu, P. Watta, and Y. L. Murphey, "Accurate vehicle position estimation using a Kalman filter and neural network-based approach," in 2017 IEEE Symposium Series on Computational Intelligence (SSCI). IEEE, 2017, pp. 1–8.
[49] J. H. Painter, D. Kerstetter, and S. Jowers, "Reconciling steady-state Kalman and alpha-beta filter design," 1990.
[50] L. Chu, Y. Shi, Y. Zhang, H. Liu, and M. Xu, "Vehicle lateral and longitudinal velocity estimation based on adaptive kalman filter," in ICACTE 2010 - 2010 3rd International Conference on Advanced Computer Theory and Engineering, Proceedings, 2010.
[51] H. N. Mahjoub, B. Toghi, S. M. O. Gani, and Y. P. Fallah, "V2X system architecture utilizing hybrid gaussian process-based model structures," in 2019 IEEE International Systems Conference (SysCon), April 2019, pp. 1–7.
[52] N. Lee and K. M. Kitani, "Predicting wide receiver trajectories in american football," in 2016 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE, 2016, pp. 1–9.
[53] J. Morton, T. A. Wheeler, and M. J. Kochenderfer, "Analysis of recurrent neural networks for probabilistic modeling of driver behavior," IEEE Transactions on Intelligent Transportation Systems, vol. 18, no. 5, pp. 1289–1298, 2016.
[54] A. Alahi, V. Ramanathan, K. Goel, A. Robicquet, A. A. Sadeghian, L. Fei-Fei, and S. Savarese, "Learning to predict human behavior in crowded scenes," in Group and Crowd Behavior for Computer Vision. Elsevier, 2017, pp. 183–207.
[55] Y. Wang, Z. Wang, K. Han, P. Tiwari, and D. B. Work, "Gaussian process-based personalized adaptive cruise control," IEEE Transactions on Intelligent Transportation Systems, pp. 1–12, 2022.
[56] S. Mosharafian, M. Razzaghpour, Y. P. Fallah, and J. M. Velni, "Gaussian process based stochastic model predictive control for cooperative adaptive cruise control," in 2021 IEEE Vehicular Networking Conference (VNC), 2021, pp. 17–23.
[57] K. Das and A. N. Srivastava, "Block-gp: Scalable gaussian process regression for multimodal data," in 2010 IEEE International Conference on Data Mining. IEEE, 2010, pp. 791–796.
[58] J. M. Wang, D. J. Fleet, and A. Hertzmann, "Gaussian process dynamical models for human motion," IEEE transactions on pattern analysis and machine intelligence, vol. 30, no. 2, pp. 283–298, 2007.
[59] D. Guan, H. Zhao, L. Zhao, and K. Zheng, "Intelligent prediction of mobile vehicle trajectory based on space-time information," in 2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring). IEEE, 2019, pp. 1–5.
[60] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," Advances in neural information processing systems, vol. 27, 2014.
[61] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," Advances in neural information processing systems, vol. 28, 2015.
[62] C. E. Rasmussen and C. K. I. Williams, Gaussian Processes for Machine Learning. The MIT Press, 2006.
[63] B. Toghi, R. Valiente, D. Sadigh, R. Pedarsani, and Y. P. Fallah, "Social coordination and altruism in autonomous driving," arXiv preprint arXiv:2107.00200, 2021.
[64] R. Valiente, B. Toghi, R. Pedarsani, and Y. P. Fallah, "Robustness and adaptability of reinforcement learning-based cooperative autonomous driving in mixed-autonomy traffic," IEEE Open Journal of Intelligent Transportation Systems, vol. 3, pp. 397–410, 2022.
[65] E. Leurent, Y. Blanco, D. Efimov, and O.-A. Maillard, "Approximate robust control of uncertain dynamical systems," arXiv preprint arXiv:1903.00220, 2019.
[66] A. Kesting, M. Treiber, and D. Helbing, "General lane-changing model mobil for car-following models," Transportation Research Record, vol. 1999, no. 1, pp. 86–94, 2007.
[67] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," Physical review E, vol. 62, no. 2, p. 1805, 2000.
[68] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in Proceedings of the AAAI conference on artificial intelligence, vol. 30, no. 1, 2016.

Rodolfo Valiente is a Ph.D. candidate in Computer Engineering at the University of Central Florida. His research interests include connected autonomous vehicles, reinforcement learning, computer vision, and deep learning with a focus on the autonomous driving problem. He received a M.Sc. degree from the University of Sao Paulo (USP) in 2017 and his B.Sc. degree from the Technological University Jose Antonio Echeverria in 2014.

Mahdi Razzaghpour is a Ph.D. candidate in Computer Engineering at the University of Central Florida and a member of the Connected and Autonomous Vehicles Research Lab (CAVREL). His research interests include Reinforcement Learning, Machine Learning, and deep learning with a focus on the Cooperative driving problem. He received the M.Sc. degree in Computer Engineering from the University of Central Florida in 2021 and the B.Sc. degree in Electrical Engineering from Sharif University of Technology in 2019.

Behrad Toghi is a Ph.D. candidate at the University of Central Florida. He received the B.Sc. degree in electrical engineering from Sharif University of Technology in 2016 and has worked as a research intern at Mercedes-Benz R&D North America and Ford Motor Company R&D between 2018 and 2020. His work is in the intersection of artificial intelligence and cooperative networked systems with a focus on autonomous driving.

Ghayoor Shah is a Ph.D. candidate at the University of Central Florida. He received the B.Sc. degree in Computer Engineering from University of Illinois at Urbana-Champaign in 2018. He has previously worked as a mobility engineering intern at Phantom Auto and as a research intern at Ford Motor Company. His research interests include Connected and Autonomous Vehicles (CAVs), scalability analysis of V2X, and applications of artificial intelligence to cooperative driving.

Yaser P. Fallah is an Associate Professor in the ECE Department at the University of Central Florida. He received the Ph.D. degree from the University of British Columbia, Vancouver, BC, Canada, in 2007. From 2008 to 2011, he was a Research Scientist with the Institute of Transportation Studies, University of California Berkeley, Berkeley, CA, USA. His research, sponsored by industry, USDoT, and NSF, is focused on intelligent transportation systems and automated and networked vehicle safety systems.