Learning Interaction-Aware Guidance for Trajectory Optimization in Dense Traffic Scenarios

Bruno Brito[1], Achin Agarwal[1], and Javier Alonso-Mora[1]

***Abstract*—Autonomous navigation in dense traffic scenarios remains challenging for autonomous vehicles (AVs) because the intentions of other drivers are not directly observable and AVs have to deal with a wide range of driving behaviors. To maneuver through dense traffic, AVs must be able to reason how their actions affect others (interaction model) and exploit this reasoning to navigate through dense traffic safely. This paper presents a novel framework for interaction-aware motion planning in dense traffic scenarios. We explore the connection between human driving behavior and their velocity changes when interacting. Hence, we propose to learn, via deep Reinforcement Learning (RL), an interaction-aware policy providing global guidance about the cooperativeness of other vehicles to an optimization-based planner ensuring safety and kinematic feasibility through constraint satisfaction. The learned policy can reason and guide the local optimization-based planner with interactive behavior to pro-actively merge in dense traffic while remaining safe in case other vehicles do not yield. We present qualitative and quantitative results in highly interactive simulation environments (highway merging and unprotected left turns) against two baseline approaches, a learning-based and an optimization-based method. The presented results show that our method significantly reduces the number of collisions and increases the success rate with respect to both learning-based and optimization-based baselines.**

***Index Terms*—Deep reinforcement learning, dense traffic, motion planning, safe learning, trajectory optimization.**

I. Introduction

DESPITE recent advancements in autonomous driving solutions (e.g., Waymo [1], Uber [2]), driving in real-world dense traffic scenarios such as highway merging and unprotected left turns still stands as a hurdle in the widespread deployment of autonomous vehicles (AVs) [3]. Driving in dense traffic conditions is intrinsically an interactive task [4], where the AVs' actions elicit immediate reactions from nearby traffic participants and vice-versa. An example of such behavior is illustrated in Fig. 1, where the autonomous vehicle needs to perform a merging maneuver onto the main lane. To accomplish this task, it needs to first reason about the other driver's intentions (e.g., to yield or not to yield) without any explicit inter-vehicle communication. Then, it needs to know how to interact with multiple road-users and leverage other vehicles' cooperativeness to induce them to yield, such that they create room for the AV to merge safely.

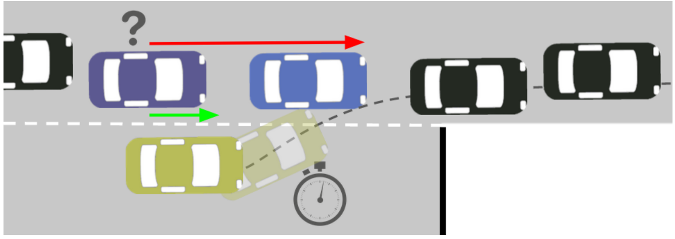


Fig. 1: Illustration of a dense on-ramp merging traffic scenario where the autonomous vehicle (yellow) needs to interact with other traffic participants in order to merge onto the main lane in a timely and safe manner. The potential follower (purple) may yield (green arrow) to the autonomous vehicle leaving space for the autonomous vehicle to merge or behave non-cooperatively (red arrow) to deter the autonomous vehicle from merging. To successfully merge, the autonomous vehicle needs to identify the cooperative ones by interacting with them without any explicit inter-vehicle communication.

The development of interaction-aware prediction models has been studied [5], [6], allowing AVs to reason about other drivers' intentions. In contrast, developing interactive motion planning algorithms that can reason and exploit other drivers cooperativeness is still challenging [7]. The majority of traditional motion planning methods are too conservative and fail in dense scenarios because they do not account for the interaction between the autonomous vehicle and nearby traffic [3], [8]. However, works that account for the interaction among agents do not scale for many agents due to the curse of dimensionality [9], [10], [11]. Deep Reinforcement Learning (DRL) methods can overcome the latter, but either do not provide any safety guarantees [12] or are overly conservative to ensure safety [13].

In this paper, we introduce an interactive Model Predictive Controller (IntMPC) for safe navigation in dense traffic scenarios. We explore the insight that human drivers communicate their intentions and negotiate their driving maneuvers by adjusting both distance and time headway to the other vehicles [14], [15]. Studies show that in dense traffic scenarios, such as merging and left-turning, cooperative or aggressive behavior is strongly connected to higher or smaller average distance and time headway [16], [17], respectively. These driving features (i.e., relative distance and time headway) can be directly translated into a velocity reference. Hence, we propose to learn, via Deep Reinforcement Learning (DRL), an interaction-aware policy as a velocity reference. This reference provides global

This work was supported by the Amsterdam Institute for Advanced Metropolitan Solutions and the Netherlands Organisation for Scientific Research (NWO) domain Applied Sciences (Veni 15916 and NWA.1292.19.298).

[1] The authors are with the Cognitive Robotics (CoR) department, Delft University of Technology, 2628 CD Delft, The Netherlands {bruno.debrito, j.alonsomora}@tudelft.nl

Video: https://youtu.be/xPomLRHs-II

guidance to a local optimization-based planner, which ensures that the generated trajectories are kino-dynamically feasible and safety constraints are respected. Our method leverages vehicles' interaction effects to create free-space areas for the AV to navigate and complete various driving maneuvers in cluttered environments. The main contribution of this work is an Interactive Model Predictive Controller (IntMPC) for navigation in dense traffic environments combining DRL to learn an interaction-aware policy providing global guidance (velocity reference) in the cost function to a local optimization-based planner.

Extensive simulation results show that our approach triggers interactive negotiating behavior to reason about the other drivers' cooperation and exploit their cooperativeness to induce them to yield while remaining safe.

II. Related Work

The literature devoted to the problem of modeling human interactions among traffic participants is vast [3] and includes rule-based, optimization-based, game theoretic and learning-based methods.

A. Traditional Methods

Traditional autonomous navigation systems typically employ a sequential planning architecture hierarchically decomposing the planning and decision-making pipeline into different blocks such as perception, behavioral planning, motion planning and low-level control [18]. For instance, rule-based methods translate implicit and explicit human-driving behavior into handcrafted functions describing a set of rules directly influencing the motion planning phase. In addition to rules, risk metrics can also be employed to generate cautious driving behavior [19]. For instance, [20] used predictive risk maps to plan the navigation behavior for an AV. These methods have demonstrated excellent ability to solve specific problems (e.g., precedence at an intersection followed by waiting for the availability of enough free space for the vehicle to pass safely) [21], [22], [23]. Nevertheless, these methods do not consider the interactions between multiple traffic participants and thus can fail in dense traffic scenarios.

B. Search-based Methods

The decision-making problem for autonomous navigation is inherently a Partially Observable Markov Decision Process (POMDP) because the other drivers' intentions are not directly observable but can be estimated from sensor data [24]. To improve decision-making and intention estimation, it has been proposed to incorporate the road context information [25]. To deal with a variable number of agents, dimensional reduction techniques have been employed to create a compressed and fixed-size representation of the other agents information [26]. Yet, solving a POMDP online can become infeasible if the right assumptions on the state, action and observation space are not made. For instance, [27] proposed to use Monte Carlo Tree Search (MCTS) algorithms to obtain an approximate optimal solution online and [28] improved the interaction modeling by proposing to feedback the vehicle commands into planning. These methods demonstrated promising results but are limited to environments for which they were specifically designed, demand high computational power and can only consider a discrete set of actions.

C. Optimization-based Methods

Optimization-based methods are widely used for motion planning since they allow to define collision and kino-dynamics constraints explicitly. These methods include receding-horizon control techniques which allow to plan in real-time and incorporate predicted information by optimizing over a time horizon [3], [29], [30]. However, these works employ simple prediction models and do not consider interaction. Recently, data-driven methods allow to generate interaction-aware predictions [31] that can be used for planning [32], [33]. However, these methods ignore the influence of the ego vehicle's actions in the planning phase struggling to find a collision free trajectory in highly dense traffic scenarios [34]. Not only motion planners must account for the interaction among the driving agents but also generate motions plans which respect social constraints. Hence, to generate socially compatible plans, Inverse Optimal Control techniques have been used to learn human-drivers preferences [35], [36]. These methods either fail to scale to interact with multiple agents [35] or can only handle a discrete set of actions [36] rendering them incapable to be used safely in highly interactive and dense traffic scenarios.

D. Game Theoretic Methods

Game Theoretic approaches such as [37] model the interaction among agents as a game allowing to infer the influence on each agent's plans. However, the task of modeling interactions requires the inter-dependency of all agents on each other's actions to be embedded within the framework. This results in an exponential growth of interactions as the number of agents increases, rendering the problem computationally intractable. Social Value Orientation (SVO) is a psychological metric used to classify human driving behavior. [7] models the interaction problem as a dynamic game given the other driver‘s SVO. Similarly, a unscented Kalman filter is used to iteratively update an estimate of the other drivers‘cost parameters [10]. Nevertheless, these approaches require local approximations to find a solution in a tractable manner. Cognitive hierarchy reasoning [38] allows to reduce the complexity of these algorithms by assuming that an agent performs a limited number of iterations of strategic reasoning. For instance, iterative level-k model based on cognitive hierarchy reasoning [38] has been used to obtain a near optimal policy for performing merge maneuvers [39] and lane change [40] in highly dense traffic scenarios. However, these approaches consider a discrete action space and do not scale well with the number of vehicles.

E. Learning-based Methods

Learning-based approaches leverage on large data collection to build interaction-aware prediction models [31] or to learn a driving policy directly from sensor data [41]. For instance, generative adversarial networks can be used to learn a driving policy imitating human-driving behavior [42]. Conditioning these policies on high-level driving information allows to use it for planning [43]. Moreover, to account for human-robot interaction these policies can be conditioned on the interaction history [9]. Yet, the deployment of these models can lead to catastrophic failures when evaluated in new scenarios or if the training dataset is biased and unbalanced [44].

Reinforcement Learning (RL) has shown great potential for autonomous driving in dense traffic scenarios [45], [46]. For example, DQN has been employed to learn negotiating behavior for lane change [47], [48] and intersection scenarios [49]. Yet, the latter consider a discrete and limited action space. In contrast, in [12] it is proposed to learn a continuous policy (jerk and steering rate) allowing to achieve smooth control of the vehicle. These methods are able to learn a working policy under highly interactive traffic conditions involving multiple entities. However, they fail to provide safety guarantees and reliability, rendering these methods vulnerable to collisions. Recently, a vast amount of works has proposed different ways to introduce safety guarantees of learned RL policies [50]. The key idea behind these works is to synthesize a safety controller when an unsafe action is detected by employing formal verification methods [51], computing offline safe reachability sets [52] or employing safe barrier functions [53]. To reduce conservativeness, [13] proposes to use Linear Temporal Logic to enforce safety probabilistic guarantees. However, *safe RL* methods do not account for interaction among the agents, being highly conservative in dense environments. Finally, close to our work, [54] learned a decision-making policy to select from a discrete and limited set of predefined constraints which ones to enable in an MPC formulation and thus, controlling the vehicle behavior applied to intersection scenarios. In contrast, we propose to learn a continuous interaction-aware policy providing global guidance to an MPC through the cost function.

F. Combining Optimization and Learning Methods

Recently, there is increasing interest in approaches combining optimization and learning methods [55]. For instance, optimization-based planning has been used to explore high-reward regions and distill the knowledge into a policy neural network [56], [57], [58]. Similar to our approach, [59] utilizes the RL policy during training to ensure exploration and employs an MPC to optimize sampled trajectories from the learned policy at test time. Similarly, [60] uses RL to learn a driving policy and employs an MPC as a supervisor to ensure safety. Moreover, policy networks have been used to generate proposals for a sampling-based MPC [61] or select goal positions from a predefined set [62]. In contrast, we propose to learn an interaction-aware policy to provide information through high-level decision variables directly in the MPC's cost function.

III. Problem Formulation

Throughout this paper, vectors are denoted in bold lowercase letters, $\mathbf{x}$, matrices in capital, M, and sets in calligraphic uppercase, $\mathcal{S}$. $\|\mathbf{x}\|$ denotes the Euclidean norm of $\mathbf{x}$ and $\|\mathbf{x}\|_Q = \mathbf{x}^T Q \mathbf{x}$ denotes the weighted squared norm. Variables $\{\mathbf{s}, \mathbf{a}\}$ denote the state and action used in the RL formulation, and $\mathbf{u}$ denotes the control command for the AV.

Consider a set $\mathcal{X}$ of n vehicles interacting in a dense traffic scenario comprising an autonomous vehicle (AV) and $n-1$ human drivers, henceforth referred to as other vehicles, exhibiting different levels of willingness to yield. The term ”vehicles” is used to collectively refer to the AV and other vehicles. At the beginning of an episode, the AV receives a global reference path $\mathcal{P}$ to follow from a path planner consisting of a sequence of M waypoints $\mathbf{p}^r_m = [x^r_m, y^r_m] \in \mathbb{R}^2$ with $m \in \mathcal{M} := \{1, \dots, M\}$. For each time-step k, the AV observes its state $\mathbf{s}_k$ and the states of other agents $\mathbf{S}_k = [\mathbf{s}^1_k, \dots, \mathbf{s}^{n-1}_k]$, then takes action $\mathbf{a}_k$, leading to the immediate reward $R(\mathbf{s}_k, \mathbf{a}_k)$ and next state $\mathbf{s}_{k+1} = f(\mathbf{s}_k, \mathbf{u}_k)$, under the dynamic model f[1] and controller model h, with $\mathbf{u}_k = h(\mathbf{s}_k, \mathbf{a}_k)$. The vehicle's state is defined as

$$\mathbf{s}^i_k = \{x_k, y_k, \psi_k, v_k\} \forall i \in \{0, \dots, n-1\}$$

where x_k and y_k are the Cartesian position coordinates, ψ_k the heading angle and v_k the forward velocity in a global inertial frame $\mathcal{W}$ fixed in the main lane (see Fig. 2). $\mathcal{A}^{\text{ego}}$ and $\mathcal{A}^i$ denote the area occupied by the AV and the i-th other vehicle, respectively. We aim to learn a continuous policy $\pi(\mathbf{a}_k|\mathbf{s}_k, \mathbf{S}_k)$ conditioned on the AV's and other vehicles' states minimizing the expected driving time $\mathbb{E}[t_g]$ for the AV to reach its goal position while ensuring collision-free motions, defined as the following optimization problem:

$$\begin{aligned}
\pi^* = &\underset{\pi}{\operatorname{argmin}} \quad \mathbb{E}\left[t_g \mid \pi(\mathbf{a}_k|\mathbf{s}_k, \mathbf{S}_k)\right] \\
\text{s.t.} \quad & \mathbf{s}_{k+1} = f(\mathbf{s}_k, \mathbf{u}_k), && \text{(1a)} \\
& \mathbf{u}_k = h(\mathbf{s}_k, \pi(\mathbf{a}_k|\mathbf{s}_k, \mathbf{S}_k)) && \text{(1b)} \\
& \mathcal{A}^{\text{ego}}_k \cap \mathcal{A}^i_k = \emptyset && \text{(1c)} \\
& \mathbf{u}_k \in \mathcal{U}, \ \mathbf{s}_k \in \mathcal{S}, \ \mathbf{a}_t \in \mathcal{A}, && \text{(1d)} \\
& \forall i \in \{1 \dots n-1\} \ \forall k \in \{0 \dots t_g\}
\end{aligned}$$

where (1a) are the kino-dynamic constraints, (1c) the collision avoidance constraints, and $\mathcal{S}$, $\mathcal{A}$ and $\mathcal{U}$ are the set of admissible states, actions, and control inputs (e.g., maximum vehicles' speed), respectively. We assume that each vehicle's current position and velocity are observed (e.g., from on-board sensor data) and no inter-vehicle communication.

IV. Interactive Model Predictive Control

A. Overview

This section introduces the proposed Interactive Model Predictive Control (IntMPC) framework for safe navigation in

[1] This is identical to the Vehicle Model used in the simulation defined in Section IV-C1

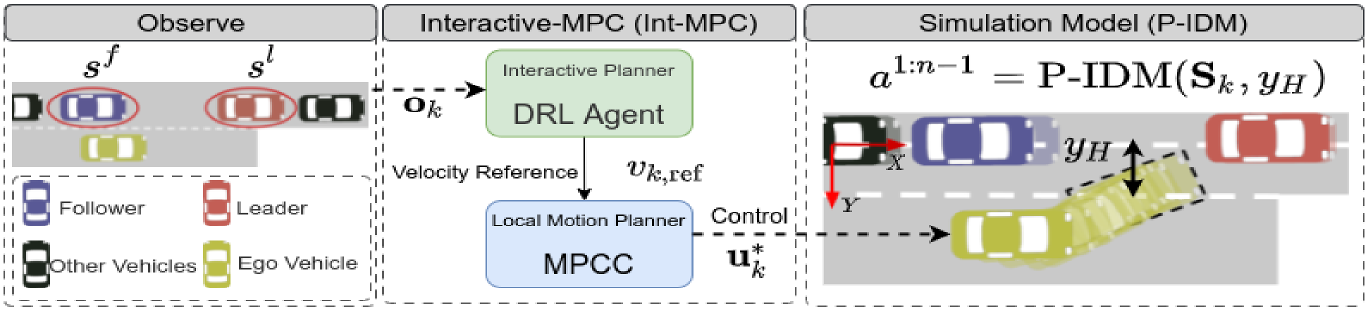


Fig. 2: Our proposed architecture comprises of three main modules: an Interactive Reinforcement Learner (DRL Agent), a Local Motion Planner (MPCC) and a Simulation Model (P-IDM). The AV observes the leader state s^l and follower state s^f relative to it, which serves as input to the Interactive Planner providing a reference velocity $v_{k,\text{ref}} = \pi(\mathbf{s}_k, \mathbf{S}_k)$ for the MPCC to follow. The MPCC then computes locally optimal sequence of control commands $\mathbf{u}^*_{0:H-1}$ minimizing a cost function $J(\mathbf{s_k}, \mathbf{u}_k)$ (See Section IV-C). The reference velocity v_{ref} allows to directly control the AV aggressiveness and thus, to control the interaction with the other vehicles. Finally, P-IDM then computes acceleration command for the other vehicles based on the estimated AV's motion plan (Section VI-H3).

dense traffic scenarios. Figure 2 depicts our proposed motion planning architecture incorporating three main modules: an interactive reinforcement learner, a local optimization planner, and an interactive simulation environment. Firstly, we define the RL framework to learn an interaction-aware navigation policy (Section IV-B), providing global guidance to a local optimization planner (Section IV-C). Secondly, we introduce the behavior module used to simulate dense traffic scenarios with various driving behavior, ranging from cooperative to non-cooperative. Here, we propose an expansion for the Intelligent Driver Model (IDM) model allowing the other vehicles to react to the other's predicted plans (SectionV). To finalize, we introduce our training algorithm to jointly train the interaction-aware policy and the local optimization planner (SectionIV-D). Our IntMPC enhances the AV with interactive behavior, exploiting the other traffic participants' interaction effects.

B. Interactive Planner

Here, we propose to use deep RL to learn an interaction-aware velocity reference exploiting the interaction effects between the vehicles and providing global guidance to a local optimization-based planner.

1) RL Formulation: The AV's observation vector is composed by the leader's (vehicle in front) and the follower's (vehicle behind the AV) state, $\mathbf{o}_k = [\mathbf{s}^l_k, \mathbf{s}^f_k]$, relative to AV's frame. To enable interactive behavior with the other traffic participants, we define the RL policy's action as a velocity reference to directly control the interaction at the merging point. High-speed values lead to more aggressive and low-speed to more conservative behavior, respectively. Hence, we consider a continuous action space $\mathcal{A} \subset \mathbb{R}$ and aim to learn the optimal policy π mapping the AV's state and observation to a probability distribution of actions.

$$\pi_\theta(\mathbf{s}_k, \mathbf{o}_k) = \mathbf{a}_k = v_{k,\text{ref}} \tag{2a}$$

$$\pi_\theta(\mathbf{s}_k, \mathbf{o}_k) \sim \mathcal{N}(\mu_k, \sigma_k) \tag{2b}$$

where θ are the policy's network parameters, $\mathcal{N}$ is a multivariate Gaussian density function, and μ and σ are the Gaussian's mean and variance, respectively.

We formulate a reward function to motivate progress along a reference path, to penalize collisions and infeasible solutions, and when moving too close to another vehicle. The reward function is the summation of the four terms described as follows:

$$R(\mathbf{s}_k, \mathbf{o}_k, \mathbf{a}_k) = \begin{cases} v_k & \\ r_{\text{infeasible}} & \\ r_{\text{collision}} & \text{if } \mathcal{A}^{\text{ego}}_k \cap \mathcal{A}^{\text{i}}_k \neq \emptyset \\ r_{\text{near}} & d_{\text{min}}(\mathbf{s}_k, \mathbf{s}^i_k) \leq \Delta d_{\text{min}} \end{cases} \tag{3}$$

where $c^{c,i}_k$ is the collision avoidance constraint between the AV and the vehicle i (Section IV-C3), $\mathcal{A}^{ego}_k \cap \mathcal{A}^i_k$ represents the common area occupied by the AV and the i-th other vehicle at step k. d_{min} is the minimum distance to the closest nearby vehicle i and Δd_{min} is a hyper-parameter distance threshold. The first term v_k is a reward proportional to the AV's velocity encouraging higher velocities and thus, encouraging interaction and minimizing the time to goal. The second $r_{\text{infeasible}}$, third $r_{\text{collision}}$ and fourth term r_{near} penalize the AV for infeasible solutions, collisions and for driving too close to other vehicles, respectively.

C. Local Motion Planner

Deep RL can be used to learn an end-to-end control policy in dense traffic scenarios [12], [46]. However, their sample inefficiency [63] and transferability issues [64] makes it hard to apply them in real-world settings. In contrast, optimization-based methods have been widely used and deployed into actual autonomous vehicles [65], [29]. Therefore, we employ Model Predictive Contour Control (MPCC) to generate locally optimal control commands following a reference path while satisfying kino-dynamics and collision avoidance constraints if a feasible solution is found. The reference path can be provided by a global path planner such as Rapidly-exploring Random Trees (RRT) [66].

1) Vehicle Model: We employ a kinematic bicycle model for the AV, described as follows:

$$\begin{aligned}
\dot{x} &= v\cos(\phi+\beta) \\
\dot{y} &= v\sin(\phi+\beta) \\
\dot{\phi} &= \frac{v}{l_r}\sin(\beta) \\
\dot{v} &= u^a \\
\beta &= \arctan\left(\frac{l_r}{l_f+l_r}\tan\left(u^\delta\right)\right)
\end{aligned} \tag{4}$$

where β is the velocity angle. The distances of the rear and front tires from the center of gravity of the vehicle are l_r and l_f, respectively, and are assumed to be identical for simplicity. The vehicle control input $\mathbf{u}$ is the forward acceleration u^a and steering angle u^δ, $\mathbf{u} = [u^a, u^\delta]$.

2) Cost Function: The local controller receives a velocity reference v_{ref}, from the Interactive Planner (Section IV-B), exploiting for the interaction effects of the AV in the other vehicles to maximize long-term rewards. To enable the AV to follow the reference path while tracking the velocity reference, we define the stage cost as follows:

$$\begin{aligned}
J(\mathbf{s}_{\mathbf{k}}, \mathbf{u}_k, \lambda_k) = &\left\|e^c_k(\mathbf{s}_k, \lambda_k)\right\|_{q_c} + \left\|e^l_k(\mathbf{s}_k, \lambda_k)\right\|_{q_l} \\
&+ \left\|v_{k,ref} - v_k\right\|_{q_v} + \left\|u^a_k\right\|_{q_u} + \left\|u^\delta_k\right\|_{q_\delta}
\end{aligned} \tag{5}$$

where $\mathcal{Q} = \{q_c, q_l, q_v, q_u, q_\delta\}$ denotes the set of cost weights and λ_k is the estimated progress along the reference path. To track the reference path closely, we minimize two cost terms: the contour error (e^c_k) and lag error (e^l_k). Contour error gives a measure of how far the ego vehicle deviates from the reference path laterally whereas lag error measures the deviation of the ego vehicle from the reference path in the longitudinal direction. For more details on path parameterization and tracking error, please refer to [29]. The third term, $\|v_{k,\text{ref}} - v_k\|$, motivates the planner to follow v_{ref} closely. Finally, to generate smooth trajectories, we add a quadratic penalty to the control commands u^a_k and u^δ_k.

3) Dynamic Obstacle Avoidance: The occupied area by the AV, $\mathcal{A}^{\text{ego}}(\boldsymbol{s}_k)$, is approximated with a union of n_c circles i.e $\bar{A}^{\text{ego}}(\boldsymbol{s}_k) \subseteq \bigcup_{c\in\{1,\ldots,n_c\}} \mathcal{A}_c(\boldsymbol{s}_k)$, where $\mathcal{A}_c$ is the area occupied for a circle with radius r. For each vehicle i, the occupied area $\mathcal{A}^i$ is approximated by an ellipse of semi-major axis a_i, semi-minor axis b_i and orientation ϕ. To ensure collision-free motions, we define a set of non-linear constraints imposing that each circle c of the AV with the elliptical area occupied by the i-th vehicle does not intersect:

$$c^{i,c}_k(\mathbf{s}_k, \mathbf{s}^i_k) = \begin{bmatrix}\Delta x^c_k \\ \Delta y^c_k\end{bmatrix}^{\mathrm{T}} R(\phi)^{\mathrm{T}} \begin{bmatrix}\frac{1}{\alpha^2} & 0 \\ 0 & \frac{1}{\beta^2}\end{bmatrix} R(\phi) \begin{bmatrix}\Delta x^c_k \\ \Delta y^c_k\end{bmatrix} > 1, \tag{6}$$

$\forall k \in \{0, \ldots, H\}$ and $\forall i \in \{1, \ldots, n-1\}$. The parameters Δx^c_k and Δy^c_k represent x-y relative distances in AV's frame between the disc c and the ellipse i for prediction step k. $R(\psi)$ is the rotation matrix. To guarantee collision avoidance we enlarge the other vehicle's semi-major and semi-minor axis with a r_{disc} coefficient, assuming $\alpha = a + r_{\text{disc}}$ and $\beta = b + r_{\text{disc}}$ as described in [67].

4) Road boundaries: We introduce constraints on the lateral distance (i.e., contour error) of the AV with respect to the reference path to ensure that the vehicle stays within the road boundaries [28]:

$$-w^{\text{road}}_{\text{left}} \leq e^c_k(\mathbf{s}_k) \leq w^{\text{road}}_{\text{right}} \tag{7}$$

where $w^{\text{road}}_{\text{left}}$ and $w^{\text{road}}_{\text{right}}$ are the left and right road limits, respectively.

5) MPC Formulation: We formulate the motion planning problem as a Receding Horizon Trajectory Optimization problem (8) with planning horizon H conditioned on the following constraints:

$$\begin{aligned}
\mathbf{u}^*_{0:H-1} = \min_{u_{0:H-1}} &\sum_{k=0}^{H-1} J(\mathbf{s}_k, \mathbf{u}_k, \lambda_k) + J(\mathbf{s}_H, \lambda_H) && \text{(8a)} \\
\text{s.t.} \quad & \mathbf{s}_{k+1} = f(\mathbf{s}_k, \mathbf{u}_k), && \text{(8b)} \\
& \lambda_{k+1} = \lambda_k + v_k \Delta t && \text{(8c)} \\
& -w^{\text{road}}_{\text{left}} \leq e^c(\mathbf{s}_k) \leq w^{\text{road}}_{\text{right}} && \text{(8d)} \\
& c^{i,c}_k(\mathbf{s}_k, \mathbf{s}^i_k) > 1 \ \ \forall c \in \{1, \ldots, n_c\}, && \text{(8e)} \\
& \mathbf{u}_k \in \mathcal{U}, \quad \mathbf{s}_k \in \mathcal{S}, && \text{(8f)} \\
& \forall k \in \{0, \ldots, H\}. && \text{(8g)}
\end{aligned}$$

where Δt is the discretization time and $\mathbf{u}^*_{0:H-1}$ the locally optimal control sequence for H time-steps. In this work, we assume a constant velocity model to estimate of the other vehicles' future positions, as in [67].

D. Training Procedure

In this work, we train the policy using Soft Actor-Critic (SAC) [68] to learn the policy's probability distribution parameters. SAC augments traditional RL algorithms' objective with the policy's entropy, embedding the notion of exploration into the policy while giving up on clearly unpromising paths [68]. We propose to jointly train the guidance policy with the local motion planner allowing the trained policy to directly implement our method on a real system and learn with the cases resulting in infeasible solutions for the optimization solver. In contrast to prior works on safe RL [53], during training, we do not employ collision constraints (Eq.8e), exposing the policy to dangerous situations or collisions which is necessary to learn how to interact with other vehicles closely.

Algorithm 1 describes the proposed training strategy. Each episode begins with the initialization of all vehicle's states (see Sections VI-B and VI-C for more details). Every K cycles, we sample a reference velocity v_{ref} from the policy π_θ. Querying the interaction-aware policy every K control cycles helps to stabilize the training procedure and better assess the policy's impact on the environment (see Section VI-H2). Then,

Algorithm 1 Training Procedure

```
Inputs: planning horizon $H$, initial policy's parameters $\theta$, Q-functions' parameters $\{\phi_1, \phi_2\}$, number of training episodes $n_{\text{episodes}}$, number of vehicles $n$, reward function $R(\mathbf{s}_k, \mathbf{o}_k, \mathbf{a}_k)$ and number of control steps $K$
Initialize initial states: $\{\mathbf{s}_0, \dots, \mathbf{s}_0^{n-1}\} \sim \mathcal{S}$
Initialize replay buffer: $\mathcal{D} \leftarrow \emptyset$
while $episode < n_{\text{episodes}}$ do
    Get observation $\mathbf{o}_k$ and AV's state $\mathbf{s}_k$
    if $k \bmod K == 0$ then
        Sample velocity reference for the AV:
        $v_{k,\text{ref}} \sim \pi_\theta(\mathbf{s}_k, \mathbf{o}_k)$
    end if
    Solve the optimization problem of Eq.8 without collision constraints (Eq.8e) :
        $\mathbf{u}^*_{k:k+H}$ = MPCC$(v_{k,\text{ref}}, \mathbf{s}_k, \mathbf{o}_k)$
    Estimate AV's lateral position:
        $\tilde{y}_H$ = PredictionModel$(v_k, \mathbf{s}_k, \mathbf{o}_k)$ (Section VI-H3)
    $\{\mathbf{s}_{k+1}$, done, $\mathbf{r}_k\}$ = Step$(\mathbf{s}_k, \mathbf{u}_k)$
    Store $(\mathbf{s}_k, \mathbf{a}_k, r_k, \mathbf{s}_{k+1}, \text{done})$ in replay buffer $\mathcal{D}$
    if done then
        $episode$ += 1
        Initialize: $\{\mathbf{s}_0, \dots, \mathbf{s}_0^n\} \sim \mathcal{S}$
    end if
    if it's time to update then
        SAC training [68]
    end if
end while
return $\{\theta, \phi_1, \phi_2\}$
```

the MPCC computes a locally optimal sequence of steering and acceleration commands $u^*_{0:H-1}$ for the AV. If a feasible solution is found, we apply the first control command of the sequence and re-compute the motion plan in the next cycle considering new observations. If no feasible solution is found, we apply a braking command. Training the interaction-aware policy with the MPCC controller enables the policy to account for the controller and AV constraints. Afterward, the P-IDM computes an action for each vehicle on the main lane while being aware of the AV on the adjacent lane. An episode is over if: the AV reaches the goal position (finishes merging or turning left); the AV collides with another vehicle; it does not finish the maneuver in time (i.e., timeout). Finally, to update the policy's distribution parameters, we employ the Soft Actor-Critic (SAC) [68] method. We refer the reader to [68] for more details about the learning method's equations. Please note that our approach is agnostic to which RL algorithm we use.

E. Online Planning

Algorithm 2 describes our Interactive Model Predictive Controller (IntMPC) algorithm. For every step k, we first obtain a velocity reference, v_{ref}, from the trained policy. Then, by solving the MPCC problem (Eq. (8)), we obtain a locally optimal sequence of control commands $\mathbf{u}^*_{k:k+H}$. Finally, if the MPCC plan is feasible we employ the first control command, $\mathbf{u}^*_k$, and re-compute a new plan considering the new observations following a receding horizon control strategy. Else, we apply a braking command, $\mathbf{u}_{\text{safe}}$.

Algorithm 2 Int-MPC

```
Inputs: AV's state $\mathbf{s}_k$, observation $\mathbf{o}_k$ and reference path $\mathbf{p}^r_m = [x^r_m, y^r_m] \in \mathbb{R}^2$ with $m \in \mathcal{M} := \{1, \dots, M\}$ waypoints.
for $k = 0, 1, 2, \dots$ do
    Get observation $\mathbf{o}_k$ and AV's state $\mathbf{s}_k$
    Sample velocity reference for the AV:
        $v_{k,\text{ref}} = \pi_\theta(\mathbf{s}_k, \mathbf{o}_k)$
    Compute MPCC trajectory by solving Eq.8:
        $\mathbf{u}^*_{k:k+H}$ = MPCC$(v_{k,\text{ref}}, \mathbf{s}_k, \mathbf{o}_k)$
    if $\mathbf{u}^*_{k:k+H}$ is feasible then
        Apply $\mathbf{u}^*_k$
    else
        Apply $\mathbf{u}_{\text{safe}}$
    end if
end for
```

V. Modeling the Behavior of other Drivers

We aim to simulate dense and complex negotiating behavior with varying degrees of willingness to yield. For instance, in a typical dense traffic scenario (e.g., on-ramp merging), human drivers trying to merge onto the main lane need to leverage other drivers' cooperativeness to create obstacle-free space to merge safely. In contrast, drivers on the main lane exhibit different levels of willingness to yield. Some drivers stop as soon as they spot the other vehicle on the adjacent lane (Cooperative). Other drivers ignore the other vehicles entirely and may even accelerate to deter it from merging (Non-Cooperative). Moreover, they also consider an internal belief about the other vehicle's motion plan on the adjacent lane in their decision-making process at the merging point. Here, we introduce the Predictive Intelligent Driver Model (P-IDM) to control the longitudinal driving behavior of the other vehicles, built on the Intelligent Driver Model (IDM) [69]. Our proposed model consists of three main steps: leader and follower selection, other vehicles' motion estimation, and control command computation.

1) Leader & Follower Selection: For each vehicle, the model assigns a leader, denoted with up-script l, and a follower, denoted with up-script f. Consider $\mathcal{X}_i^{l,\text{IDM}}$ as the set of potential leaders following the IDM model for the vehicle i, then

***Definition* 1.** *IDM: A set $\mathcal{X}_i^{l,IDM} \subseteq \mathcal{X}$ is the set of possible leaders for the vehicle i if $\forall j \in [0, n-1], j \neq i : x^j_k > x^i_k$ and $y^i_k < c_i$.*

***Definition* 2.** *IDM: A set $\mathcal{X}_i^f \subseteq \mathcal{X}$ is the set of possible followers for the vehicle i if $\forall j \in [0, n-1], j \neq i : x^j_k < x^i_k$ and $y^i_k < c_i$.*

where c_i is a hyper-parameter threshold used to model cooperation (Section VI-C) [12]. Fig. 3 shows an example of

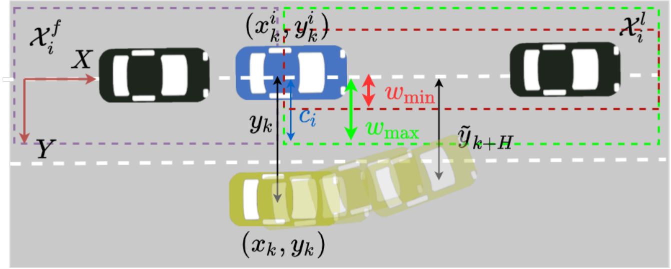


Fig. 3: Leader & Follower Selection Process. The AV is depicted in yellow, the i-th interacting vehicle in blue, and the i-th vehicle's follower and leader in black. (x_k, y_k) are the x-y position coordinates in the main lane frame of the AV and (x_k^i, y_k^i) of the i-th vehicle on the main lane at time-step k. Dashed purple represents the followers' area set. Dashed red and green represent the leader's area set. To model mixed driving behavior, the i-th vehicle cooperation coefficient c_i is randomly sampled from a uniform bounded distribution $c_i \sim \mathcal{U}([w_{\min}, w_{\max}])$ (defined in Section VI-C). $w_{\max}$ and $w_{\min}$ represents a maximum and minimum distance between the center of the current lane and the adjacent lane.

the leader's and follower's sets for the merging scenario as well as the physical representation of the cooperation coefficient c_i. In the IDM, the leaders' and followers' sets are defined based on the vehicle's current lateral position, y_k^i, leading to reactive behavior. In contrast, we propose to define the leader's and follower's sets based on the estimated lateral position at time-step H, $\tilde{y}_H^i$, as it follows

***Definition* 3.** *P-IDM: A set $\mathcal{X}_i^{l,\text{P-IDM}} \subseteq \mathcal{X}$ is the set of possible leaders for the vehicle i if $\forall j \in [0, n-1], j \neq i : x_k^j < x_k^i$ and $\left\|\tilde{y}_H^i\right\| < c_i$.*

Employing the predicted lateral position $\tilde{y}_H$ instead of the current lateral position y_k allows to elicit non-reactive behavior from the other vehicles. The leader for the vehicle i is defined as it follows

***Definition* 4.** *A vehicle $j \in \mathcal{X}_i^l$ is the leader of vehicle i if $\forall m \in \mathcal{X}_i^l, m \neq j : \left\|x_k^j - x_k^i\right\| \leq \left\|x_k^m - x_k^i\right\|$.*

where $\mathcal{X}_i^l$ is either $\mathcal{X}_i^{l,\text{P-IDM}}$ or $\mathcal{X}_i^{l,\text{IDM}}$ depending on the model used. Please note that the followers' set definition is the same for the IDM and P-IDM model.

To model mixed driving behavior, c_i is sampled from a uniform bounded distribution $c_i \sim \mathcal{U}([w_{\min}, w_{\max}])$ (defined in Section VI-C). $w_{\max}$ and $w_{\min}$ represents a maximum and minimum distance between the center of the current lane and the adjacent lane, as depicted in Fig. 3, respectively. Moreover, the c_i values' range plays an essential role in the final policy's behavior by controlling the proportion of cooperative and non-cooperative vehicles encountered by the AV during training resulting in a more aggressive or conservative final policy.

2) Motion Plan Estimation: To enhance the IDM model with predictive driving behavior, we propose to condition the IDM on the beliefs of the other drivers' motion plans. Specifically, we assume that each vehicle on the main lane maintains an internal belief about the AV's motion plan (on the adjacent lane)[2]. To estimate the AV's motion plans, different prediction models can be employed (e.g., constant velocity model). Later, in Section VI-H3, we investigate our method's performance for different prediction models.

3) Control Command Computation: For each time-step k and for each vehicle i, the acceleration control is computed depending on the vehicle's velocity v_k^i and current distance to the leader $\Delta x_k^i = \left\|(x_k^i, y_k^i) - (x_k^l, y_k^l)\right\|$:

$$u_k^{a,i} = a_{\max}\left[1 - \left(\frac{v_k^i}{v^*}\right)^4 - \left(\frac{s^*\left(v_k^i, \Delta v_k^i\right)}{\Delta x_k^i}\right)^2\right] \qquad (9)$$

where s^* is the desired minimum gap, $a_{\max}$ the maximum acceleration, $\Delta v_k^i = v_k^i - v_k^l$ the i-th vehicle approach rate to the preceding vehicle, and v^* the desired velocity. Please note that we only do longitudinal control for the other vehicles on the main lane by employing Eq. (9). For the AV, we employ a local optimization-based planner (Section IV-C) for steering and acceleration control.

VI. Experiments

This section presents simulation results for two dense traffic scenarios (Section VI-B) considering different cooperation settings for the other vehicles (Section VI-C). First, we present qualitative (Section VI-F) and performance results (Section VI-G) of our approach against two baselines:

- DRL : state-of-the-art Deep Reinforcement Learning approach, SAC [68], learning a continuous policy controlling the AV's forward velocity.
- MPCC [29]: Model Predictive Contour Controller with a constant velocity reference.

After, we provide an ablation study analyzing our method's design choices (Section VI-H). All controller parameters were manually tuned to get the best possible performance.

A. Experimental setup

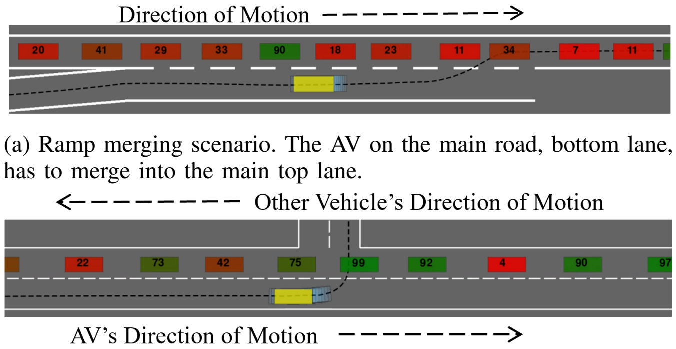
Simulation results were carried out on an Intel Core i9, 32GB of RAM CPU @ 2.40GHz taking approximately 20 hours to train, approximately 20 million simulation steps. The non-linear and non-convex MPCC problem of Eq. (8) was solved using the ForcesPro [70] solver. Our simulation environment, P-IDM, builds on an open-source highway simulator [71] expanding it to incorporate complex interaction behavior. Hyperparameters values can be found in Table I. Our motion planner and simulation environment are open source[3].

[2] For the Ramp Merging scenario (detailed in Sec. VI-B1), the current lane corresponds to the main lane whereas the adjacent lane refers to the merge lane whereas for the Unprotected Left Turn scenario (detailed in Section VI-B2), the current lane refers to the top lane and the adjacent lane corresponds to the bottom lane.

[3] https://github.com/tud-amr/highway-env

TABLE I: Hyperparameters

Hyperparameter	Value
Planning Horizon	1.5 s
Number of Stages N	15
Number of parallel workers	7
Q neural network model	2 dense layers of 256
Policy neural network model	2 dense layers of 256
Activation units	Relu
Training batch size	2048
Discount factor	0.99
Optimizer	Adam
Initial entropy weight (α)	1.0
Target update (τ)	5×10^{-3}
Target entropy lower bound	-1.0
Target network update frequency	1
Learning rate	3×10^{-4}
Replay buffer size	10^6
$r_{\text{infeasible}}$	-1
$r_{\text{collision}}$	-300
r_{near}	-1.5
$\{q_c, q_l, q_v, q_u, q_\delta\}$	$\{0.1, 0.2, 1.0, 0.1, 0.1\}$
K	2
Timestep	0.1 s
Control cycle	0.2 s



(a) Ramp merging scenario. The AV on the main road, bottom lane, has to merge into the main top lane.

(b) Unprotected left-turn scenario: The AV on the main road, bottom lane, has to make a left-turn while avoiding collision with the other vehicles on the main road, top lane.

Fig. 4: Evaluation environments: The AV is depicted in yellow and the reference path is depicted by the black dashed line. Each other vehicle is assigned with a color transitioning from red (i.e., non-cooperative) to green (i.e., cooperative). The number displayed by each vehicle represents its cooperation coefficient.

B. Driving scenarios

We consider two densely populated driving scenarios: merging on a highway and unprotected left turn. The vehicles are modeled as rectangles with 5 m length and 2 m width. For each episode, the initial distance between the other vehicles is drawn from a uniform distribution ranging from [7, 10] m. Their initial and target velocities are sampled from a uniform distribution, $v_0^{0:n} \sim \mathcal{U}(3, 4)$ m/s. This initial configuration prevents early collisions while ensuring no gaps of more than 2 meters [72], typical of dense traffic scenarios. These scenarios compel the AV to leverage other vehicles' cooperativeness while also exposing it to a myriad of critical scenarios for the final policy's performance.

1) Ramp Merging: Fig. 4a depicts an instance of the merging scenario. It comprises two lanes: the main lane and a merging lane. At the beginning of each episode, the main lane is populated with the other vehicles, moving from left to right. In contrast, the merge lane only includes the AV.

2) Unprotected Left Turn: Fig. 4b illustrates the unprotected left turn scenario. It consists of two roads: the main road and the left road perpendicular to each other. The main road is populated with the other vehicles (on the top lane) and the AV (on the bottom lane). The other vehicles move from right to left on the main road, whereas the AV is initialized at the bottom lane of the main road, and its objective is to take an unprotected left turn onto the left road.

C. Evaluation Scenarios

We present simulation results considering different settings for the other vehicles' cooperation coefficient:

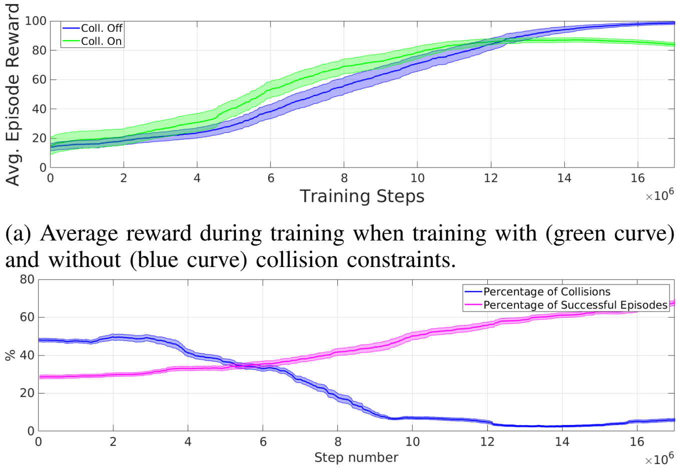
- Cooperative: In this scenario, most vehicles are cooperative ($c^i \sim \mathcal{U}(2, 4)$ m), implying that as soon as the AV shows intentions of merging into the main lane, the other vehicle starts considering the AV as its new leader, leaving space for it to merge into the main lane. This evaluation scenario helps in assessing the merging speed of the policy.
- Non-Cooperative: This scenario comprises mostly non-cooperative vehicles ($c^i \sim \mathcal{U}(0, 2)$ m), meaning that the other vehicles would not stop for the AV unless the AV's lateral horizon state is in the top lane. This scenario explicitly assesses the policy's aggressiveness. In these scenarios, the best option for the AV is to stop and wait for gaps and then merge in as quickly as possible.
- Mixed: This traffic scenario involves agents with varying degrees of cooperativeness ($c^i \sim \mathcal{U}(0, 4)$ m), featuring a continuous transition from cooperative to non-cooperative vehicles. Here, the goal is to assess how differently the AV behaves with cooperative and non-cooperative vehicles.

During training, we consider a mixed setting for the other vehicles. Rule based methods such as IDM, MOBIL fail in highly dense traffic conditions and thus have not been included for evaluation purposes [12].

D. Evaluation metrics

To evaluate our proposed method, we employ the following evaluation metrics:

- *Success Rate*: Percentage of successful episodes. An episode is deemed successful if the AV is able to merge on to the main highway or perform a left term without colliding and before timeout.
- *Collisions*: Percentage of episodes resulting in collision.
- *Timeout*: Percentage of episodes in which the AV did not reach the goal within the maximum specified time. This metric does not include those episodes that resulted in collision.



(a) Average reward during training when training with (green curve) and without (blue curve) collision constraints.

(b) Percentage of successful and failure episodes during training when training without collision constraints.

Fig. 5: Training performance.

- *Time-to-goal*: Time in seconds for the AV to reach the goal position.

E. Training Procedure

The interactive policy was trained considering a mixed setting of other vehicles following a P-IDM model with CV predictions. Fig. 5 shows the performance of the learning policy during training. The top sub-plot (Fig. 5a) shows the average reward evolution when training a policy with and without collision constraints. Training with collision constraints enables faster growth of the average rewards until 12×10^6 training steps. This phenomenon happens because the policy's task is simpler as the local controller overwrites the policy's actions that may lead to a collision. Nevertheless, employing collision constraints does not allow the AV to interact closely with the other vehicles. Hence, after the 12×10^6 training steps, the policy trained without collision constraints achieves a higher average reward. The bottom sub-plot (Fig. 5b) shows the percentage of failure and collision episodes during training, demonstrating that the learning policy effectively decreases the percentage of collisions while increasing the rate of successful episodes throughout training.

F. Qualitative Results

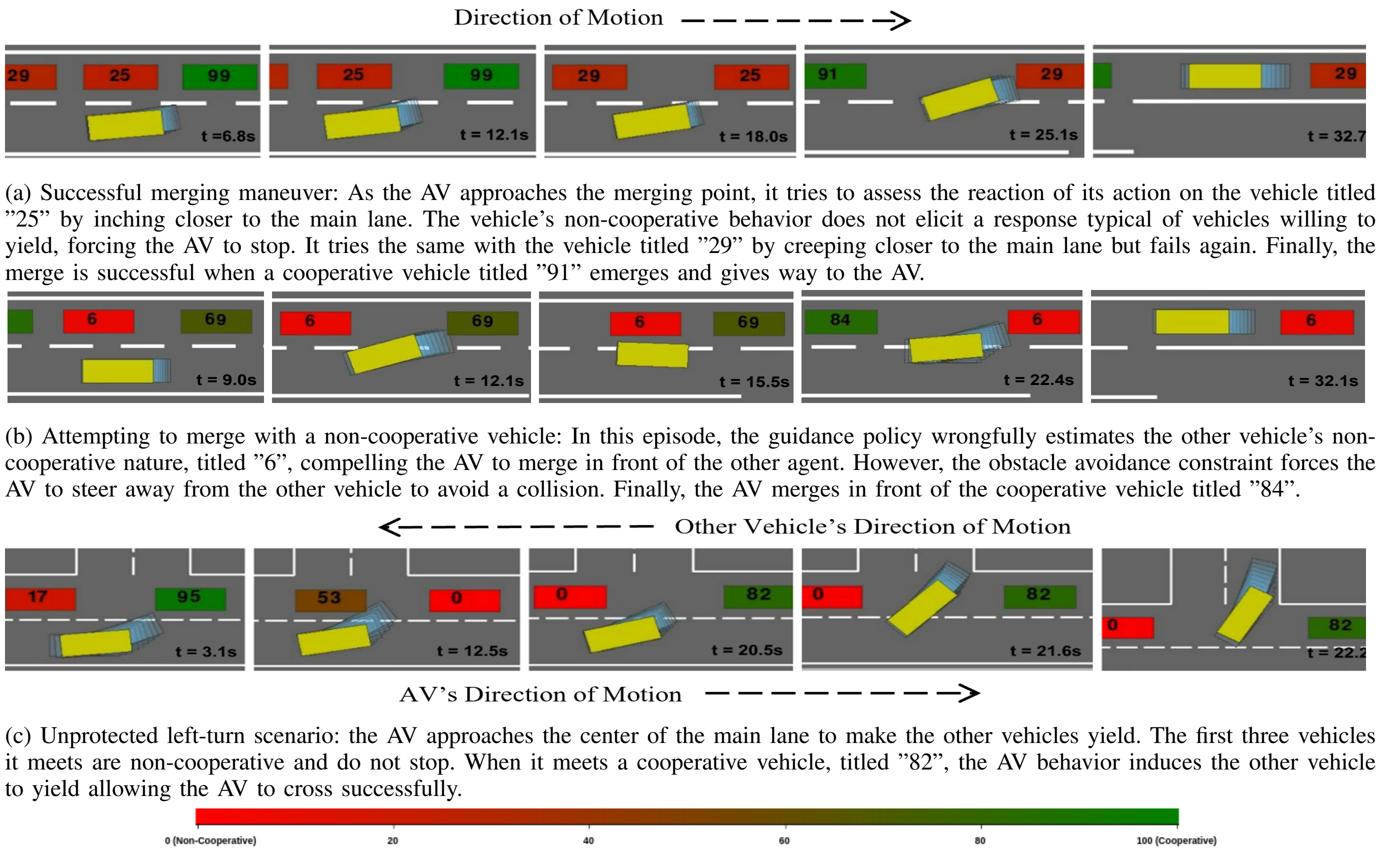
Fig. 6 presents visual results for our method for the merging and left-turn scenarios. In Fig. 6a, the AV successfully merged onto the main lane by leveraging other vehicles' cooperativeness. In contrast, in Fig. 6b, we highlight a critical advantage of our framework: the ability to perform a collision avoidance maneuver when the guidance policy wrongly estimates the other vehicle's cooperativeness. In this episode, at 12.1 s, the AV initiates a merging maneuver. However, the non-cooperative vehicle does not allow it. The local planner aborts and starts a collision avoidance maneuver at 15.5 s, merging successfully later when encountering a cooperative vehicle at 22.4 s. Finally, Fig. 6c shows the AV performing an unprotected left-turn maneuver successfully. The presented qualitative results show that our proposed method enables the AV to safely and efficiently navigate in dense traffic scenarios. We refer the reader to the video accompanying this paper for more qualitative results.

G. Quantitative Results

Aggregated results in Table II show that our method outperforms the baseline methods in terms of successful merges and number of collisions considering different settings for the other vehicles' behaviors (i.e., cooperative, mixed and, non-cooperative). The combined capability of interactive RL policy to implicitly embed inter-vehicle interactions into the velocity's policy and the safety provided by the collision avoidance constraints allows our method to succeed in all the environments. The optimization-based baseline (MPCC) shows poor performance for all settings, i.e., high collision rate. The reason is the lack of assimilation of inter-vehicle interactions into the policy and a tracking velocity reference error term in the cost function formulation that motivates the AV to keep the same velocity disregarding the nearby vehicles' cooperativeness. The DRL baseline achieves significantly higher performance, i.e., lower collision rate and a higher number of successful episodes. Nevertheless, it still leads to a significant number of collisions due to the lack of collision avoidance constraints to ensure safety when closely interacting with other vehicles. This demonstrates that employing collision constraints for navigation in dense traffic scenarios leads to superior performance over solely learning-based methods. In contrast, safety comes with the cost of larger average time-to-goal because the AV has to find the right time-window to merge.

Table III presents statistical results of the *time-to-goal* for all methods. To evaluate the statistical significance, we performed pairwise Mann–Whitney U-tests between each method, considering a 95% confidence level. The results show statistical significance for the MPCC's results against the other methods for cooperative and mixed settings. In contrast, there is no statistical difference in terms of *time-to-goal* between the DRL and IntMPC. Similarly, between all methods in non-cooperative environments. The presented results show that employing collision avoidance constraints do not increase the average *time-to-goal* while improving safety. Moreover, in non-cooperative environments, all methods achieve comparable performance in terms of *time-to-goal*.

To demonstrate our policy's ability to leverage agents' cooperativeness explicitly, we evaluate 600 episodes in a mixed scenario where we track the other vehicle' cooperation level in front of which the AV performs a successful merging maneuver. Fig. 7 depicts a histogram illustrating the number of successful episodes per cooperation coefficient, demonstrating that our method mostly merges with cooperative vehicles. A small number of successful merges can be seen with non-cooperative vehicles as well. This behavior can be attributed to the random sampling of IDM parameters resulting in different agents' acceleration values. Thus, the agents might leave a gap



(a) Successful merging maneuver: As the AV approaches the merging point, it tries to assess the reaction of its action on the vehicle titled ”25” by inching closer to the main lane. The vehicle's non-cooperative behavior does not elicit a response typical of vehicles willing to yield, forcing the AV to stop. It tries the same with the vehicle titled ”29” by creeping closer to the main lane but fails again. Finally, the merge is successful when a cooperative vehicle titled ”91” emerges and gives way to the AV.

(b) Attempting to merge with a non-cooperative vehicle: In this episode, the guidance policy wrongfully estimates the other vehicle's non-cooperative nature, titled ”6”, compelling the AV to merge in front of the other agent. However, the obstacle avoidance constraint forces the AV to steer away from the other vehicle to avoid a collision. Finally, the AV merges in front of the cooperative vehicle titled ”84”.

(c) Unprotected left-turn scenario: the AV approaches the center of the main lane to make the other vehicles yield. The first three vehicles it meets are non-cooperative and do not stop. When it meets a cooperative vehicle, titled ”82”, the AV behavior induces the other vehicle to yield allowing the AV to cross successfully.

Fig. 6: All the scenarios employ the P-IDM model (Section V) to simulate the other vehicles. The AV is represented in yellow, whereas the future states, as computed by the MPCC, are plotted in light blue. Each other vehicle is assigned with a color transitioning from red (i.e., non-cooperative) to green (i.e., cooperative) to highlight the other vehicles' cooperativeness. The number displayed by each other vehicle represents its cooperation coefficient. All the numbers in between show a continuous transition from non-cooperative (0) to cooperative (100).

TABLE II: Statistic results for 1200 runs of proposed method (IntMPC) compared to baselines (MPCC [29] and DRL [68]) considering three different settings for the other vehicles (Section VI-C): percentage of success, collisions and timeout episodes.

	Cooperative			Mixed			Non-cooperative		
	Success(%)	Collision(%)	Timeout(%)	Success(%)	Collision(%)	Timeout(%)	Success(%)	Collision(%)	Timeout(%)
MPCC [29]	78.0	11.0	10.0	62.0	22.0	16.0	26.0	57.0	19.0
RL [68]	86.0	3.0	11.0	69.0	2.0	29.0	31.0	5.0	64.0
Int-MPC	**86.0**	**0.0**	14.0	**70.0**	**0.0**	30.0	**36.0**	**0.0**	64.0

TABLE III: Statistical results on the *time-to-goal* [s]. Only the episodes where all methods are successful are considered in the presented results. Bold values represent the results with statistical significance.

	Cooperative	Mixed	Non-cooperative
MPCC [29]	**34.7 ± 4.1**	**35.9 ± 6.9**	40.1 ± 5.8
DRL [68]	37.5 ± 7.8	37.9 ± 6.9	41.8 ± 7.4
IntMPC	37.6 ± 8.0	37.7 ± 6.0	41.0 ± 7.3

big enough for the AV to merge onto the lane when moving from a standstill position.

Fig. 8 presents the number of infeasible solutions for our method (IntMPC) and the MPCC baseline. To jointly train the RL policy with the local controller and penalize the state and action tuples resulting in the solver infeasibility, significantly reduces the number of infeasible solutions. Finally, in terms of computation performance, our policy's network has an average computation time of 1.35 ± 0.5 ms. To solve the IntMPC's optimization problem (Eq. (8)) takes on average 3.0 ± 1.35 ms for all experiments. There was no statistical difference on the policy's and solver's computation times for the different settings of the other vehicles (e.g., cooperative, mixed and non-cooperative). These results demonstrate out method's real-time applicability.

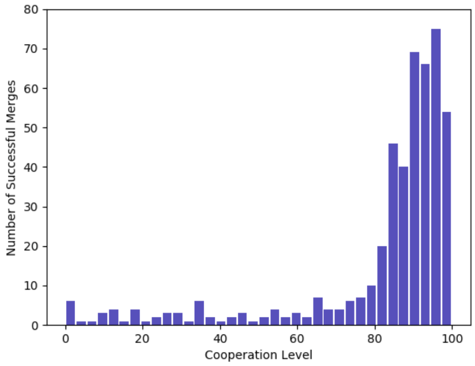


Fig. 7: Histogram of the agents' cooperation level (0 - non cooperative, 100 - cooperative) in front of which the ego vehicle was able to merge successfully.

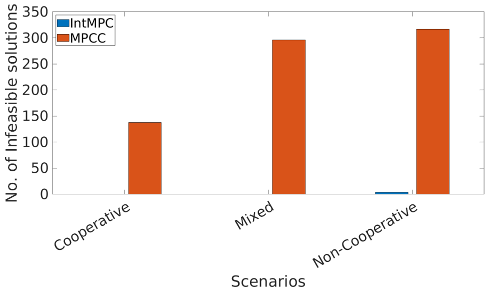


Fig. 8: Number of infeasible solutions encountered by the solver for our method (IntMPC) versus the optimization-based baseline (MPCC).

TABLE IV: Ablation study of the MPCC's parameters considering a mixed setting for the other vehicles.

	Success (%) / Collision (%) / Timeout (%)		
	$q_v = 0.1$	$q_v = 1.0$	$q_v = 10.0$
$v_{\text{ref}} = 2$ m/s	58 / 26 / 16	**62 / 22 / 16**	58 / 34 / 8
$v_{\text{ref}} = 3$ m/s	48 / 26 / 26	62 / 29 / 9	55 / 42 / 3
$v_{\text{ref}} = 4$ m/s	56 / 25 / 19	57 / 35 / 8	53 / 46 / 1

H. Performance analysis

This section investigates the impact of two critical design choices for our proposed approach: MPCC's parameters and using a different number of control cycles per RL policy query. Moreover, we evaluate our method's robustness to different prediction models used by the other vehicles to estimate the AV's motion plans. To finalize, we compare the risk-level that the AV takes with our method and the two planning baselines.

1) Local controller parameters: The MPCC's parameters (i.e., weights and velocity reference) highly influence the local planner's performance. Here, we study the two key components controlling the AV's interaction with the other vehicles: the velocity tracking weight (q_v) and the reference velocity (v_{ref}). Table IV presents performance results for different q_v and v_{ref} values. Increasing the reference velocity combined with high q_v values generates more aggressive behavior and significantly reduces the timeout rate. However, it also increases the collision rate. In contrast, low q_v values weaken the influence of the velocity reference on the MPCC performance. The presented results demonstrate that fine-tuning the MPCC's weights and velocity reference is insufficient for safe and efficient navigation in dense traffic environments, supporting the need for an interaction-aware velocity reference. $q_v = 1.0$ and $v_{\text{ref}} = 2$ m/s lead to the best performance, i.e., higher success rate and lower collision and timeout rate. For the following experiments, we use $q_v = 1.0$ and a velocity reference of $v_{\text{ref}} = 2$ m/s for the MPCC baseline.

2) Hyperparameter selection: A key design choice of the proposed framework is the number of control cycles per policy query, denoted by K. For instance, for $K = 1$, we query the policy network for a new velocity reference for each control cycle, while for $K = 4$, we use the same queried velocity reference during 4 control cycles. Here, we study the impact on the learned policy's performance for $K = \{1, \dots, 4\}$. During testing, all the policies are evaluated using $K = 1$. Table V summarizes the obtained performance results. The policy trained with $K = 2$ outperforms the other policies in terms of success and collision rate. The policy trained with $K = 1$ elicits an overly aggressive response from AV, evident from a high collision rate and a low timeout percentage. In contrast, higher K values lead the AV to exhibit an overly conservative behavior, thus, higher timeout percentage. This behavior can be attributed to the long duration for which the same action is applied after querying the interactive policy. For instance, using a large velocity reference value during many control cycles highly increases the collision likelihood at the merging point. This compels the RL algorithm to learn biased policy towards low-velocity references to avoid an impending collision resulting in an overly conservative behavior. Finally, the policy trained with $K = 2$ elicits a balanced response from the AV that is neither too conservative nor too aggressive, resulting in a high success rate and a low collision rate for all the scenarios.

3) Simulation environment: This work introduces an IDM variant enhancing the other vehicles with anticipatory behavior. Our proposed model (P-IDM in Section V) relies on the assumption that the other vehicles can infer the AV's motion plans. Here, we evaluate the influence of the prediction model used to infer the AV's plans on our method's performance. We consider the following prediction models variants:

1) CV: Constant velocity (CV) model;
2) CVPath: Constant velocity (CV) model along the AV's reference path;
3) MPCC: MPCC plan (Eq. (8)) assuming the AV's current velocity as the velocity reference, $v_{\text{ref}} = v_k$.

Moreover, we also evaluate our method's performance in reactive scenarios employing the IDM [69] to model the other vehicles' behaviors. The presented results in Table VI demonstrate that our proposed approach is robust and generalizes well to environments with other vehicles exhibiting different behaviors. Employing the CV-Path prediction model results in highly cooperative behavior for other vehicles as shown by the high success rate. In contrast, the scenarios

TABLE V: Sensitivity analysis of the hyperparameter K, i.e., number of control cycles per RL policy query, on the learned policy's performance. All policies were trained considering a mixed setting of other vehicles. Querying the RL policy for a new velocity for each two control cycles leads to the best performance (bold values).

	Cooperative			Mixed			Non-cooperative		
	Success(%)	Collision(%)	Timeout(%)	Success(%)	Collision(%)	Timeout(%)	Success(%)	Collision(%)	Timeout(%)
K = 1	80.0	0.0	20.0	70.0	0.0	30.0	33.0	0.0	67.0
K = 2	**88.0**	0.0	**12.0**	**72.0**	0.0	**28.0**	**37.0**	0.0	**63.0**
K = 3	71.5	0.0	28.5	46.75	0.0	53.25	5.5	0.0	94.5
K = 4	72.0	0.0	28.0	47.0	0.0	53.0	0.0	0.0	100.0

TABLE VI: Analysis of the proposed method's performance when interacting with reactive (IDM [69]) and predictive vehicles (CV, CV-Path and MPCC).

	Cooperative			Mixed			Non-cooperative		
	Success(%)	Collision(%)	Timeout(%)	Success(%)	Collision(%)	Timeout(%)	Success(%)	Collision(%)	Timeout(%)
React. Model									
IDM [69]	86.0	0.0	14.0	70.0	1.0	29.0	36.0	0.0	64.0
Pred. Model									
CV	88.0	0.0	12.0	72.0	0.0	28.0	37.0	0.0	63.0
CV-Path	98.0	0.0	2.0	88.0	0.0	12.0	37.0	0.0	63.0
MPCC	89.0	0.0	11.0	76.0	0.0	24.0	39.0	0.0	61.0

TABLE VII: Risk-level analysis: Time of Closest Encounter (TCE) and Distance of Closest Encounter (DCE).

	TCE [s] / DCE [m]		
	Cooperative	Mixed	Non-cooperative
MPCC [29]	9.3 / 3.44	13.6 / 3.13	13.5 / 3.07
RL [68]	26.5 / 3.45	45.6 / 3.15	43.6 / 3.15
IntMPC	25.6 / 3.51	46.0 / 3.20	43.1 / 3.17

with vehicles following an IDM [69] represents the most challenging scenario.

4) Risk-level analysis: Table VII compares the risk-level that the AV takes using our approach against the baseline methods for two risk metrics: Time of Closest Encounter (TCE) and the Distance-of-Closest-Encounter (DCE) [19]. DCE models how close the AV gets to the other vehicles meaning that lower DCE represents higher risk. TCE models the risk time-dependency, assuming that risk events further away in time have a lower probability of occurrence. Hence, the larger TCE, the lower the risk. The presented results show that our method incurs the lowest risk.

I. Discussion

The presented performance and ablation results demonstrate that our approach improves performance and safety significantly compared to pure learning or optimization baselines. Our approach enables the AV to exploit the interaction effects in the other agents to efficiently and safely perform different driving maneuvers by employing RL to learn an interaction-aware velocity reference directly fed into the MPCC's cost function. Nevertheless, the sensitivity analysis results presented in Table VI show some performance degradation when evaluating our approach in scenarios containing agents following different policies from those used in the training scenarios. This effect is due to the *sim-to-real* gap inherent to RL methods [73], and it can be exacerbated when evaluating our approach in real environments.

VII. Conclusion

This paper introduced an interaction-aware policy for guiding a local optimization planner through dense traffic scenarios. We proposed to model the interaction policy as a velocity reference and employed DRL methods to learn a policy maximizing long-term rewards by exploiting the interaction effects. Then, a MPCC is used to generate control commands satisfying collision and kino-dynamic constraints when a feasible solution is found. Learning an interaction-aware velocity reference policy enhances the MPCC planner with interactive behavior necessary to safely and efficiently navigate in dense traffic. The presented results show that our method outperforms solely learning-based and optimization-based planners in terms of collisions, successful maneuvers, and fewer deadlocks in cooperative, mixed, and non-cooperative scenarios.

Future works could replace the simple constant velocity model with an interaction-aware prediction model learned from data, such as [74], [31], expand the interaction-aware policy's network to account for a variable number of other vehicles and control the merging point required for lane-changing in highways. This will improve the prediction performance significantly and so, safety and performance. Finally, future works could implement and evaluate our method in a real autonomous vehicle.

References

[1] M. Bansal, A. Krizhevsky, and A. Ogale, "Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst," *arXiv preprint arXiv:1812.03079*, 2018.

[2] A. Sadat, S. Casas, M. Ren, X. Wu, P. Dhawan, and R. Urtasun, "Perceive, predict, and plan: Safe motion planning through interpretable semantic representations," in *European Conference on Computer Vision*. Springer, 2020, pp. 414–430.

[3] W. Schwarting, J. Alonso-Mora, and D. Rus, "Planning and Decision-Making for Autonomous Vehicles," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, no. 1, pp. 060 117–105 157, 2018. [Online]. Available: http://www.annualreviews.org/doi/10.1146/annurev-control-060117-105157

[4] S. Ulbrich, S. Grossjohann, C. Appelt, K. Homeier, J. Rieken, and M. Maurer, "Structuring cooperative behavior planning implementations for automated driving," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*. IEEE, 2015, pp. 2159–2165.

[5] A. Rudenko, L. Palmieri, M. Herman, K. M. Kitani, D. M. Gavrila, and K. O. Arras, "Human motion trajectory prediction: A survey," *The International Journal of Robotics Research*, vol. 39, no. 8, pp. 895–935, 2020.

[6] 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*, 2020.

[7] W. Schwarting, A. Pierson, J. Alonso-Mora, S. Karaman, and D. Rus, "Social behavior for autonomous vehicles," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 116, no. 50, pp. 2492–24 978, 2019.

[8] D. González, J. Pérez, V. Milanés, and F. Nashashibi, "A Review of Motion Planning Techniques for Automated Vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1135–1145, 2016.

[9] E. Schmerling, K. Leung, W. Vollprecht, and M. Pavone, "Multimodal Probabilistic Model-Based Planning for Human-Robot Interaction," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 3399–3406, 2018.

[10] S. Le Cleac'h, M. Schwager, and Z. Manchester, "Lucidgames: Online unscented inverse dynamic games for adaptive trajectory prediction and planning," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5485–5492, 2021.

[11] P. Trautman, "Sparse interacting Gaussian processes: Efficiency and optimality theorems of autonomous crowd navigation," *2017 IEEE 56th Annual Conference on Decision and Control, CDC 2017*, vol. 2018-Janua, pp. 327–334, 2018.

[12] D. M. Saxena, S. Bae, A. Nakhaei, K. Fujimura, and M. Likhachev, "Driving in dense traffic with model-free reinforcement learning," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 5385–5392.

[13] M. Bouton, J. Karlsson, A. Nakhaei, K. Fujimura, M. J. Kochenderfer, and J. Tumova, "Reinforcement learning with probabilistic guarantees for autonomous driving," 2019.

[14] A. Zgonnikov, D. Abbink, and G. Markkula, "Should i stay or should i go? evidence accumulation drives decision making in human drivers," 2020.

[15] T. Toledo, H. N. Koutsopoulos, and M. E. Ben-Akiva, "Modeling integrated lane-changing behavior," *Transportation Research Record*, vol. 1857, no. 1, pp. 30–38, 2003.

[16] F. Marczak, W. Daamen, and C. Buisson, "Key variables of merging behaviour: empirical comparison between two sites and assessment of gap acceptance theory," *Procedia-Social and Behavioral Sciences*, vol. 80, pp. 678–697, 2013.

[17] G. A. Davis and T. Swenson, "Field study of gap acceptance by left-turning drivers," *Transportation Research Record*, vol. 1899, no. 1, pp. 71–75, 2004.

[18] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Transactions on intelligent vehicles*, vol. 1, no. 1, pp. 33–55, 2016.

[19] J. Eggert, "Predictive risk estimation for intelligent adas functions," in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2014, pp. 711–718.

[20] F. Damerow and J. Eggert, "Predictive risk maps," in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2014, pp. 703–710.

[21] C. R. Baker and J. M. Dolan, "Traffic interaction in the urban challenge: Putting boss on its best behavior," *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, pp. 1752–1758, 2008.

[22] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. Clark, J. Dolan, D. Duggins, T. Galatali, and C. G. e. Al, "Autonomous Driving in Urban Environments: Boss and the Urban Challenge," *Journal of Field Robotics*, vol. 25, no. 8, pp. 425–466, 2008. [Online]. Available: http://onlinelibrary.wiley.com/doi/10.1002/rob.21514/abstract

[23] M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, D. Haehnel, T. Hilden, G. Hoffmann, and B. H. e. Al, "Junior: The stanford entry in the urban challenge," *Journal of Field Robotics*, vol. 25, no. 1, pp. 569–597, 2008. [Online]. Available: http://onlinelibrary.wiley.com/doi/10.1002/rob.21514/abstract

[24] H. Bai, S. Cai, N. Ye, D. Hsu, and W. S. Lee, "Intention-aware online pomdp planning for autonomous driving in a crowd," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 454–460.

[25] W. Liu, S. Kim, S. Pendleton, and M. H. Ang, "Situation-aware decision making for autonomous driving on urban road using online POMDP," in *2015 IEEE Intelligent Vehicles Symposium (IV)*, 6 2015, pp. 1126–1133.

[26] C. Hubmann, M. Becker, D. Althoff, D. Lenz, and C. Stiller, "Decision making for autonomous driving considering interaction and uncertain prediction of surrounding vehicles," *IEEE Intelligent Vehicles Symposium, Proceedings*, no. June, pp. 1671–1678, 2017.

[27] C. Hubmann, J. Schulz, G. Xu, D. Althoff, and C. Stiller, "A Belief State Planner for Interactive Merge Maneuvers in Congested Traffic," *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, vol. 2018-Novem, no. November 2018, pp. 1617–1624, 2018.

[28] B. Zhou, W. Schwarting, D. Rus, and J. Alonso-Mora, "Joint multi-policy behavior estimation and receding-horizon trajectory planning for automated urban driving," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 2388–2394.

[29] L. Ferranti, B. Brito, E. Pool, Y. Zheng, R. M. Ensing, R. Happee, B. Shyrokau, J. F. Kooij, J. Alonso-Mora, and D. M. Gavrila, "Safevru: A research platform for the interaction of self-driving vehicles with vulnerable road users," in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 1660–1666.

[30] J. S. Park, C. Park, and D. Manocha, "I-planner: Intention-aware motion planning using learning-based human motion prediction," *The International Journal of Robotics Research*, vol. 38, no. 1, pp. 23–39, 2019.

[31] B. Brito, H. Zhu, W. Pan, and J. Alonso-Mora, "Social-vrnn: One-shot multi-modal trajectory prediction for interacting pedestrians," *Conference on Robot Learning*, 2020.

[32] S. Bae, D. Saxena, A. Nakhaei, C. Choi, K. Fujimura, and S. Moura, "Cooperation-aware lane change maneuver in dense traffic based on model predictive control with recurrent neural network," in *2020 American Control Conference (ACC)*. IEEE, 2020, pp. 1209–1216.

[33] B. Ivanovic, A. Elhafsi, G. Rosman, A. Gaidon, and M. Pavone, "Mats: An interpretable trajectory forecasting representation for planning and control," *Conference on Robot Learning*, 2020.

[34] 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.

[35] D. Sadigh, S. Sastry, S. A. Seshia, and A. D. Dragan, "Planning for autonomous cars that leverage effects on human actions," *Robotics: Science and Systems*, vol. 12, 2016.

[36] C. You, J. Lu, D. Filev, and P. Tsiotras, "Advanced planning for autonomous vehicles using reinforcement learning and deep inverse reinforcement learning," *Robotics and Autonomous Systems*, vol. 114, pp. 1–18, 2019. [Online]. Available: https://doi.org/10.1016/j.robot.2019.01.003

[37] J. F. Fisac, E. Bronstein, E. Stefansson, D. Sadigh, S. S. Sastry, and A. D. Dragan, "Hierarchical game-theoretic planning for autonomous vehicles," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 9590–9596.

[38] C. F. Camerer, T. H. Ho, and J. K. Chong, "A cognitive hierarchy model of games," *Quarterly Journal of Economics*, vol. 119, no. 3, pp. 861–898, 2004.

[39] M. Garzón and A. Spalanzani, "Game theoretic decision making for autonomous vehicles' merge manoeuvre in high traffic scenarios," *2019 IEEE Intelligent Transportation Systems Conference, ITSC 2019*, pp. 3448–3453, 2019.

[40] M. Bouton, A. Nakhaei, D. Isele, K. Fujimura, and M. J. Kochenderfer, "Reinforcement learning with iterative reasoning for merging in dense traffic," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2020, pp. 1–6.

[41] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang *et al.*, "End to end learning for self-driving cars," *arXiv preprint arXiv:1604.07316*, 2016.

[42] A. Kuefler, J. Morton, T. Wheeler, and M. Kochenderfer, "Imitating driver behavior with generative adversarial networks," in *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2017, pp. 204–211.

[43] F. Codevilla, M. Müller, A. López, V. Koltun, and A. Dosovitskiy, "End-to-end driving via conditional imitation learning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 4693–4700.

[44] A. Amini, W. Schwarting, G. Rosman, B. Araki, S. Karaman, and D. Rus, "Variational autoencoder for end-to-end control of autonomous driving with novelty detection and training de-biasing," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 568–575.

[45] C. Wu, A. Kreidieh, K. Parvate, E. Vinitsky, and A. Bayen, "Flow: Architecture and benchmarking for reinforcement learning in traffic control," *ArXiv*, vol. abs/1710.05465, 2017.

[46] M. Bouton, A. Nakhaei, K. Fujimura, and M. J. Kochenderfer, "Cooperation-aware reinforcement learning for merging in dense traffic," *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 3441–3447, 2019.

[47] P. Wang, C.-y. Chan, and A. D. L. Fortelle, "A Reinforcement Learning Based Approach for Automated Lane Change Maneuvers," no. Iv, pp. 1379–1384, 2018.

[48] M. Mukadam, A. Cosgun, A. Nakhaei, and K. Fujimura, "Tactical decision making for lane changing with deep reinforcement learning," 2017.

[49] T. Tram, A. Jansson, R. Grönberg, M. Ali, and J. Sjöberg, "Learning negotiating behavior between cars in intersections using deep q-learning," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 3169–3174.

[50] L. Hewing, K. P. Wabersich, M. Menner, and M. N. Zeilinger, "Learning-based model predictive control: Toward safe learning in control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, pp. 269–296, 2020.

[51] N. Fulton and A. Platzer, "Safe reinforcement learning via formal methods: Toward safe control through proof and learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[52] J. F. Fisac, N. F. Lugovoy, V. Rubies-Royo, S. Ghosh, and C. J. Tomlin, "Bridging hamilton-jacobi safety analysis and reinforcement learning," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8550–8556.

[53] R. Cheng, M. J. Khojasteh, A. D. Ames, and J. W. Burdick, "Safe Multi-Agent Interaction through Robust Control Barrier Functions with Learned Uncertainties," in *2020 59th IEEE Conference on Decision and Control (CDC)*, vol. 2020-Decem, no. Cdc. IEEE, dec 2020, pp. 777–783. [Online]. Available: https://ieeexplore.ieee.org/document/9304395/

[54] T. Tram, I. Batkovic, M. Ali, and J. Sjöberg, "Learning when to drive in intersections by combining reinforcement learning and model predictive control," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 3263–3268.

[55] B. Brito, M. Everett, J. P. How, and J. Alonso-Mora, "Where to go next: Learning a subgoal recommendation policy for navigation in dynamic environments," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4616–4623, 2021.

[56] S. Levine and V. Koltun, "Guided policy search," in *International Conference on Machine Learning*, 2013, pp. 1–9.

[57] ——, "Variational policy search via trajectory optimization," in *Advances in neural information processing systems*, 2013.

[58] I. Mordatch and E. Todorov, "Combining the benefits of function approximation and trajectory optimization." in *Robotics: Science and Systems*, vol. 4, 2014.

[59] Z.-W. Hong, J. Pajarinen, and J. Peters, "Model-based lookahead reinforcement learning," 2019.

[60] J. Lubars, H. Gupta, A. Raja, R. Srikant, L. Li, and X. Wu, "Combining reinforcement learning with model predictive control for on-ramp merging," *CoRR*, vol. abs/2011.08484, 2020. [Online]. Available: https://arxiv.org/abs/2011.08484

[61] T. Wang and J. Ba, "Exploring model-based planning with policy networks," in *International Conference on Learning Representations*.

[62] C. Greatwood and A. G. Richards, "Reinforcement learning and model predictive control for robust embedded quadrotor guidance and control," *Autonomous Robots*, vol. 43, no. 7, pp. 1681–1693, 2019.

[63] Y. Yu, "Towards sample efficient reinforcement learning." in *IJCAI*, 2018, pp. 5739–5743.

[64] W. Zhao, J. P. Queralta, and T. Westerlund, "Sim-to-real transfer in deep reinforcement learning for robotics: a survey," in *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2020, pp. 737–744.

[65] W. Schwarting, J. Alonso-Mora, L. Paull, S. Karaman, and D. Rus, "Safe Nonlinear Trajectory Generation for Parallel Autonomy with a Dynamic Vehicle Model," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 9, pp. 2994–3008, 2018.

[66] B. Berg, B. Brito, J. Alonso-Mora, and M. Alirezaei, "Curvature Aware Motion Planning with Closed-Loop Rapidly-exploring Random Trees," in *2021 IEEE Intelligent Vehicles Symposium (IV)*, 2021.

[67] B. Brito, B. Floor, L. Ferranti, and J. Alonso-Mora, "Model Predictive Contouring Control for Collision Avoidance in Unstructured Dynamic Environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4459–4466, 2019.

[68] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *ICML*, 2018.

[69] 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.

[70] A. Zanelli, A. Domahidi, J. Jerez, and M. Morari, "Forces nlp: an efficient implementation of interior-point methods for multistage nonlinear nonconvex programs," *International Journal of Control*, vol. 93, no. 1, pp. 13–29, 2020.

[71] E. Leurent, "An Environment for Autonomous Driving Decision-Making," 2018. [Online]. Available: https://github.com/eleurent/highway-env

[72] D. Ni, J. D. Leonard, C. Jia, and J. Wang, "Vehicle longitudinal control and traffic stream modeling," *Transportation Science*, vol. 50, no. 3, pp. 1016–1031, 2016.

[73] J. Matas, S. James, and A. J. Davison, "Sim-to-real reinforcement learning for deformable object manipulation," in *Conference on Robot Learning*. PMLR, 2018, pp. 734–743.

[74] X. Ma, J. Li, M. J. Kochenderfer, D. Isele, and K. Fujimura, "Reinforcement learning for autonomous driving with latent state inference and spatial-temporal relationships," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 6064–6071.

Bruno Brito Bruno Brito is a Ph.D. student at the Department of Cognitive Robotics at the Delft University of Technology. He received the M.Sc. (2013) degree from the Faculty of Engineering of the University of Porto. Between 2014 and 2016, he was a trainee at the European Space Agency (ESA) in the Guidance, Navigation and Control section. After, he was a Research Associate, between 2016 and 2018, in the Fraunhofer Institute for Manufacturing Engineering and Automation. Currently, his research is focused on developing motion planning algorithms bridging learning-based and optimization-based methods for autonomous navigation among humans.

Achin Argawal Achin Agarwal received the M.Sc. degree in Mechanical Engineering from the Delft University of Technology in December 2020, with a Master Thesis focused on modeling highly interactive behavior between various traffic entities and navigation in dense traffic. Currently, he is exploring the application of Artificial Intelligence techniques for modeling the influence of various parameters involved in financial markets.

Javier Alonso-Mora Javier Alonso-Mora is an Associate Professor at the Delft University of Technology and head of the Autonomous Multi-robots Laboratory. Dr. Alonso-Mora was a Postdoctoral Associate at MIT and received his Ph.D. degree from ETH Zurich, where he worked in a partnership with Disney Research Zurich. His main research interest is in navigation, motion planning and control of autonomous mobile robots, with a special emphasis on multi-robot systems, on-demand transportation and robots that interact with other robots and humans in dynamic and uncertain environments. He is the recipient of multiple prizes and grants, including an ERC Starting grant (2021), the ICRA Best Paper Award on Multi-robot Systems (2019), an Amazon Research Award (2019) and a talent scheme VENI award from the Netherlands Organisation for Scientific Research (2017).