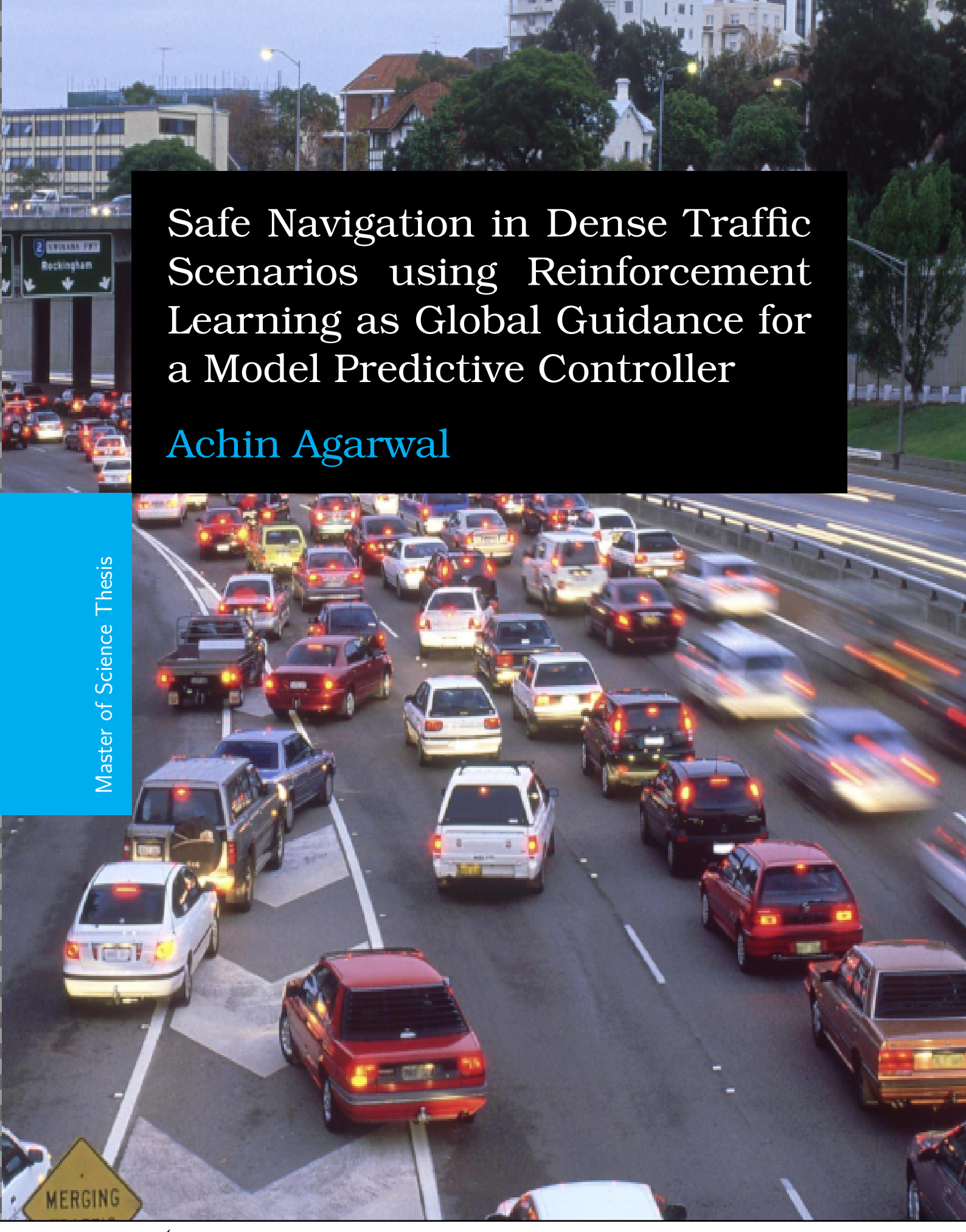


Safe Navigation in Dense Traffic Scenarios using Reinforcement Learning as Global Guidance for a Model Predictive Controller

Achin Agarwal

Master of Science Thesis

TU Delft Delft University of Technology

Department of Cognitive Robotics

Safe Navigation in Dense Traffic Scenarios using Reinforcement Learning as Global Guidance for a Model Predictive Controller

by

Achin Agarwal

For the degree of Master of Science in Vehicle Engineering at
Delft University of Technology

Student number: 4727592
Project duration: October, 2019 – December, 2020
Thesis committee:

Dr. Javier Alonso-Mora,	TU Delft, supervisor
Dr. Wei Pan,	TU Delft
Bruno Brito,	TU Delft, supervisor

Department of Cognitive Robotics (CoR)
Faculty of Mechanical, Maritime and Materials Engineering (3mE)

TU Delft

Acknowledgements

I would like to thank my supervisor Dr Javier Alonso-Mora for his invaluable insights and meticulous analysis of my work. His kind nature coupled with his excellent analytical skills helped me immensely in honing my skills as a researcher. I would also like to thank my PhD supervisor Bruno Brito for his indispensable feedback and guidance throughout the project. The conversations and weekly meetings were crucial in inspiring me to think outside the box and provided me with multiple perspectives that proved quite resourceful in the completion of the project.

I would also like to thank my parents for their unconditional support throughout the project. Finally, I would like to express my gratitude and appreciation for my friends (Ewoud, Yannick, Carla, Vishant, Vishruth, Anoosh, Sneha) for the intellectual conversations and the fun weekends.

Delft, University of Technology
November 30, 2020

Achin Agarwal

Safe Navigation in Dense Traffic Scenarios using Reinforcement Learning as Global Guidance for a Model Predictive Controller

Achin Agarwal

***Abstract*—The successful integration of autonomous vehicles (AVs) in human environments is highly dependent on their ability to navigate safely and timely through dense traffic conditions. Such conditions involve a diverse range of human behaviors, ranging from cooperative (willing to yield) to non-cooperative human drivers (unwilling to yield) that need to be identified without any explicit inter-vehicle communication. In order to maneuver through such conditions, AVs must not only compute a collision-free trajectory but also account for the effects of its actions on the surrounding agents to negotiate the navigation maneuver safely. Existing motion planning techniques fail in these environments because they suffer from one or more of the following drawbacks: suffer from "the curse of dimensionality" due to the high number of agents (e.g., optimization-based methods); do not account for the interaction effects among the agents; do not provide any collision avoidance or trajectory feasibility guarantees (e.g., learning-based methods). In this paper, we propose a novel navigation framework combining the strengths of learning-based with optimization-based algorithms. More specifically, we employ a Soft Actor-Critic agent to learn a continuous guidance policy that provides global guidance to an optimization-based planner generating feasible and collision-free trajectories. We evaluate our method in a highly interactive simulation environment where we compare our method with two baseline approaches, a learning-based method and an optimization-based method, and present performance results demonstrating our method significantly reduces the number of collisions and increase the success rate with fewer number of deadlocks. We also show that that our method is able to generalise and applicable to other traffic scenarios (e.g., an unprotected left turn).**

***Index Terms*—Safe Navigation, Motion Planning, Deep Reinforcement Learning, Optimal Control**

I. Introduction

Robust driving in real-world dense traffic scenarios requires interacting with human drivers, which still stands as a hurdle in the widespread deployment of autonomous vehicles [1]. To successfully integrate autonomous vehicles (AVs) into our society, AVs need to deal with cluttered environments such as highway merging and unprotected left turns where navigation is not possible without interacting with other traffic participants. These type of cluttered scenarios involve intricate observations and interactions that even human drivers find challenging.

Driving in dense traffic conditions is intrinsically an interactive task, where the actions executed by the AVs 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 leverage the cooperativeness of other vehicles to make them yield, creating room in the process for it to merge safely.

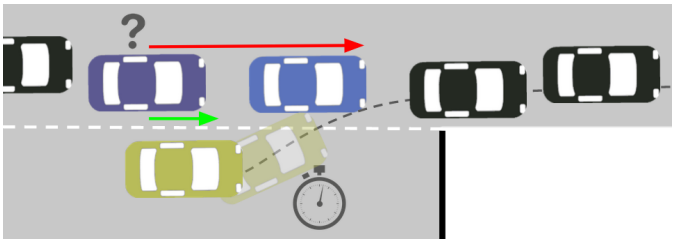


Fig. 1. Illustration of a dense on-ramp merging traffic scenario where the autonomous vehicle (yellow) needs to interact with other traffic participants in order to merge onto the main highway in a timely and safe manner. The potential follower (purple) may yield (green arrow) to the autonomous vehicle 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.

Despite all the recent advancements in learning and optimization methods, mobile computational power that enable all the current autonomous driving solutions (e.g., Waymo [2], Uber [3]), current motion planning solutions still fail to scale in cluttered environments. The majority of the 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 the nearby traffic [1], [4]. Nevertheless, works that account for the interaction among the agents either do not scale for a large number of agents due to the curse of dimensionality [5] or cannot be in real-time [6] or do not allow to define collision constraints explicitly [7]. Moreover, most of these methods rely on finding obstacle-free space, which is hard to find in these dense traffic scenarios [4]. Hence, motion planning methods that can safely interact with the other navigating agents while generating kino-dynamically feasible trajectories are necessary.

In this paper, we propose a method for autonomous navigation in dense traffic scenarios. Our method leverages the interaction effects among the vehicles to create free-space areas for the ego-vehicle to navigate and allowing it to successfully complete various driving maneuvers in cluttered environments. More specifically, we propose to use Model-free Deep Reinforcement Learning (MDRL) to learn a continuous policy that provides global guidance to a local optimization-based

planner. By exploiting Deep Reinforcement Learning's ability to learn joint interaction behavior, we learn complex policies from data while the optimization-based planner ensures that the generated trajectories are kino-dynamically feasible and safety constraints are respected.

II. Related Work

The literature devoted to the problem of modelling human interactions among traffic participants is vast. A recent survey by [1] divides these works into four main categories: rule-based, game theoretic, learning-based and optimization-based methods.

Firstly, rule-based methods [8]–[10] have been proposed to tackle the decision-making problems demonstrating excellent ability to solve specific problems (e.g., precedence at an intersection followed by waiting for availability of enough free space for the vehicle to pass safely [8]). Nevertheless, these methods do not consider the interactions between multiple traffic participants and fail in dense traffic scenarios.

Real driving environments are intrinsically partially observable. Hence, to account for un-observable states of the other agents, [11] considers partially observable Markov decision process (POMDPs) into the formulation. Furthermore, [12] proposed to use dimensional reduction techniques creating a compressed and fixed-size representation of the other agents information and incorporate road context [13] to scale for larger number of vehicles. 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. 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 Reinforcement Techniques have been used to learn human-drivers preferences [14], [15]. These methods either fail to scale to interact with multiple agents [14] or can only handle a discrete set of actions [15] rendering them incapable to be used safely in highly interactive and dense traffic scenarios.

Game Theoretic approaches such as [16] propose to model the interaction among the agents as a game allowing to infer the influence on each agent's plans. However, the task of modelling interactions requires inter-dependency of all agents on each other actions, to be embedded within the framework. This results in an exponential growth of interactions as the number of agents increase, rendering the problem computationally intractable. Another example would be [17] where an iterative level-k model based on cognitive hierarchy reasoning [18] has been used to obtain a near optimal policy for performing merge maneuvers in highly dense traffic scenarios. This method shows promise but is limited to discrete action spaces and still needs to be analysed in an interactive simulation exhibiting a diverse range of human driving behaviors.

Learning-based approaches leverage on large data collection to build interaction models. In [5], deep neural networks were used to learn from data the action distributions of a driving agent conditioned on the interaction history. Recently, Reinforcement Learning has shown a potential in modelling interactions in highly dense and uncertain traffic scenarios [19], [7], [20]. These methods are able to learn a working policy under highly interactive traffic conditions involving multiple entities. However, they fail to provide any safety guarantees and reliability, rendering these methods vulnerable to collisions. Moreover, MDRL's policies trained in simulations rarely transfer to real life situations (e.g., due to kino-dynamic constraints associated with vehicles in real life) thus, limiting their practical application.

In contrast, optimal control methods consider the vehicle dynamics model to generate kino-dynamically feasible trajectories, allow to follow a pre-defined path while avoiding static and dynamic obstacles [1], [21]. However, they fail to account for interaction and struggle to find a collision free trajectory in highly dense traffic scenarios. Efforts have been made to embody interaction within the optimal controller framework [22], [23] but they either assume complete control over other agent's actions [22] or can only handle interaction with a single agent [23]. Moreover, these methods also suffer from the curse of dimensionality and their performance in a highly interactive simulation still needs to be investigated.

A. Contribution

The main contributions of this work are:

- A novel navigation framework for interaction-aware and safe navigation in cluttered environments that involves providing global guidance to a local optimization-based controller by an interaction-aware MDRL agent.
- Extensive simulation results demonstrating the ability of our approach to negotiate with cooperative and non-cooperative vehicles in a highly interactive simulation environment capable of simulating complex negotiating behavior that considers future state of the ego vehicle to achieve non-reactive interaction-aware behavior for the agents.

III. Problem Formulation

Let us consider a set of vehicles interacting in a dense traffic scenario comprising of an autonomous vehicle (henceforth referred to as the ego vehicle) and n human drivers (henceforth referred to as agents) exhibiting different levels of willingness to yield. The term "vehicles" is used to collectively refer to the ego vehicle and agents. The state $\mathbf{z}^0$ of the ego vehicle and the agent $i \in \{1, \dots, n\}$ is defined by

$$\mathbf{z}^{0,i} = \{x, y, \psi, v\} \forall i \in \{1, \dots, n\}$$

where $\boldsymbol{p} = [x, y]$ is the position, ψ the heading angle and v the forward velocity in a global inertial frame $\mathcal{W}$. We define the joint state as the set of all agent states $\mathbf{Z} = \{\mathbf{z}^0, \dots, \mathbf{z}^n\}$. The area occupied by the ego vehicle is given by A^{ego} whereas for the agents, it is represented by A^{obs}. The ego vehicle needs to be cognizant of the effects of its own actions on nearby vehicles (referred to as "interaction") which is required to

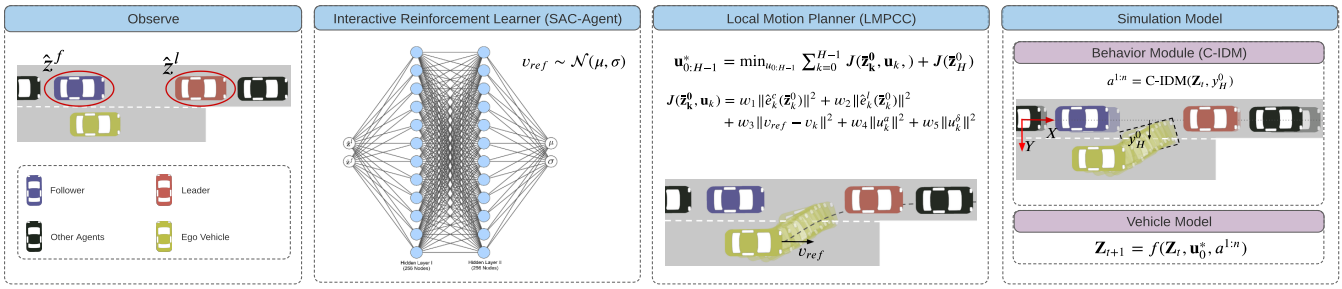


Fig. 2. Our proposed architecture comprises of three main modules: Interactive Reinforcement Learner (SAC-Agent), Local Motion Planner (LMPCC) and Simulation Model. The ego vehicle observes the leader state $\hat{z}^l$ and follower state $\hat{z}^f$ relative to it which serves as input to the Interactive Reinforcement Learner. The SAC-Agent samples a reference velocity $v_{\text{ref}} \sim \mathcal{N}(\mu, \sigma)$ (μ-mean, σ-standard deviation) for LMPCC to follow. The Local Motion Planner LMPCC then computes locally optimal sequence of commands $\mathbf{u}^*_{0:H-1}$ while minimizing a cost function $J(\bar{\mathbf{z}}^{\mathbf{0}}_{\mathbf{k}}, \mathbf{u}_k)$. $\mathbf{w} = [w_1, \ldots, w_5]$ denotes a set of cost weights. The cost function includes penalties for contour error ($\hat{e}^c_k$), lag error ($\hat{e}^l_k$) and control commands u^a_k and u^δ_k. The term $\|v_{\text{ref}} - v_k\|$ motivates the planner to follow v_{ref} closely provided by the SAC-Agent. C-IDM then computes acceleration values $a^{1:n}$ for the agents while being aware of ego vehicle's future plan y^0_H. The joint state for the next time step $\mathbf{Z}_{t+1}$ is then computed using the Vehicle model.

maneuver through a cluttered environment. The objective is to learn a continuous policy π controlling the interaction with the other agents and minimizing the expected driving time $\mathbb{E}[t_g]$ for the ego vehicle to reach its goal position. We formulate our objective as the following optimization problem:

$$\underset{\pi(\mathbf{Z})}{\text{argmin}} \quad \mathbb{E}\left[t_g \mid \mathbf{Z}_0, \pi\right] \tag{1a}$$

$$\text{s.t.} \quad \mathbf{Z}^0_{k+1} = f(\mathbf{Z}^0_k, \mathbf{u}_k), \tag{1b}$$

$$A^{ego}_k \cap A^{obs}_k = \emptyset \tag{1c}$$

where $f(\mathbf{Z}^0_k, \mathbf{u}_k)$ is the state transition function[1] for the ego vehicle and $\mathbf{u}_k$ the control command. Moreover, the policy must respect the ego-vehicle kino-dynamics (Equation 1(b)) and ensure collision-free motion (Equation 1(c)). In this paper, we assume that $\mathbf{Z}$ is accessible (within a limited perception range) to the ego vehicle using sensor fusion of various on board sensors (e.g., cameras, Lidar, Radar, IMUs, GPS) capable of processing incoming information in real time. Furthermore, it is assumed that there is no sensor noise and inter-vehicle communication. Moreover, to consider a more realistic scenario, we consider a mixed-agents setting with a continuously varying cooperation level $c_i \in [w_{\min}, w_{\max}] \forall i \in \{1, n\}$ (detailed in Sec. IV-D1) ranging from $w_{\min}$, a non-cooperative agent, to $w_{\max}$, a fully-cooperative agent.

IV. Interactive Policy Construction and Planning

A. Overview

In this section, we introduce our motion planning framework for safe navigation in dense and interactive scenarios. Figure 2 depicts our proposed motion planning architecture. The proposed approach comprises of three main modules: An Interactive Reinforcement Learner (SAC-Agent), a local optimization planner (LMPCC) and a simulation model. Firstly, we present a training algorithm employing an MDRL agent (Section IV-B) combined with a local optimization planner (Section IV-C) to jointly learn an interaction-aware safe navigation policy. Then, to model interaction, we propose an expansion for the IDM model allowing the agents to react on the other's predicted plans (SectionIV-D).

[1]This is identical to the Vehicle Model used in the simulation.

B. Interactive Reinforcement Learner

In this paper, we propose to use MDRL to learn a global guidance policy taking advantage of the interactions among the agents to enable navigation in dense environments, and thus, solving the *Freezing Robot Problem*.

The task of autonomous driving is inherently a continuous control task. Hence, in this work we consider a continuous control policy allowing fine control for the ego-vehicle. Moreover, we consider a stochastic policy and aim to learn probability distribution $\pi(\hat{z}^l_t, \hat{z}^f_t) \rightarrow v_{\text{ref}}$ conditioned on leader state $\hat{z}^l_t$ and follower state $\hat{z}^f_t$ relative to the ego vehicle. To train our policy, we use the Soft Actor Critic Agent (SAC) [24] method. SAC agent augments the objective of traditional RL algorithms with the entropy of the policy. This embeds the notion of exploration into the policy while giving up on clearly unpromising paths [24]. Yet, note that our approach is agnostic to which learning algorithm we use.

1) Observation Space: For the ego vehicle, a leader (agent in front) and a follower (agent behind the ego vehicle) are computed at every timestep. We define the observation vector as the joint state for the leader and the follower relative to ego vehicle's state ($\mathbf{z}^0$). The state for the leader relative to $\mathbf{z}^0$ is denoted by $\hat{\mathbf{z}}^l$ whereas the state of the follower relative to $\mathbf{z}^0$ is represented by $\hat{\mathbf{z}}^f$.

This observation space is used as an input to the neural network parameterizing our policy. The ego vehicle is assumed to have a limited perception range meaning it only has access to states of the agents within the perception range.

2) Action Space: We propose to model the guidance information to the ego-vehicle as a velocity reference $v_{ref} \sim \mathcal{N}(\mu, \sigma)$, where μ is the mean and σ is the variance. By providing a velocity reference to a local motion planner, the Interactive Learner can directly control the interaction at the

merging point with the other agents by being more aggressive (high speed values) or more conservative (low speed values).

3) Reward Function: We formulate a reward function to motivate progress along a reference path, to penalise 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:

$$R_k\left(\mathbf{Z},\mathbf{a}\right)=\begin{cases} v_t^0 & \\ r_{\text{infeasible}} & \text{if } c_k^{\text{obst},i}>1\ \forall i\in\{1,n\} \\ r_{\text{collision}} & \text{if } A_k^{ego}\cap A_k^{obs}\neq\emptyset \\ r_{\text{near}} & d_{\text{min}}(\mathbf{x}^0,\mathbf{x}^i)\leq\Delta d_{\text{min}}\ \forall i\in\{1,n\} \end{cases} \tag{2}$$

where $c_k^{\text{obst},i}$ is the collision avoidance constraint between the ego vehicle and the agent i as described in eq. (5), $A_k^{ego}\cap A_k^{obs}$ represents the common area occupied by the ego vehicle and the agents at step k. d_{min} is the minimum distance to the closest agent i and Δd_{min} is a hyper-parameter distance threshold. The first term v_t^0 is a reward proportional to the ego-vehicle's velocity encouraging higher velocities and thus, minimizing the time to goal. The second $r_{\text{infeasible}}$, third $r_{\text{collision}}$ and fourth term r_{near} penalize the ego-vehicle for infeasible solutions, collisions and for driving too close to other agents, respectively.

C. Local Motion Planner

An MDRL agent can be directly used to learn a control policy in dense traffic scenarios [7], [19] however, there is no guarantee that the computed plan would be safe and kino-dynamic constraints would be satisfied. MPC, on the other hand, can help provide these assurances. In this paper, we use an MPC to generate locally optimal trajectories satisfying kino-dynamics and collision avoidance constraints. Additionally, MPC follows a global reference path while respecting the road boundaries.

In our method we used the LMPCC inspired by [25]. To make the MPC formulation more legible, we augment the state $\mathbf{z}^0$ of the ego vehicle with the progress variable θ that represents the progress along the reference path. The new state for the ego vehicle in the MPC formulation is now denoted by $\bar{\mathbf{z}}^0$.

1) Cost Function: The LMPCC receives a velocity reference v_{ref}, from the Interactive Learner (SectionIV-B), accounting for the interaction effects of the ego-vehicle with the other agents. We design the stage cost motivating the ego-vehicle to follow a pre-defined path and tracking the velocity reference, defined as follows:

$$\begin{aligned} J(\bar{\mathbf{z}}_{\mathbf{k}}^{\mathbf{0}},\mathbf{u}_k)=w_1\|\hat{e}_k^c(\bar{\mathbf{z}}_k^0)\|^2+w_2\|\hat{e}_k^l(\bar{\mathbf{z}}_k^0)\|^2 \\ +w_3\|v_{ref}-v_k\|^2+w_4\|u_k^a\|^2+w_5\|u_k^\delta\|^2 \end{aligned} \tag{3}$$

where $\mathbf{w}=[w_1,\ldots,w_5]$ denotes a set of cost weights.To track the reference path closely, we minimize two cost terms: the contour error ($\hat{e}_k^c$) and lag error ($\hat{e}_k^l$). 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 [25]. The third term, $\|v_{\text{ref}}-v_k\|$, motivates the planner to follow v_{ref} closely. Finally, to generate smooth trajectories, we add a quadratic penalty to the control commands u_k^a and u_k^δ.

2) Dynamic Obstacle Avoidance: The occupied area for the ego vehicle is represented by $\bar{A}^{\text{ego}}(\boldsymbol{p})$, which is approximated by a union of n_c circles i.e $\bar{A}^{\text{ego}}(\boldsymbol{p})\subseteq\bigcup_{c\in\{1,\ldots,n_c\}}\mathcal{A}_c(\boldsymbol{p})$, where $\mathcal{A}_c$ is the area occupied for a circle with radius r. For every agent i, the occupied area is represented by $\mathcal{A}_i$ which is approximated by an ellipse of semi-major axis a_i and semi-minor axis b_i and orientation ψ. At any instant, the area occupied by all the agents is given by $\bar{A}^{\text{obs}}=\bigcup_{i\in\{1,\ldots,n\}}\mathcal{A}_i$. To ensure collision-free motion over the planning horizon, the following condition must be satisfied:

$$\bar{A}^{\text{ego}}(\boldsymbol{p}_k)\cap\bar{A}_k^{\text{obs}}=\emptyset\ \forall k\in\{0,\ldots,N-1\} \tag{4}$$

For every agent $i\in\{1,\ldots,n\}$ and prediction step k, we define a non-linear constraint imposing that each circle j of the ego vehicle with the elliptical area occupied by the agent do not intersect:

$$c_k^{\text{obst},j}(\mathbf{Z}_k)=\begin{bmatrix}\Delta x_k^j\\ \Delta y_k^j\end{bmatrix}^{\mathrm{T}}R(\psi)^{\mathrm{T}}\begin{bmatrix}\frac{1}{\alpha^2} & 0\\ 0 & \frac{1}{\beta^2}\end{bmatrix}R(\psi)\begin{bmatrix}\Delta x_k^j\\ \Delta y_k^j\end{bmatrix}>1, \tag{5}$$

The parameters Δx_k^j and Δy_k^j represent x-y relative distances in ego-agent's frame between the disc j and the agent for prediction step k. To guarantee collision avoidance we enlarge the other agent 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 [21].

3) Road boundaries: To ensure that the vehicle stays within the road boundaries, we introduce constraints on the lateral distance $d(\bar{\mathbf{z}}_k^0)$, computed as the contour error of the ego-vehicle with respect to the reference path is bounded [11].

$$-w_{maxim}\leq d(\bar{\mathbf{z}}_k^0)\leq w_{maxim} \tag{6}$$

where w_{maxim} is the maximum value of the vehicle's boundary projected in the norm direction of the reference path.

4) MPC Formulation: We formulate the motion planning problem as a Receding Horizon Trajectory Optimization problem (7) with planning horizon H conditioned on the following constraints:

$$\mathbf{u}_{0:H-1}^*=\min_{u_{0:H-1}}\sum_{k=0}^{H-1}J(\bar{\mathbf{z}}_{\mathbf{k}}^{\mathbf{0}},\mathbf{u}_k)+J(\bar{\mathbf{z}}_H^0) \tag{7a}$$

$$\text{s.t.}\quad \mathbf{z}_{k+1}^0=f(\mathbf{z}_k^0,\mathbf{u}_k), \tag{7b}$$

$$-w_{maxim}\leq d(\bar{\mathbf{z}}_k^0)\leq w_{maxim} \tag{7c}$$

$$c_k^{\text{obst},j}(\mathbf{Z}_k)>1\ \ \forall j\in\{1,\ldots,n_c\}\ \ \forall\,\text{obst} \tag{7d}$$

D. Simulation Model

We build our simulation environment[2] on [26] expanding it to incorporate complex interaction behavior.

[2]https://github.com/eleurent/highway-env

Algorithm 1: Training Procedure of SAC + LMPCC

// Collect batches from multiple instances of the same environment running in parallel
Initialize Global path for ego vehicle;
while *episodes* $<$ *max_episodes* **do**
  // For every episode:
  Initialize states - $\mathbf{Z}_0$*;*
  while *not terminal* **do**
    // Sample v_{ref} *from the policy given the current joint state* $\mathbf{Z}_t$ (Section IV-B)
    $v_{ref} \sim \pi(\mathbf{Z}_t)$;
    // Using the same v_{ref}*, compute actions and transition the states for 2 consecutive time-steps*
    K = 0;
    while $K \leq 2$ **do**
      // Solve the trajectory optimization problem to compute optimal actions for the ego vehicle with obstacle avoidance constraints disabled (Section IV-C)
      $\mathbf{u}_0^*, y_H^0$ = *LMPCC*$(v_{ref}, \mathbf{Z}_{t+K})$*;*
      // Compute longitudinal accelerations for agents using C-IDM (Section IV-D1)
      $a^{1:n}$ = *C-IDM*$(\mathbf{Z}_{t+K}, y_H^0)$*;*
      // Step the states using vehicle model (Section IV-D2) by executing $\mathbf{u}_0^*, a^{1:n}$ *and get the new joint state* $\mathbf{Z}_{t+K+1}$
      $\mathbf{Z}_{t+K+1} = f(\mathbf{Z}_{t+K}, \mathbf{u}_0^*, a^{1:n})$*;*
      $K = K + 1$*;*
      Check terminal;
    end
  end
  // Concatenate samples from all the instances of the environment
  // Run SAC algorithm
end

1) Behavior Module (C-IDM): The main objective of the Behavior Module is to be able to simulate dense and complex negotiating behavior with varying degrees of willingness to yield. In a typical dense traffic scenario (e.g. on-ramp merging), agents trying to merge onto the main lane needs to leverage the cooperativeness of other agents to create obstacle free space to be able to safely merge. On the other hand, agents on the main lane exhibit different levels of willingness to yield with some agents stopping as soon as they spot the agent on the adjacent lane (Cooperative) while other agents ignore the agent entirely and may even accelerate to deter it from merging (Non-Cooperative). Moreover, at the merging point, they also consider the future plan of the agent on the adjacent lane to decide if they should yield or not. Modelling such complex behavior is non-trivial. This section provides insights on the method used to model such complex negotiating behavior exhibited by the agents.

Past works use the popular rule based method Intelligent Driver Model (IDM) to model the other agents driving policy [27]. In IDM, an agent compromises between reaching a pre-defined maximum velocity and maintaining a minimum safe distance to the agent in front (leader). However, this model only considers the current state of the agents in the same lane leading to reactive and incomplete representation of the behavior of the agents typical in dense traffic scenarios such as on-ramp merging and unprotected left turn.

We extend IDM to compute longitudinal accelerations for agents on the current lane while being aware of the future plan of the agents on the adjacent lane[3] The agents on the main lane maintain an internal belief about the other agents plan (on the adjacent lane) which in the case of the ego vehicle, is an approximation of the LMPCC plan. Specifically, this is achieved by introducing a continuous cooperation constant for every agent (c^i) that controls whether an agent cooperates with the ego vehicle or not.

At the beginning of the simulation, the value of c_i is sampled from the uniform distribution defined bounded by $[w_{\min}, w_{\max}]$ where $w_{\max}$ is the distance between the center of the current lane and the adjacent lane which is a constant (4 m) in our case and $w_{\min} \in [0, w_{\max}]$. $w_{\min}$ plays in important role in the behavior of the final policy as it controls the proportion of cooperative and non-cooperative agents encountered by the ego vehicle during training which can either make the final policy too aggressive or conservative. At every time step, the cooperativeness value of every agent is compared with the lateral horizon parameter (y_H^0) (future plan) which is defined as the the lateral position of the H horizon step state output of the solution of Eq. 7 If $c^i > y_H^0$, the ego vehicle is included in the set of potential leaders following which a leader is chosen for the agent's IDM based on the closest longitudinal distance. For the case where the leader comes out to be the ego vehicle on the adjacent lane, the agent moves with the projection of the ego vehicle on the current lane.

One of the main reasons for using y_H^0 instead of the current lateral position y_0^0 is to incorporate the future plan of ego vehicle into the behavior of the agents. This helps in eliciting non-reactive behavior from the agents. Moreover, the notion of ego vehicle's aggressiveness is also inculcated into their behavior as the lateral horizon parameter (y_H^0) is a direct function of the current velocity of the ego vehicle. By using c^i as a parameter to incorporate varying degrees of cooperativeness and y_H^0 to elicit non-reactive behavior from agents, a wide variety of behaviors can be simulated which helps in evaluating the efficacy of the proposed approach.

2) Vehicle Model: We employ a kinematic bicycle model to model the motion of the vehicles, described as follows:

[3]For the Ramp Merging scenario (detailed in Sec. V-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 V-B2), the current lane refers to the top lane and the adjacent lane corresponds to the bottom lane.

$$
\begin{aligned}
\dot{x} &= v\cos(\psi+\beta)\\
\dot{y} &= v\sin(\psi+\beta)\\
\dot{\psi} &= \frac{v}{l_r}\sin(\beta)\\
\dot{v} &= a\\
\beta &= \arctan\left(\frac{l_r}{l_f+l_r}\tan(\delta)\right)
\end{aligned}
\tag{8}
$$

where δ is the front tyre's angle, β is the velocity angle and a the linear acceleration. 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 is the steering angle and acceleration $\mathbf{u} = [a, \delta]$.

E. Training Procedure

We jointly train the guidance policy with local motion planner allowing the trained policy to learn with the cases that result in an infeasible solution for the MPC solver.

Algorithm 1 presents a step by step overview of our training procedure. We employ a parallel implementation of the SAC method allowing to speed-up the sample collection process. Every episode begins with the initialisation of state $\mathbf{Z_0}$ of all the vehicles. For every K control cycles, we sample a reference velocity v_{ref} from the policy π. Using the same v_{ref} for K control cycles helps the learning algorithm to better assess the impact of its action on the environment. A detailed analysis of the impact of different values of K on the behavior of the final policy can be found in Section VI-F. Using v_{ref}, LMPCC computes locally optimal sequence of commands $u^*_{0:H-1}$ for the ego vehicle to execute out of which only the action for the first time-step is applied. During the training, we disable the obstacle avoidance constraint from the LMPCC formulation to allow the ego-vehicle to be exposed to dangerous situations or to collisions. This helps our policy to learn to closely interact with nearby agents. A detailed analysis of the effects of training with obstacle avoidance constraints enabled and disabled on the final policy can be seen in Appendix B. C-IDM then computes actions $a^{1:n}$ for the agents while being aware of the ego vehicle on the adjacent lane. The states are then advanced using the Vehicle Model.

An episode is executed until a terminal state is reached which can happen due to one of the following reasons: the ego vehicle reaches the goal position; the ego vehicle collides with other agent; An infeasible solution is computed by the solver or a time out. In the context of our implementation, the solver outputs an infeasible solution if the obstacle avoidance constraints or road boundary constraints are violated.

V. Implementation

A. Experimental setup

The training was carried out on an Intel(R) Core(TM) i9-9980HK CPU @ 2.40GHz processor with 16 cores and took approximately 15 hours to train. This corresponds to 12 million training steps of the environment. The hyperparameters for the SAC algorithm can be found in Table I. Our code can be found here[4].

TABLE I
Hyperparameters for SAC

Hyperparameter	Value
Number of workers in parallel	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	2100
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
Actor learning rate	3×10^4
Critic learning rate	3×10^4
Entropy learning rate	3×10^4
Replay buffer size	10^6

B. Driving scenarios

We consider two driving scenarios: merging on a densely populated highway and maneuvering through dense traffic to take an 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 agents is drawn from a uniform distribution bounded by [7, 10] m and kept constant for every agent. This is followed by addition of uniform noise from the interval [-1, 1] m. Their initial and target velocities and sampled from a uniform distribution bounded by [3, 4] m/s. Finally, to simulate more realistic scenarios, we consider agents with different cooperation levels sampled from a uniform distribution $c_i \sim \mathcal{U}(0, 4)$ m. The motivation behind choosing w_{min} as zero can be found in Appendix A. This initial configuration of agents prevents early collisions while ensuring there are no gaps of more than 2 m present that are typical of highly dense traffic scenarios. This compels the ego vehicle to leverage other agents' cooperativeness while also exposing it to a myriad of different scenarios that are critical for the performance of the final policy.

1) Ramp Merging: The merging scenario can be seen in Fig. 3. It comprises of two lanes: main lane and merge lane. The main lane has a length of 230 m with a width of 4 m. It is populated with only agents at the beginning of the simulation where the agents move from left to right. In contrast, the merge lane only includes the ego vehicle and stretches for 50 m with the same width as the main lane, followed by a dead end.

Fig. 3. Ramp Merging scenario

[4]https://github.com/Achin17/highwayenv

2) Unprotected Left Turn: The unprotected left turn scenario is illustrated in Fig. 4. It consists of two roads : main road and left road, that are perpendicular to each other. The main road is populated with agents (on the top lane) and the ego vehicle (on the bottom lane). The agents move from right to left on the main road whereas the ego vehicle is initialised at the bottom lane of the main road and it's objective is to take an unprotected left turn onto the left road. The length of the main highway is 108 m with a width of 8 m whereas the length of the left highway is 40 m with the same width as the main highway.

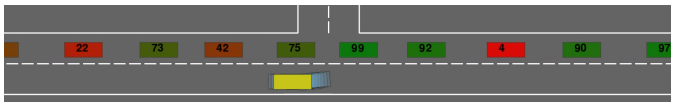


Fig. 4. Unprotected Left Turn scenario

VI. Experiments

In this section, we shed insights on the different types of traffic scenarios (Sec. VI-A) used to evaluate the efficacy of our proposed method. Then, we compare our proposed method against different baselines detailed in Sec. VI-B using evaluation metrics detailed in Sec. VI-C. Additionally, we present qualitative results (Sec. VI-D) and quantitative results (Sec. VI-E) for different policies in a variety of traffic conditions with varying degrees of cooperativeness to check their generalization capabilities in leveraging the cooperativeness of other agents. Finally, we present an ablation study in Sec. VI-F to determine the optimal number of time-steps for which same action queried from the Interactive Learner should be executed.

A. Evaluation Scenarios

We present simulation results for the following traffic scenarios:

- Cooperative: In this scenario, the majority of the agents are cooperative ($c_i \sim \mathcal{U}(2,4)$ m), implying that as soon as the ego vehicle shows intentions of merging into the main lane, the agent starts considering the ego vehicle 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 of mostly non-cooperative agents ($c_i \sim \mathcal{U}(0,2)$ m), meaning that the agents would not stop for the ego vehicle unless the ego vehicle'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 ego vehicle 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 agents. Evaluating policies in this scenario helps to assess the policy's generalization capabilities as the ego-vehicle is exposed to a much more diverse range of behaviors. This scenario also allows to assess if a policy can behave differently to cooperative and non-cooperative agents.

B. Baseline Policies

We compare our proposed method with a learning based approach and an optimization based method. For both the baselines and our proposed method, we control the longitudinal behavior of the ego vehicle. The lateral behavior of the ego vehicle is controlled by the LMPCC for all the policies as the lateral control is only contingent on the global reference path which is identical for all the policies.

- **RL**: A learning based method which involves learning a continuous policy using SAC algorithm. A reference velocity is sampled from the policy followed by computation of acceleration for the ego vehicle using equation $a = (v_{ref} - v^0)/\Delta t$.
  Since the interaction-aware behavior for the agents in the simulation is a function of lateral horizon parameter (y_H^0), an online optimization problem is still solved in the background using the reference velocity sampled from the policy.
- **LMPCC**: We use the state-of-the-art trajectory optimization method minimizing a contour and lag error to follow a reference path while respecting kino-dynamics and collision avoidance constraints. For evaluation purposes, the reference velocity is set to a constant reference, $v_{\text{ref}} = 3m/s$, and the weights are manually tuned to get the best possible performance while executing the merging maneuver.

Rule based methods such as IDM, MOBIL fail in highly dense traffic conditions and thus have not been included for evaluation purposes [7].

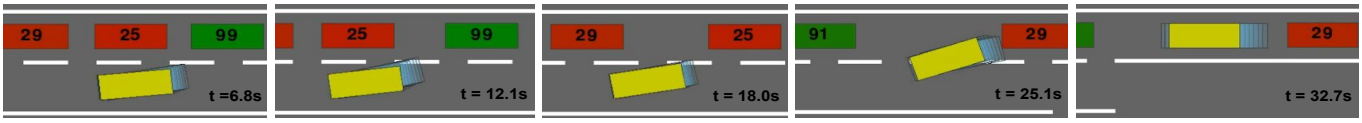
C. Evaluation metrics

To evaluate our proposed method, we compare our policy with the baseline policies in a total of 1200 episodes with 400 accounting for each of the cooperative, non-cooperative and mixed scenarios. The following metrics are used to compare the performance of learning based method, Optimization methods and our proposed method.

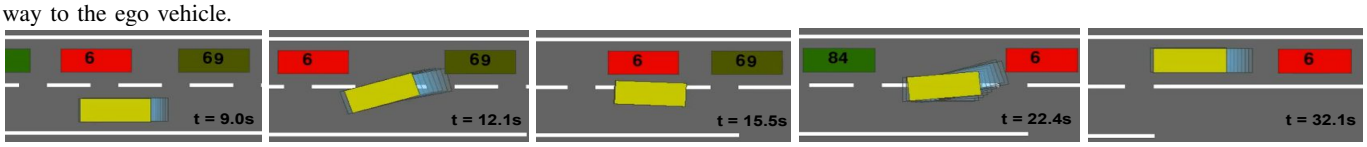
- *Success Rate*: This metric measures the number of successful merge maneuvers. A merge maneuver is deemed successful if the ego vehicle is able to reach the goal after merging on to the main highway before the time out.
- *Collisions*: This returns the number of collisions encountered by the ego vehicle during the roll-out of the policy.
- *Time-out*: This metric returns the number of times, the ego vehicle is not able to reach the goal position before the time out. This metric does not include those episodes that terminate due to collisions or infeasible solutions.

D. Qualitative Results

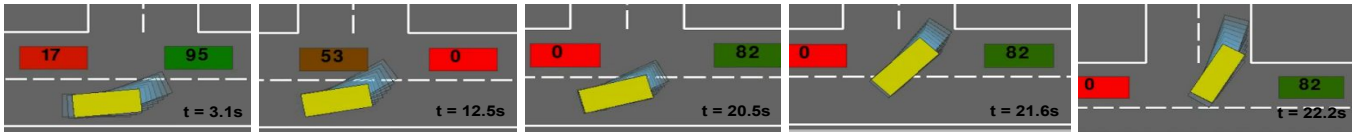
In this section, we present simulation results for two driving scenarios of our learned behavior, as depicted in Fig. 5. The qualitative results in Fig. 5(a) clearly show that our policy can successfully leverage the cooperativeness of other



a) This sequence of figures illustrate a typical behavior learnt by our policy. As the ego vehicle approaches the merging point, it tries to assess the reaction of its action on the agent titled "25" by inching closer to the main lane. The agent's non cooperative behavior does not elicit a response typical of agents that are willing to yield forcing the ego vehicle to stop. It tries the same with the agent titled "29" by creeping closer to the main lane but fails again. Finally, the merge is successful when a cooperative driver titled "91" emerges and gives way to the ego vehicle.



b) This sub-figure highlights one of the most important benefits of our approach. In this case, the global guidance provided by the RL agent wrongfully assumes the non-cooperative nature of the agent titled "6" to be cooperative. This guidance compels the ego vehicle to merge in front of the agent but the obstacle avoidance constraint forces the ego vehicle to steer away from the agent in order to avoid an impending collision. Finally, the agent merges in front of the cooperative agent titled "84".



c) This shows an unprotected left turn scenario where we use our proposed approach. Our policy is first trained in this environment and then evaluated. The behavior is similar to the one demonstrated by figure V-B2. This shows that our proposed method can be adapted to quite a different number of environments and is therefore, versatile in nature.

Fig. 5. All the scenarios were simulated by using the C-IDM model as discussed in subsection IV-D1 . The ego vehicle is represented in yellow whereas the future states, as computed by LMPCC have been shown in light blue. The agents on the main lane show a transition from red to green with red being non-cooperative (0) and green being cooperative (100). All the numbers in between show a continuous transition from non-cooperative to cooperative.

agents to perform the merging maneuver. Another example has been shown in Fig. 5(b) that illustrates one of the critical aspects of our implementation: safety guarantees that come with obstacle avoidance constraints. In this scenario, at 12.1 s, the ego-agent initiates a merging maneuver. However, the non-cooperative agent does not allow it and the local planner aborts and initiates a collision avoidance maneuver, at 15.5 s, merging successfully later when encountering a cooperative agent, at 22.4 s. Moreover, we demonstrate the generalization capabilities of our proposed method in Fig. 5(c), where we evaluate our method in an unprotected left turn scenario. Our policy successfully navigates the ego vehicle through dense traffic in an unprotected left turn scenario.

E. Quantitative Results

Aggregated results in Table II show that our method outperforms the baseline methods in terms of successful merges, number of collisions and infeasible solutions (Fig. 6) for all scenarios. The combined capability of Interactive Learner to implicitly embed inter-vehicle interactions into the policy and the safety provided by the collision avoidance guarantees allows our method to succeed in all the environments.

As far as optimization methods are concerned, the policy is biased towards executing more aggressive actions, exhibiting abysmal performance. The reason is the lack of assimilation of inter-vehicle interactions into the policy and a tracking error term in the cost function formulation that drives the ego vehicle towards the goal, disregarding any consequences of its actions on the nearby agents. This compels the ego vehicle to behave in a reactive manner, which leads to more collisions and infeasible solutions. In general, optimization methods fail to account for the effect of their actions on the nearby agents and thus do not consider the future evolution of states of the nearby agents, thereby exhibiting poor performance.

Regarding the comparison between RL and RL + LMPCC, both the methods achieve similar performance. The exception case is collision and success rate for non-cooperative scenarios which proves the superiority of our method over solely learning based methods. Our policy results in 15% more success rate and 5.5% less collision rate which can be attributed to the inclusion of obstacle avoidance constraint in the RL + LMPCC formulation.

To demonstrate our policy's ability to leverage agents cooperativeness explicitly, we evaluate 600 episodes in a mixed scenario where we track the cooperation level of agents in front of which the ego vehicle performs a successful merging maneuver. The results are shown in Fig. 7, which clearly illustrate our method's ability to identify cooperative agents by interacting with them successfully. A small number of successful merges can be seen with non-cooperative agents as well. This behavior can be attributed to the random sampling

TABLE II
RESULTS FOR EVALUATION OF DIFFERENT POLICIES FOR DIFFERENT TRAFFIC CONDITIONS IN MERGING SCENARIO

	Cooperative			Mixed			Non Cooperative		
	Success(%)	Collision(%)	Time-out(%)	Success(%)	Collision(%)	Time-out(%)	Success(%)	Collision(%)	Time-out(%)
RL	97.0	1.5	**1.5**	80.0	2.0	**18.0**	20.25	5.75	74.0
LMPCC	58.75	6.0	35.25	37.75	30.0	32.25	9.25	54.25	**36.5**
RL+LMPCC	**97.5**	**0.0**	2.5	**81.0**	**0.0**	19.0	**35.25**	**0.25**	64.5

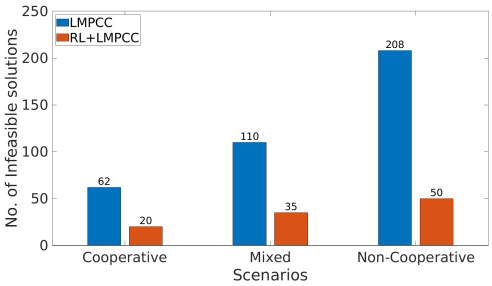


Fig. 6. Number of infeasible solutions encountered by the solver for different scenarios while evaluating the baselines.

of IDM parameters resulting in different agents' acceleration values. Thus, when moving from a standstill position, the agents might leave a gap big enough for the ego vehicle to merge onto the lane.

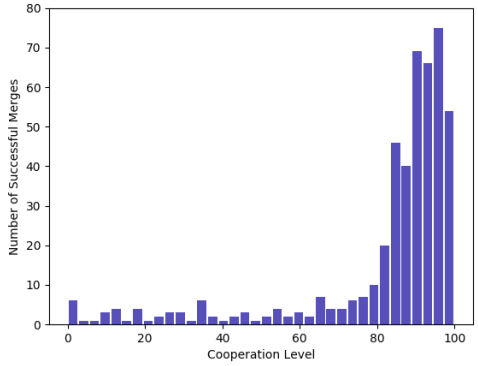


Fig. 7. This figure provides a comprehensive analysis of the cooperation level of agents (0 - non cooperative, 100 - cooperative) in front of which the ego vehicle was able to merge successfully.

F. Ablation study

In this section, we analyse the effect of the query's frequency to the Interactive Learner per number of control cycles K on the training performance. More specifically, every K control cycles we query the Interactive Learner for a new velocity reference. We assess the influence of training policies with different K values on the behavior of the final policy. All the policies are evaluated using $K = 1$ which means that the Interactive Learner is queried after every 0.1 s and then executed for just one time step.

The results have been summarized in Table III. The results clearly show that the policy trained with $K = 2$ outperforms other policies in terms of success and collision rate. The policy trained with $K = 1$ elicits an overly aggressive response from the ego vehicle which is evident from a really high collision rate and a low Time-out percentage. On the other hand, the ego vehicle exhibits an overly conservative behavior when evaluated with the policy trained with $K = 4$. This behavior can be attributed to the long duration (0.4 s) for which the same action is applied after querying from the Interactive Learner. After reaching the merging point, if the same action is applied for 0.4 s, it becomes highly likely that a collision will transpire for high reference velocities for the ego vehicle. This compels the Interactive Learner to learn to sample really low or zero reference velocities to avoid an impending collision resulting in an overly conservative behavior.

The policy trained with $K = 2$ elicits a balanced response from the ego vehicle that is neither too conservative nor too aggressive resulting in a high success rate and a low collision rate for all the scenarios.

VII. CONCLUSION

We present a novel navigation framework for maneuvering through highly interactive and dense traffic scenarios in a timely and safe manner. We combine Deep Reinforcement Learning's capability of learning complex policies from simulated data with an optimization-based planner that can provide safety guarantees and compute kino-dynamically feasible trajectories. We train and evaluate our proposed method in a highly interactive simulation environment capable of simulating diverse range of human behaviors. Moreover, the simulation takes autonomous vehicle's future plan into account and thus, elicits non-reactive behavior from human drivers. The results show that our method outperforms solely learning based and optimization based planner in terms of collisions, successful maneuvers and fewer deadlocks. We also showcase our method's generalisation capabilities in a different dense traffic scenario (unprotected left turn). Our proposed method can easily handle interaction with multiple traffic participants exhibiting a wide variety of behaviors and compute a collision-free trajectory while respecting kino-dynamic and real-time constraints and provide safety guarantees.

Future works involves investigation into different neural network architectures for guidance policy construction capable of handling inputs from a dynamic number of traffic participants. Moreover, simulation needs to be validated by comparing it with real life dataset of a highly interactive dense traffic scenario.

TABLE III
RESULTS FOR ANALYSING THE EFFECT OF THE QUERY'S FREQUENCY TO THE INTERACTIVE LEARNER PER NUMBER OF CONTROL CYCLES K ON THE FINAL BEHAVIOR OF THE POLICY

	Cooperative			Mixed			Non Cooperative		
	Success(%)	Collision(%)	Time-out(%)	Success(%)	Collision(%)	Time-out(%)	Success(%)	Collision(%)	Time-out(%)
K = 1	90.25	**0.0**	9.75	75.75	0.0	24.25	33.5	0.25	66.25
K = 2	**97.5**	**0.0**	**2.5**	**81.0**	**0.0**	**19.0**	**35.25**	0.25	**64.5**
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

REFERENCES

[1] 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. 060117–105157, 2018.

[2] M. Bansal, A. Krizhevsky, and A. Ogale, "ChauffeurNet: Learning to Drive by Imitating the Best and Synthesizing the Worst," pp. 1–20, 2019.

[3] A. Sadat, S. Casas, M. Ren, X. Wu, P. Dhawan, and R. Urtasun, "Perceive, Predict, and Plan: Safe Motion Planning Through Interpretable Semantic Representations," 2020.

[4] 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.

[5] 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.

[6] 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.

[7] D. M. Saxena, S. Bae, A. Nakhaei, K. Fujimura, and M. Likhachev, "Driving in Dense Traffic with Model-Free Reinforcement Learning," 2019.

[8] 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.

[9] 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.

[10] 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.

[11] B. Zhou, W. Schwarting, D. Rus, and J. Alonso-Mora, "Joint Multi-Policy Behavior Estimation and Receding-Horizon Trajectory Planning for Automated Urban Driving," *2018 Ieee International Conference on Robotics and Automation (Icra)*, pp. 2388–2394, 2018.

[12] 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.

[13] 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)*, pp. 1126–1133, 6 2015.

[14] 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.

[15] 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.

[16] J. F. Fisac, E. Bronstein, E. Stefansson, D. Sadigh, S. S. Sastry, and A. D. Dragan, "Hierarchical Game-Theoretic Planning for Autonomous Vehicles," pp. 9590–9596, 2019.

[17] 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.

[18] 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.

[19] M. Bouton, A. Nakhaei, K. Fujimura, and M. J. Kochenderfer, "Cooperation-Aware Reinforcement Learning for Merging in Dense Traffic," 2019.

[20] 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.

[21] 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.

[22] 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. 2492–24978, 2019.

[23] S. Li, N. Li, A. Girard, and I. Kolmanovsky, "Decision making in dynamic and interactive environments based on cognitive hierarchy theory, Bayesian inference, and predictive control," *Proceedings of the IEEE Conference on Decision and Control*, vol. 2019-Decem, pp. 2181–2187, 2019.

[24] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft Actor-Critic : Off-Policy Maximum Entropy Deep Reinforcement," *Icml 2018*, vol. 80, pp. 1861–1870, 2018.

[25] 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.

[26] E. Leurent, "An Environment for Autonomous Driving Decision-Making," 2018.

[27] M. Treiber, E. Kometani, T. Sasaki, H. A. H. D. Treiber, M., M. Treiber, A. Kesting, A. K. Das, and J. Asundi, "Congested traffic states in empirical observations and microscopic simulations," *The American Physical Society*, vol. 2, no. 2/3, pp. 1805–1824, 2000.

APPENDIX A

The proportion of cooperative and non-cooperative agents during training has major influence on the learned policy's cooperativeness/aggressiveness. The behavior of the learned policy is contingent on the cooperativeness of agents encountered during the training process. For instance, if the ego vehicle is exposed to solely cooperative agents, the learned policy will result in aggressive actions as the ego vehicle will always expect the agents to yield, resulting in many collisions during evaluation where the ego vehicle faces a lot more diverse range of behaviors. On the contrary, if the ego vehicle encounters only non-cooperative agents during training, the resultant policy will result in very conservative actions leading to *Freezing Robot Problem*. This section sheds insights on the procedure followed to find the optimal ratio of cooperative and non-cooperative agents that elicits balanced behavior from the ego vehicle.

A. Ablation study for cooperativeness variable (c_i)

In our work, the sampling distribution of cooperativeness variable c_i controls the ratio of cooperative and non-cooperative agents encountered by the ego vehicle during training. The intervals for the sampling distribution of c_i are estimated in an experimental manner. The values of c_i are sampled from three different uniform distributions during training followed by evaluation of the final policy in a diverse range of scenarios. The three uniform distributions are selected as follows:

- $c_i \sim \mathcal{U}(0, 4)$ m
- $c_i \sim \mathcal{U}(1, 4)$ m
- $c_i \sim \mathcal{U}(2, 4)$ m

By incrementing the minimum sampling value w_{min} of c_i, there is an increase in the proportion of the agents that start yielding to the ego vehicle as soon as it shows any intention of merging in (e.g, by moving closer to the main lane). The final policy obtained using three different sampling distributions is evaluated using RL+LMPCC as it outperforms other policies across the board (as can be seen in Table II). The values of c_i during evaluation are sampled from $c_i \sim \mathcal{U}(2, 4)$ m to ensure a fair evaluation for all the three training sampling distributions. Evaluations are done using 1200 episodes in total with 400 accounting for cooperative $c_i \sim \mathcal{U}(3, 4)$ m, mixed $c_i \sim \mathcal{U}(2, 4)$ m and non-cooperative $c_i \sim \mathcal{U}(2, 3)$ m scenarios.

The results have been shown in Table IV. The results clearly show that for the merging scenario, the policy trained using c_i values sampled from $c_i \sim \mathcal{U}(0, 4)$ m learns the most balanced behavior that outperforms other c_i training sampling distributions across the board. As the proportion of cooperative agents increase during training (for $c_i \sim \mathcal{U}(1, 4)$ m and $c_i \sim \mathcal{U}(2, 4)$ m), the policy tends to be biased towards executing more aggressive actions which can be seen by a high collision rate and a decreasing Time-out percentage. This can be attributed to high reward gained by sampling aggressive actions (high v_{ref} values) during training leading to successful maneuvers due to the presence of mostly cooperative agents. Thus, for our final policy we sampled $c_i \sim \mathcal{U}(0, 4)$ m to elicit a balanced behavior from the ego vehicle.

APPENDIX B

This section sheds insights on the effects of obstacle avoidance constraints during training on the behavior of the final policy. Learning navigation in dense traffic using RL requires being aware of the effects of one's own actions on the neighboring agents. This involves getting feedback on actions that result in a collision or an infeasible solution by the solver. The learning algorithm needs to be cognizant of the states preceding a collision or an infeasible solution. This can only be achieved if the ego vehicle interacts closely with the neighboring agents. However, inclusion of obstacle avoidance constraints in the motion planning framework might restrict movement with the nearby agents.

In our proposed method, obstacle avoidance constraints are incorporated in the LMPCC framework as inequality constraints using eqn. 5. Depending on $\bar{A}^{\text{ego}}(\boldsymbol{p})$ and $\bar{A}^{\text{obs}}$ (Sec. IV-C2), the ego vehicle can either interact closely or maintain its distance and may be never be exposed to perilous states. Thus, it is imperative to study the influence of these constraints during the learning process on the behavior of the final policy. To quantitatively assess the effects of training with obstacle avoidance constraints, we compare a policy where obstacle avoidance constraints are enabled during the training process with a policy where obstacle avoidance constraints are disabled. To summarize, we compare the policies trained with the following criteria:

- *Obs on*: In this formulation, the policy is trained with obstacle avoidance constraints enabled.
- *Obs off*: In this formulation, the policy is trained with obstacle avoidance constraints disabled.

We evaluate the aforementioned training methods in merging scenarios (see Sec. V-B1) for three different traffic conditions (see Sec. VI-A) on the basis of the metrics defined in Sec. VI-C. The results have been shown in Table V. The results clearly show that *Obs off* performs the best in terms of successful merging maneuvers and collision rate.

These results can be attributed to the behavior learnt during training with the inclusion of obstacle avoidance constraints. With obstacle avoidance constraints enabled during training, the ego vehicle becomes too reliant on these constraints to steer it out of an impending collision state. This leads to aggressive behavior during the evaluation of the policy where the ego vehicle gets too close to the agents on the main lane and then expects the other agent to yield. Moreover, over-dependency on Obstacle avoidance constraints and close proximity to non-cooperative agents that don't yield culminates in high number of collisions.

Obstacle avoidance constraints are meant to be use as a safety precaution to assure obstacle avoidance and not as a means to model interactions to decide if other agents yield or not. Therefore, we train our final policy with obstacle avoidance constraints disabled.

APPENDIX C

This section sheds insights on the various implementation aspects of our method.

A. Intelligent Driver Model Parameters

A modification of the Intelligent Driver Model (IDM) was implemented to achieve interaction-aware behavior for the agents (Sec. IV-D1). However, the underlying model that controls the longitudinal behavior of the agents is still IDM which has been represented by eqn. 9.

$$\begin{aligned} \dot{x} &= \tfrac{\mathrm{d}x}{\mathrm{d}t} = v \\ \dot{v} &= \tfrac{\mathrm{d}v}{\mathrm{d}t} = a\left(1 - \left(\tfrac{v}{v_0}\right)^{\delta} - \left(\tfrac{s^*(v,\Delta v)}{s}\right)^2\right) \\ &\text{with } s^*(v, \Delta v) = s_0 + vT + \tfrac{v\Delta v}{2\sqrt{ab}} \end{aligned} \tag{9}$$

where v_0, s_0, T, a, b and δ are constants. v_0 represents the desired velocity, s_0 gives the desired minimum distance to the

TABLE IV
RESULTS FOR DIFFERENT TRAINING PROCEDURES EVALUATED FOR DIFFERENT SAMPLING DISTRIBUTIONS OF THE COOPERATIVENESS VARIABLE (c_i)

	Cooperative			Mixed			Non Cooperative		
	Success(%)	Collision(%)	Time-out(%)	Success(%)	Collision(%)	Time-out(%)	Success(%)	Collision(%)	Time-out(%)
$c_i \sim \mathcal{U}(0,4)$ m	**100.0**	**0.0**	**0.0**	**97.5**	**0.0**	2.5	39.0	**0.0**	61.0
$c_i \sim \mathcal{U}(1,4)$ m	99.5	**0.0**	0.5	86.5	11.5	2.0	79.75	15.25	5.0
$c_i \sim \mathcal{U}(2,4)$ m	99.75	**0.0**	0.25	91.0	9.0	**0.0**	**86.75**	13.25	**0.0**

TABLE V
RESULTS FOR DIFFERENT TRAINING PROCEDURES WITH REGARDS TO OPERATIONAL STATUS OF OBSTACLE AVOIDANCE CONSTRAINTS

	Cooperative			Mixed			Non Cooperative		
	Success(%)	Collision(%)	Time-out(%)	Success(%)	Collision(%)	Time-out(%)	Success(%)	Collision(%)	Time-out(%)
Obs on	70.0	29.5	**0.5**	42.25	57.5	**0.25**	0.0	100.0	**0.0**
Obs off	**97.5**	**0.0**	2.5	**81.0**	**0.0**	19.0	**35.25**	**0.25**	64.5

vehicle in front, T represents the minimum possible time to the vehicle in front, a gives the maximum acceleration, b gives the comfortable braking deceleration and δ is the acceleration exponent. Δv is the difference in velocities of the vehicle in front and the ego vehicle. The value of these parameters have been shown in Table VI.

TABLE VI
PARAMETERS FOR IDM

Parameter	Value
v_0	$\mathcal{U}(3,4)$ m/s
s_0	$\mathcal{U}(2,3)$ m
T	0.5 s
a	$\mathcal{U}(1,2)$ m/s^2
b	$\mathcal{U}(-2,-1)$ m/s^2
δ	$\mathcal{U}(3,4)$

The values of the parameters have been chosen to make sure there are no collisions among the agents while exhibiting a diverse range of human behaviors by means of sampling. These values result in dense traffic conditions which helps in training and evaluating the efficacy of our proposed approach.

B. LMPCC weights

The weights for the eqn. 3 are manually tuned to ensure proper tracking of the reference path. The weights have been shown in Table VII.

TABLE VII
WEIGHTS FOR LMPCC

Weight	Value
w_1	10
w_2	0.2
w_3	20
w_4	0.1
w_5	10

C. Reward function parameters

The parameters of the reward function have been shown in Table VIII. A high negative reward is assigned to the ego vehicle in case of a collision or an infeasible solution while a small negative reward is allocated to discourage close movement with other agents.

TABLE VIII
REWARD FUNCTION CONSTANTS

Parameter	Value
$r_{infeasible}$	-150
$r_{collision}$	-150
r_{near}	-1.5