Institute of Architecture of Application Systems

University of Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Master Thesis

Occlusion Handling in Behavior Planning Using Imitation Learning for Autonomous Driving

Janaranjani Palaniswamy

Course of Study: M.Sc. Information Technology

Examiner: Prof. Dr. Marco Aiello

Supervisor: Dr. Stoll Martin, Robert Bosch GmbH,
Dr. Dolgov Maxim, Robert Bosch GmbH

Commenced: November 17, 2021

Completed: May 17, 2022

Acknowledgement

The successful completion of this master thesis would be incomplete without the support and help from many people.

First and foremost, I would like to express my heartfelt gratitude to Dr. Stoll Martin for giving me the opportunity to work on this exciting thesis topic, and for his constant support, advice, and supervision throughout this work. I learned a lot from him, not only the technical insights but also the way of scientific thinking. I would also like to thank Dr. Dolgov Maxim, for his valuable guidance and persistent encouragement. I take this opportunity to also thank my internship supervisor, Mr. Hoch Marco, without whom this opportunity would not have been possible.

I would like to express my deep gratitude to Prof. Dr. Marco Aiello for his valuable feedback, examining my thesis work, and reviewing all my steps during this thesis.

I extend my heartfelt thanks to the rest of the Master's program in Information Technology faculty at the University of Stuttgart for teaching and providing me with countless opportunities to develop my career.

I am grateful to my classmates who have been very supportive throughout my study. My special thanks to my friends for being my support system during my thesis completion.

Last but not the least, I would like to thank my family whose love has served as a driving force inside me throughout the study period. I dedicate the thesis to my father Palaniswamy, my mother Kavitha, and my younger brother Dharani Vel. I express my immense gratitude to them for all the positive thoughts, love and strength throughout my Master's studies.

Abstract

Commissioning a self driving vehicle to run on road, requires the facilitation of complete vehicle system to work at all conditions. Behavior planning is a crucial part of the autonomous driving system and it is important to ensure safe and comfortable navigation of the ego vehicle. More advancements are required to enhance the data-driven approaches for the planning systems. The urban driving scenarios always possess a variety of disturbances and inefficiencies. In which, the roundabout is a challenging driving task where uncertainties are caused due to static priority rules and occlusions that limits the field of view for the ego vehicle. Thus behavior planning must make sure to consider the uncertainty of limited visibility of the environment explicitly. Although machine learning-based approaches show promising results for behavior planning. A single planner cannot handle all other urban driving scenarios. Hence, an imitation learning-based technique can help the behavior planner to mimic the human expert behavior. In this context, an end-to-end planning system based on imitation learning proposed by Waymo is used. The behavior planning framework makes use of mid-level input and output representations making it viable to be interfaced with existing vehicle system. The planner outputs a set of waypoints to drive the vehicle controller. However, the existing imitation learning-based planning framework with the Intelligent Driver Model (IDM) as an expert and policy model made of a multi-task network did not address this use case of occluded roundabouts. As the default IDM generates training data with a visibility of the environment, there arises a need for a strategic approach to handle the occluded environments. This thesis work aims at leveraging the existing planning system to handle the situations in a roundabout with limited visibility. Ultimately, the goal is to train the policy model with more realistic data and enable it to make safe and comfortable driving decisions. For this purpose, an occlusion algorithm is implemented to induce limited visibility of the roundabout environment in simulation. And the expert model is enhanced to handle the limited field of view much similar to how a human driver behaves. Consequently, the training dataset generated from the expert is upgraded with an additional input feature. This add-on feature in the input data provides enough knowledge for the policy to perform well in the occluded environment. A study on modern architecture search is performed and a suitable convolutional network is adopted as the backbone for this multi-task model. The enhanced behavior of the proposed approach is demonstrated via detailed quantitative analysis. For this purpose, a new comfort metric is defined and used as Key performance Indicator (KPI) to evaluate the models. An ablation study is conducted with the expert and confirmed that the new extended IDM behaves more carefully in an occlusion environment. In the end, the influence of the training data is inferred by a detailed comparison of the policy model in default and occlusion environments with different dataset configurations. The importance of more realistic data is realized and also shows that the policy model can imitate the expert behavior well enough. It is exhibited that the proposed methodology can handle the occlusions in the complex roundabout situations in simulation.

Contents

List of Figures

List of Tables

1 Introduction

The autonomous driving industry has seen a rapid growth in terms of research and development since the last decade. Despite of the hard challenges the rate of advancement is consistent over the years. An autonomous driving vehicle comprises of three major systems: Perception, Planning and Control. Similar to a human driver, the perception system is comparable to the human's eyes, planning to human's brain, and consciousness and control to the human's arms and legs. Perception system perceives the environment around the vehicle and localizes the vehicle. Whereas, the planning system operates based on the information from perception to decide on the route to travel and generates the trajectory path accordingly. In this context, planning [65] is a critical task in safe navigation of autonomous vehicle through the environment. In which behavior planning involves making tactical driving decisions like distance keeping, neighboring vehicle interactions. Here the autonomous vehicle must consider the behaviors of other traffic participants as well. Decisions must be made with caution and foresight. The capabilities of behavior decision-making are decisive for enabling safe and socially acceptable autonomous driving in all traffic scenarios. Inappropriate actions may force the automated vehicle itself or another vehicle to perform an uncomfortable or even dangerous maneuver. Improving the intelligence of behavioral decision-making systems is one of the core objectives for autonomous driving. Making safe decisions involves following traffic regulations and communicating with other road users in a reasonable manner. However, due to the complexity of the road topology there exists lot of uncertainties, such as limited sensor measurements, ambiguous motion and intention prediction, and occluded objects. In order to make a good behavior planning system, it requires to learn multiple conflicting objectives like, safety, efficiency, and avoiding collisions among the traffic participants.

Despite many advances in machine learning, still majority of the decision making and planning approach are based on rule-based approaches in the industry. Based on the discussion from Jain et al. [35], it is clear that a large proportion of machine learning resides in the perception and prediction parts of the autonomous driving stack. However, machine learning is not yet in use with the planning systems. Accordingly, rule-based decision-making systems are extensively utilized for behavior planning. Such methods are initially adopted in the DARPA Urban Challenge [17] by most of the teams. Such rule-based systems manually simulate the best decisions while considering all safety and optimality features. They are more inline with human decision making. For simple driving circumstances, rule-based approach is more reliable and straightforward to interpret. On the other hand, such manually designed planning method is incapable of ensuring safe and efficient driving in a complicated environment with various uncertain situations. In addition, such methods lacks the ability to generalize. This leads to the need of more intelligent data-driven solutions in decision making approaches. On the whole, it is also learned that a single approach cannot handle all possible driving scenarios as stated by Lin et al. [48]. There arises a need of special attention to each driving scenario to attain an optimum driving behavior to enable self driving vehicles in the near future.

1.1 Motivation

A wider capabilities of self-driving vehicles were demonstrated in DARPA Urban Challenge in 2007 [9]. In this challenge, it was exposed that, with a detailed planning framework, an autonomous vehicle might handle a range of urban driving circumstances [55]. It is also noted that, the performance of autonomous driving system is far from the human driving quality. This is mainly due to the limitations of planning systems in handling complex driving scenarios. Among various complex intersections in the urban environment, roundabouts pose an exciting challenge for behavior planning autonomous driving. In this case, when an ego vehicle must enter the roundabout, the interaction among the traffic participants determines the decision [3]. An approaching car must calculate if it can enter without obstructing the flow of traffic. This interaction is highly dynamic since entering traffic does have to stop if the roundabout is not free. In general, each conceivable sequence of ego vehicle action is a plan that may be evaluated using a cost function created by hand. The optimal plan is a set of actions that constitutes minimal cost. In addition, the presence of occlusions in the roundabout elevates the problem of decision-making. Here the ego must deal with visible vehicles as well the vehicles that are blocked by occlusions. The predicted behavior of autonomous vehicle must always be like a human behavior with a broader vision and faster reaction. On comparison with a human driver, a typical driver, would follow creep-and-go method when confronted with roundabout intersection. A human driver usually goes forward carefully, while keeping an eye on the approaching traffic inside the roundabout ahead as he prepares to turn right. In this manner, he gradually improves visibility of the environment until there is a safe distance for turning.

While it is hard to design a planning system to navigate through an occluded roundabout, a human driver can solve it easily. An approach to learn the behavior planner from human driver experts is using imitation learning technique. The benefits of applying imitation learning technique are: firstly there is no need for manual policy design; secondly the planner only needs expert driving data which can be obtained using existing simulation environments. The existing works of imitation learning for driving focuses on directly predicting the control commands from raw sensor inputs [11, 14, 60] and can handle only simple driving tasks like lane following.

1.2 Problem Statement and Objective

Behavior planning is a fundamental part of autonomous vehicles, which must perform decision-making and trajectory planning tasks while coping with occlusions and uncertainty regarding the surrounding vehicles' behavior. Traffic intersections where vehicles interact with each other and cross surrounding traffic streams are particularly challenging. Among other reasons, this is due to the additional difficulties encountered in those scenarios to obtain accurate predictions and the need to generate safe and efficient plans. There are recent publications [3, 33, 51, 72] that address the challenge of handling occlusions at the intersections. However, the existing solutions are not based on end-to-end learning and also do not leverage the behavior planning system to handle the limited visibility situations in a roundabout scenario. Even the work from ChauffeurNet [2] did not address this use of occlusion handling in behavior planner. Also the existing works on imitation learning has failed to guarantee a safe driving behavior in the occluded roundabouts. In this context,

a novel workflow is proposed to facilitate imitation learning-based behavior planning to ensure safe navigation of the autonomous vehicle at populated roundabouts where dynamic and static obstacles cause occlusions.

The goal of this thesis is to develop a solution for the behavior planning system to deal with occlusions in a roundabout highway environment. The focus is on occlusions, which can have different sources such as buildings and other infrastructure elements, dynamic objects (e.g. trucks) occluding the line of sight, sensor degradation (e.g. dirt), etc. The entry to a roundabout is the focus of this research since it depicts a common traffic condition in which uncertainties play a significant role. This method can be used to solve a variety of problems. The major purpose of this planning is to choose the best acceleration while driving through a roundabout on a pre-defined route while maintaining a target speed, reducing braking, and avoiding crashes similar to what was addressed in [3]. In autonomous driving, there is a lot of emphasis on end-to-end learning approaches that primarily focus on predicting raw control outputs like steering or braking while consuming raw sensor inputs. The sample complexity is the main downside of this approach, so to overcome it, a mid-level input and output representation similar to [2] and [67] is adopted in this work. A planning system using such mid-level representation can be easily interfaced with any existing perception and control systems. Moreover, the network has the freedom to be trained on real or simulated data and can be readily tested and validated in closed-loop simulation before being deployed into the car, which is a significant advantage of mid-level representation. Thus, through this work, the problem of traversing through the roundabout with occluded regions is formulated and an approach using imitation learning is used to deal with occlusions in various driving scenarios.

1.3 Contribution

An imitation learning approach is being used to tackle the problem of occluded regions in a roundabout environment. The work begins with occluding a specific region of a roundabout environment. The decisions made in the design of the occlusion zones are detailed. As a part of imitation learning, the expert demonstrations are collected from the Intelligent Driving Model (IDM) [62]. The IDM is extended to handle the occluded zones and their ability to exhibit a safe driving behavior is assessed using suitable metrics. The extended IDM model is used to generate a set of demonstrations depicting how handle the limited visibility of the environment. The demonstrations are then used to generate the input data in the form of top-down representation. These grid images are then fed into the model architecture as input. Various convolutional layer architectures are evaluated, and an optimal model approach is chosen in the end. The impact on various data components is investigated, and the best optimal model is evaluated by comparing it to the IDM. A complete ablation study is conducted to test the entire strategy of occlusion handling of the behavior planner. Various driving scenarios are taken into consideration and appropriate Key Performance Indicators (KPI) are developed to support the detailed evaluation.

In summary, the main contributions of this work are

- Developed an algorithm that includes dynamic occlusions in the simulation environment to limit the visibility region for ego vehicle.
- Proposed an enhancement for the rule-based expert model to handle the risk in the limited visibility environment.

- Defined suitable KPI metric to evaluate the new expert behavior against the baseline.
- Enhanced the input grid images with visibility information of the surrounding to enable the policy model learn from the enhanced expert demonstrations more efficiently.
- With detailed search on modern architecture, a new backbone model suitable for the state-of-the-art policy network is proposed to use.
- Quantitative measures are used to evaluate the expert and policy models based on the influence of occluded zones in the environment.

1.4 Outline

The rest of this report is organized into several chapters as follows:

- Chapter 2 provides the necessary background knowledge to understand the fundamental terms and concepts discussed in this thesis work.
- Chapter 3 gives an overview of existing research works that are relevant to this thesis topic. The framework baseline planning system is described and its components are explained in detail. Furthermore, based on the analysis, research questions are formulated concerning to occlusion handling in behavior planning.
- Chapter 4 presents the detailed description of the approach proposed in leveraging the baseline behavior planning framework to handle the occluded environment.
- Chapter 5 illustrates the training procedure and implementation details. The results from policy training are discussed in detail.
- Chapter 6 summarizes about the evaluation setup with suitable scenarios and metrics used for closed-loop evaluation. A detailed discussion is made on the experimental evaluation of expert and policy models. A comprehensive analysis are made based on the influence of training data in policy training. This chapter ends with analysis on the failure cases observed during quantitative evaluation.
- Chapter 7 provides a summary of this thesis work and concludes the whole project. Moreover, it gives a detailed elaboration on possible future work.

2 Background Information

The topics necessary to obtain sufficient background knowledge to understand the technical concepts of the method and comprehensibility of the results in this thesis are discussed in this chapter. The topics cover the importance of behavior planning in autonomous driving system followed by a detailed discussion on behavior planner and imitation learning paradigm. These are brought closer to the reader in the following sections.

2.1 Planning System in Autonomous Vehicle

Autonomous driving vehicles traverse through the environment without the need of human involvement. Regardless of how automated driving functions are classified according to SAE J3016, they are always a technical system that senses, processes, and issues a command. This section explains the technical process inside the autonomous vehicle system and how behavior planning is integrated into the vehicle system.

2.1.1 Autonomous Driving System

According to Pendleton et al. [56], the main competencies of autonomous driving system is categorized into perception, planning and control systems. Figure 2.1, depicts the detailed overview of major components of autonomous vehicle system with a clear distinction between hardware and software parts. The focus of this thesis is narrowed to the software subsystem.

The perception component receives information about the environment through sensors data and constructs an environmental model. Some examples of perception tasks are lane markings, obstacle detection and traffic sign detection. The end of this chain is the control system, which executes the planned maneuvers via actuators and achieves the intended driving behavior.

In between is the component of our focus, the planning system. The process of making deliberate decisions in order to fulfil the vehicles goals to reach the destination point while achieving safety and optimal route, is referred to as planning. Planning takes the output of perception i.e. the environment model as the input and derives the best possible driving trajectory.

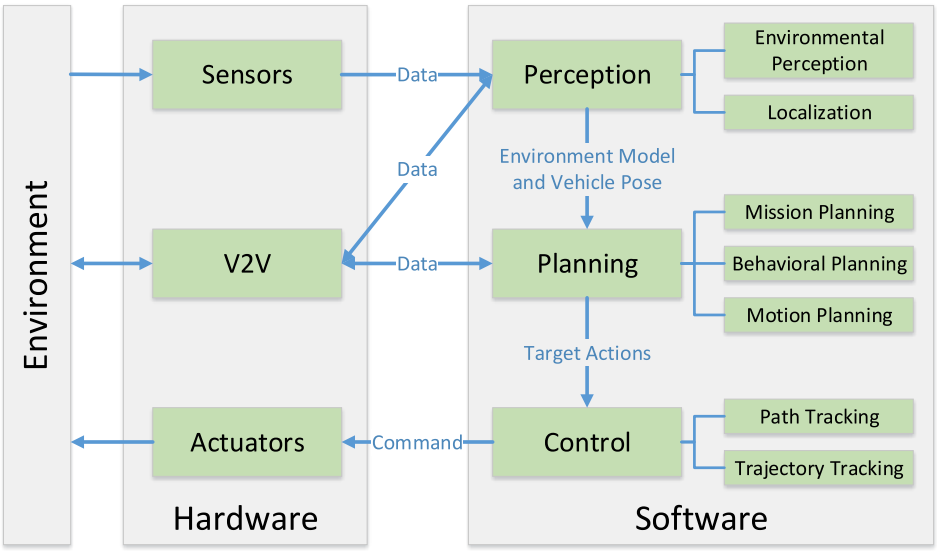


Figure 2.1: Autonomous Vehicle System Overview [56].

2.1.2 Behavior Planning

As per Ilievski et al. [30] the planning task is decomposed into three main sub-tasks: mission planning (or route planning), behavior planning (or decision-making) and local planning (or motion planning). The planning sub-tasks are illustrated in Figure 2.2.

Mission planning is the process of making the highest-level decisions about the proposed journey, such as deciding the sequence of roads to take, given the desired destination, the vehicle's current position, the user's preferences, and a priori assumptions about road conditions and availability. The mission planner must re-plan if its assumptions are disputed along the way, or if the user makes ad hoc adjustments to the journey's needs. We do not explore this part of motion planning in the current work because mission planning is already a well-discussed subject in the literature.

Figure 2.2: Motion Planning Architecture [30].

Given the route plan from the mission planner, the behavioral planner (or decision maker) makes ad hoc decisions to correctly interact with other agents and observe rules constraints, and thereby generates local objectives, e.g., changing lanes, overtaking, or proceeding through an intersection. The behavioral planner is mainly in charge of making decisions to ensure that the vehicle observes all applicable road laws and interacts with other agents in a normal, safe manner while progressing incrementally along the mission planner's path.

The motion planner (or local planning) aims to generate safe and smooth paths to achieve local objectives. Typically, local planning is used to move the autonomous vehicle from its current position to a desired position while avoiding obstacles, satisfying comfort criteria, and generally adhering to kinodynamic restrictions. As a result, the behavior planner must ensure to consider the current state of the ego vehicle and environment. Having discussed about each sub-tasks of the planning system, behavior planning is of at most importance to this work.

To move through the environment, the behavior planner must create a sequence of discrete high-level control actions. Basic motions like speeding up, slowing down, and stopping are examples of control actions. The actions must also be consistent with current road conditions, which means they must be developed online based on sensor data. As a result, perception is an important component of behavior planning, but it comes with its own set of obstacles, such as noise, occlusions, and sensor fusion. In this thesis, the problems of occlusions are acknowledged by assuming that the motion planner is given a limited visibility view of the ego vehicle and the external environment. Despite the challenges of inaccurate assumptions and sensing, the behavior planning must respond to the dynamic nature of the world.

2.2 Imitation Learning Paradigm

The goal of imitation learning is to learn to imitate the behavior of the expert. It's more suitable for autonomous driving applications, which can easily acquire a large amount of human driving data. At data collection phase, an expert (a human driver) receives observation o_t and outputs a set of actions $a_t$at time step t. This observation-action pairs are stored as the dataset, $\mathscr{D} = \left\{ \left(\mathbf{o}_i, \mathbf{a}_i \right\rangle \right\}_{i=1}^{N}$. The policy function is $fo;\theta), where\theta$ is its parameter. f can be any function approximator, such as deep neural network as used in this thesis work. Using the collected dataset $\mathscr{D}$, the aim of imitation learning is to learn a policy $\phi : \mathrm{S} \rightarrow \mathrm{A}$ that mimics the experts behavior in an optimized manner [74].

The imitation learning problem can be formulated as a supervised learning problem which is similar to the behavior cloning. Where the objective is to optimize the policy parameter θ while minimizing the loss function $\mathscr{L}$ as stated in equation 2.1. Ultimately, the main goal is to match the learned policy function f to that of its expert function f_E.

$$\min_{\theta} \underset{(\mathbf{o}_i, \mathbf{a}_i) \in \mathscr{D}}{} \mathscr{L} \left(f \left(\mathbf{o}_i; \theta \right), \mathbf{a}_i \right) \tag{2.1}$$

An implicit assumption behind this formulation is that the expert's actions are fully explained by the observations; that is, there exists a function E that maps observations to the expert's actions: $a_i = E\left(\mathbf{o}_i, \mathbf{h}_i\right)$. Given enough data, a sufficiently expressive approximator will be able to fit the function E if this assumption holds. This explains why imitation learning works so well for tasks like behavior planning.

3 State-of-the-Art

A detailed review of the literature works related to this thesis topic of occlusion handling in behavior planning are described in this chapter. As imitation learning approach is the main focus to achieve as a part of end-to-end learning. A brief overview of all the existing research works to the best of our knowledge are discussed, starting from occlusion handling in planning, probabilistic planning methods, learning-based planning approaches, and also the literature works in the area of roundabout planning is also reviewed. In addition, the overview about the Bosch in-house planning system is described. Furthermore, a detailed elaboration of various modules involved in the construction of a behavior planning system is presented.

3.1 Occlusion Handling Methods

The occlusion handling planners frequently applies probabilistic models. With the rise of learning-based techniques, machine learning-based approaches are predominantly applied in recent times to solve the problem of occlusions in a complex urban driving intersections.

3.1.1 Probabilistic Methods

Probabilistic techniques, generally assumes a distribution of participants inside an occluded area and adjusts behavior based on the likelihood of interference with the other traffic participants. Based on a probabilistic tracking method, a planning algorithm can consistently handle the occlusions from larger vehicles, such as buses in their own lane [28]. However, such a strategy necessitates the prior detection of the object before entering the obstructed area. A POMDP as a probabilistic model has advantages in terms of generating interaction-aware behavior. Thornton et al. [68] uses the POMDP method to deal with a stationary car parked in front of a crosswalk. The amount of unobservable grids in the discretized state space is used to model occlusions. However, they can only control longitudinal acceleration because the pedestrian is not moving. In [8], a POMDP model is tested for merging onto a priority road while static objects occlude the field of view. The visibility of the corresponding vehicle is indicated by a boolean flag. They search for intersections between a straight line from the respective vehicle to the autonomous car and other objects to calculate visibility. The number of cars in the occluded region is believed to be known ahead of time. Later Bouton et al. [7] presented a scalable strategy for dealing with numerous occluded traffic participants based on utility fusion. For each vehicle in a scenario, the authors solve a POMDP independently. A sum or minimum utility function is used to integrate the generated pairs of beliefs and behavior pairs. The authors show that the computation time scales linearly with the number of identified traffic participants when comparing two offline approaches, QMDP and SARSOP. However, the best behavior must be calculated in advance.

3.1.2 Learning-Based Methods for Occlusion Handling

A learning-based technique based on Deep Q-Learning is used to solve a POMDP that depicts static occlusions at junctions [34]. The authors employ a discretized state space to color label occluded regions in the occupancy map using a ray tracing technique. The agent then determines the best strategy for dealing with the hidden players in the obscured area. Deep Reinforcement Learning, on the other hand, struggles to generalize to new situations and makes approximation errors. Deep reinforcement learning outperformed rule-based systems in terms of success rate and efficiency for a certain intersection (number of lanes) and a specific action (right turn, left turn, or ward passing). When applied to intersections of a different type these models have reduced success rate as they are trained with a specific type of intersection. Isele et al. [32] also demonstrates that consecutive training with various junctions can result in catastrophic forgetting. However, [33] tries to use reinforcement learning on occluded intersections as well. However, the high collision rate demonstrates that this technique is unable to properly comprehend the circumstance and consequently makes incorrect conclusions. Therefore, in order to handle occlusions, it is required to use an imitation learning-based behavior planner.

3.2 Machine Learning-Based Behavior Planning

Planning in general refers to the process of making deliberate decisions in order to fulfill the higher-order goals of an autonomous system, such as getting the vehicle from a starting point to a destination point while avoiding obstacles and optimizing over designed heuristics [56]. There are two types of intelligent planning methods, statistical and Symbolic AI planning. The development of an effective AI system with a layer of reasoning, logic, and learning skills is a fundamental challenge in any autonomous systems. However, most AI systems today have either learning or reasoning skills. Now, a symbolic method can do well in reasoning, provide explanations, and manipulate large data structures. Some of the literature works related to symbolic planning in discussed in [18, 19, 37]. However, symbolic AIL planning has significant problems anchoring its symbols in the perceptual world. On the other hand, the statistical planning has issues with explainability.

A deep look into the works related to imitation learning-based planning that are relevant to this thesis topic are discussed as follows. The ALVINN system [58] was the first imitation learning algorithm used in autonomous driving. It used a three layer neural network to perform road following task based on front camera images [10]. Later, NVIDIA created a end-to-end driving system employing deep convolutional neural networks [5, 6], which was intended to perform a very good lane following in a more challenging circumstances where there are no lane markings recognized. Deep neural networks were also trained to anticipate the high-level control output from camera images and their open loop performance was evaluated. Generative Adversarial Imitation Learning (GAIL) proposed by Kuefler et al. [42] used a simple affordance style features as inputs. Hecker et al. [24] in 2018 published a research that reveals a steering and speed prediction model based on 360-degree camera inputs and a desired route planner. Xu et al. [71] trained a deep driving policy using an FCN-LSTM architecture with a segmentation mask. Despite the fact that [71] had good prediction performance for complicated urban scenarios, it failed to enable closed loop evaluation in real or simulated environments. The CARLA simulator [14] developed in 2019 has opened a new horizon to the autonomous driving systems. CARLA has made the autonomous driving systems to be trained and

tested in a realistic 3D urban driving simulation environment. Based on CARLA, a conditional imitation learning technique is used in [11] to develop an end-to-end deep policy which can obey the high-level commands like go straight and turn left. Additionally, Sauer et al. [61] predicts numerous affordances based on sensor inputs to operate an autonomous car in the CARLA open source simulation environment.

All of the approaches discussed above use front camera images as inputs. The major hindrance with those approaches are the complexity of direct visual information. The usage of a bird's-eye view can help to simplify visual information while still providing valuable information for driving. Uber [12, 13] employed a rasterized image with map and object information as the input and built a convolutional neural network to forecast the vehicle's future route. Waymo [2] employed a comparable mid-to-mid representation similar to Mueller et al. [52] and built a deep learning model that could drive a vehicle through a variety of urban environments when combined with perception and control modules. A complete survey for deep reinforcement and imitation learning based planning methods are discussed in detail in [74].

3.3 Behavior Planning System Architecture

An overall system architecture of neural network based self-driving system is presented in this section.

The end-to-end planning system described by Waymo [2] is shown in figure 3.1. A neural network based planner is used to predict the waypoints which aids in driving the autonomous vehicle system. At each time step, an update about the environment and ego's state is obtained from the simulation environment or from perception system in real vehicle. The intended path or route is acquired from the router. The information from the environment is rendered into the input grid images described in figure 3.4. They are given to the neural network based policy model which then outputs a future trajectory in the form of waypoints. The predicted waypoints are directly fed to the controls optimizer to extract the low-level control signals that eventually drives the vehicle either in simulation or real world environment. Additionally, Chen et al. [10] utilised the top-down input representation and provided an improved framework for handling complex urban scenarios using imitation learning and a safe set theory based safety controller.

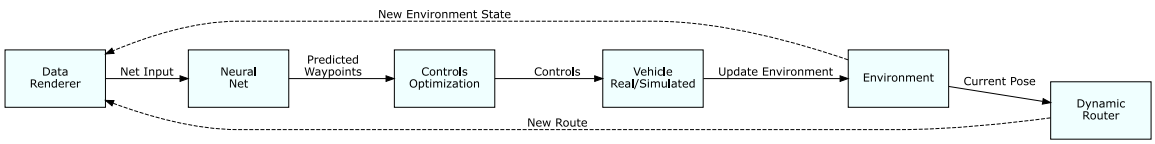


Figure 3.1: Software Architecture for End-to-End Planning System [2].

3.4 Framework Overview

The details about the in-house planning system at Bosch is described in this section.

The baseline AI planning system which employs imitation learning inspired from ChauffeurNet [2] is explained in this section. The complete workflow of the framework along with training and evaluation pipeline are described in figure 3.2. The components of the pipeline are explained in detail in the following sections.

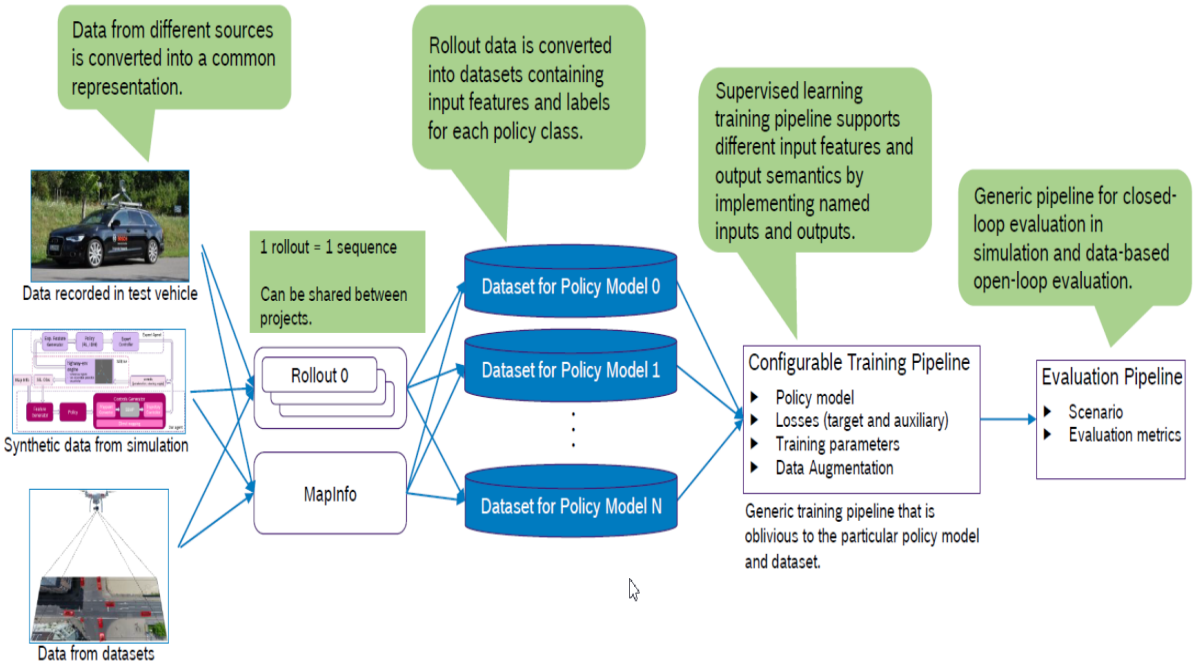


Figure 3.2: Behavior Planning Pipeline [41].

From the figure it can be seen that the data some different sources are converted into a common representation called as rollout. The data can be collected from different sources like test vehicle, simulation environment or using open source datasets. The rollouts generally comprises of one scenario with a non-zero duration. It contains the output of the perception system in the form of object lists including ego state. Each rollout is a sequence of state-action pairs. The rollouts are then converted into datasets with input features and labels for each policy class. Next the dataset is fed into a supervised learning training pipeline which supports different input features and output semantics by implementing named inputs and outputs. Finally, the trained policy model is evaluated using closed loop evaluation in simulation.

3.4.1 Simulation Environment

In practice, the neural network-based behavior planning model deployed in the real vehicle requires real-time measurement data from the vehicle perception systems to make intelligent decisions. However, as imitation learning approach relies on training data, it is useful to have a simulation environment where the road environment and vehicle movement could be simulated for the purpose of research and development. By means of such simulation environment enough training data could be simulated and the same environment could be used for closed-loop evaluation of the trained model. In this thesis, an open-source Highway-env [43] simulation environment is used. Highway-env is

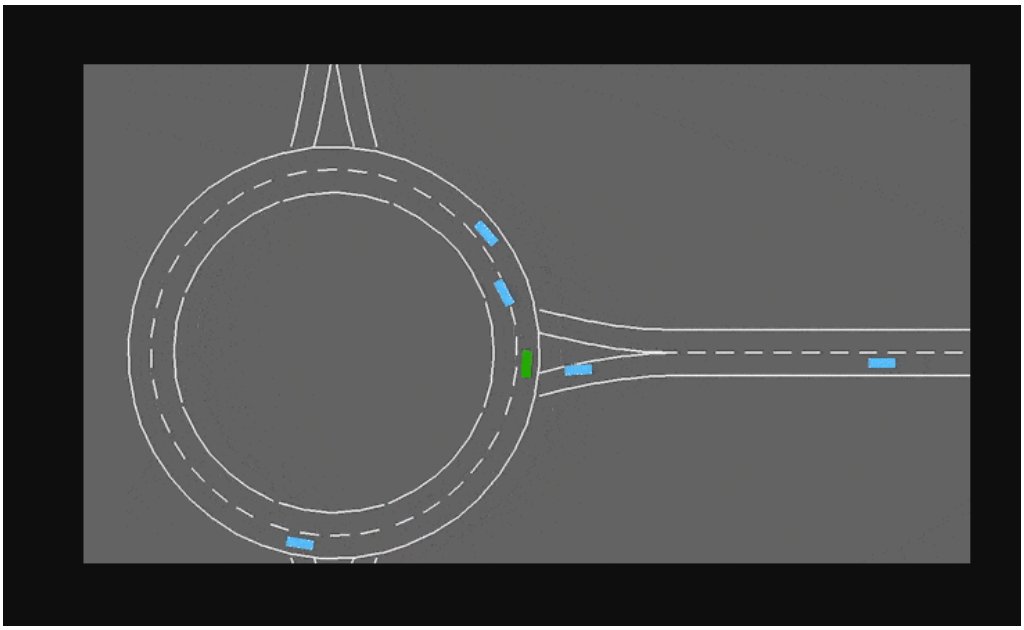


Figure 3.3: Highway-env Roundabout Environment [44].

a simulation environment for autonomous driving systems. It supports configuration of desired environment type. In our case, the environment consists of a simulated autonomous vehicles where driving in a roundabout is defined. Various in-house maps of roundabout environment are created for different scenarios. Highway-env supports to perform driving simulations using the in-house maps. The agent model consists of an autonomous vehicle which interacts with the simulated environment through a variety of actions. Other vehicles that are not controlled by the behavior planner are controlled by the IDM. The roundabout topology of the simulation environment along with simulated vehicles are shown in figure 3.3. This figure consists of a three way roundabout as an example. But the roundabout topology used for this thesis work is a four way roundabout.

The input for the highway-env simulator is the control commands which is the given by controllers. These controls are the waypoints in the environment. The training data from the simulation environment, consists of the objects and map data also known as features. These features serve as the input to the ego. The ego vehicle in turn drives through the environment and generates waypoints in return. The resulting features in conjunction with the waypoints are served as a data record. In addition to generation of training data this environment is also used for training and evaluation of the policy models. The simulated training data is stored as a rollout.

Consequently, it is worthwhile discussing the scenarios that can be generated from the highway-env simulation environment. The possible scenarios are empty, general, yielding, yielding preceding, stopping, drive-off, and off-road parked cars. These can be configured in the environment using a scenario generator. The scenario generator specifies the initial speed, target speed, destination location, initial position of ego, target speed as well as the positions of other vehicles in the environment. As the name states, the empty scenario has no traffic in the environment. In yielding, the vehicle has to slow down in response to a traffic. Yielding preceding is an extension of yielding where ego has a preceding vehicle and has to yielding in response to the traffic near the roundabout junction. Whereas, stopping and drive-off corresponding to ego's behavior of stopping due to an

ahead vehicle and driving off once the path is clear. For occlusion handling, yielding scenario is very important where the behavior planner should make more careful decision to obtain a broader visibility of the environment. It is discussed in detail in the chapter 6.

3.4.2 Rule-based Intelligent Driving Model

The expert model used in the behavior planning framework is the Intelligent Driving Model (IDM) introduced by Treiber et al. [69] is a rule-based model used for human driving behavior. As it combines realistic simulation properties with a simple formulation and calculation, IDM has been used in various scientific projects relating to driver modeling, driving assistance systems, and collective traffic simulations It has a small set of parameters that are naturally linked to driver characteristics. It is in turn a deterministic model and is constantly dependent on its input variables. IDM strikes a compromise between the desire to drive as quick as possible if there were no vehicle in front, and the requirement to maintain safe distance from the preceding vehicle. Moreover, IDM by default ensures to avoid collision [15]. Consider s and v are the longitudinal position and velocity of the ego vehicle. As illustrated in [4], the IDM takes the vehicle's current speed v at a time t, the relative velocity Δv with leading vehicle and the distance $d = |s - s_1|$ between the ego and the front vehicle. The model calculates a longitudinal acceleration a_{IDM} as an output as shown in equation 3.1.

$$a_{IDM} = a\left[1 - \left(\frac{v}{v_0}\right)^{\delta} - \left(\frac{d_{des}}{d}\right)^{2}\right] \tag{3.1}$$

where the desired distance to the preceding vehicle is given in equation 3.2,

$$d_{\text{des}} = d_0 + v \cdot T - \frac{v \cdot \Delta v}{2\sqrt{a \cdot b}} \tag{3.2}$$

The IDM has parameters v_0, d_0, T, a and b that determine the acceleration output based on The model has several parameters that determine the acceleration output based on the information about the environment. Here, v_0 refers to the desired free cruising speed, d_0 refers to the minimum distance between the ego and preceding vehicle, T refers to the minimum time gap between ego and front vehicle, and a and b refers to the maximal limits on the acceleration and comfortable deceleration, respectively. By arbitrarily setting the parameter values, a collision-free motion of a vehicle can be simulated. But this cannot guarantee a realistic driving behavior. Though IDM does not explicitly consider risks, some aspects of criticality are handled implicitly by the terms and parameters related to the time gap T and the deceleration factor b [4].

3.4.3 Input and Output Representation

Normally, the planning system takes raw sensor data as input and gives the direct control commands such as braking and steering angle as outputs. As mentioned in [10] the raw sensor data contains very high-dimensional information which is affected by various factors like, textures and looks of roads and objects, and also with weather conditions and sunlight. The dataset must contain enough

information for each feature of the sensor data, such as texture, weather, light condition, and object appearance, in order to attain generalization of the learned strategy. Also the variations in vehicle dynamics has major impact on the direct control output. If the vehicle dynamics change, a new policy has to be trained. Hence, the direct mapping of sensor data to the control commands are extremely time consuming and also difficult to obtain generalization.

Therefore, a well-suited input representation is crucial for applying imitation learning technique in the behavior planner successfully. Thus a top-down input representation is adopted inspired from [2]. Also Chen et al. [39] used the similar representation of rasterized images for learning the deep learning-based planner.

The rasterized image contains information about the map and objects in a top-down representation that can be fed to the network to output a drivable trajectory point. In such representation, for any time t the ego vehicle's can be represented in a top-down coordinate system. The advantage of this coordinate system is that the ego's position at a current time t is always fixed within the image frame. The inputs consists of grid of images of size W × H pixels with a resolution Φ rendered into the image coordinate system in a top-down manner.

The top-down input representation is composed of six parts. Out of which four are illustrated in figure 3.4. The details are described as follows:

- **High-definition (HD) Map:** The HD map contains the information about the road topology. It gives information about borderlines which is shown in figure 3.4(a). Additionally, the map also renders other features such as road surface, solid and broken lanes markings.
- **Routing:** The routing gives information about the path provided by the route planner. It is the path in which ego vehicle is traverses through the environment. The path is represented by a set of waypoints as shown in figure 3.4(b).
- **History of Detected Objects:** The surrounding objects that are detected in the past are rendered as blue boxes. Three time steps form the past are rendered into different frames.
- **Past Ego States:** The past positions of the ego vehicle are rendered into images. It contains three frames of past time steps.

The box representation of the objects helps in easier generation of input data either from simulation environment or from perception system. This enables testing and validation of the modes in a closed-loop simulations before being deployed into the real vehicle as stated by [2]. The rasterization is an image of size H × W × C. Rasterization is an encoding methodology that renders the scene into a sequence of image frames [67].

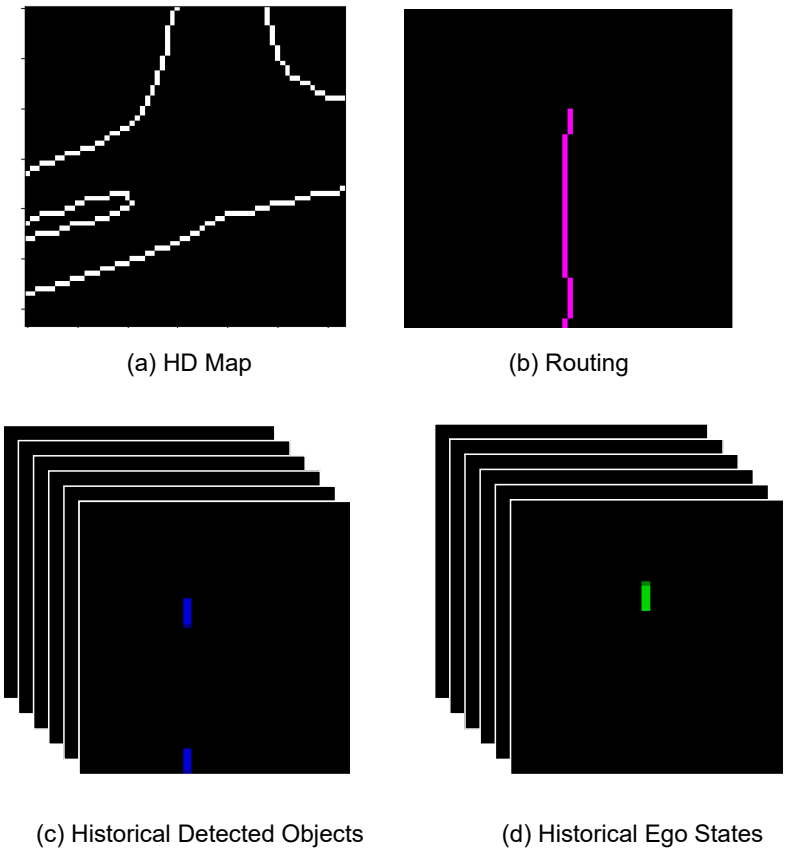


Figure 3.4: Top-down Representation of the Model Inputs.

The end-to-end approaches directly derives low-level controls as output to drive the vehicle. Steering angle and acceleration are some of the examples of low-level controls [54]. Although, with this representation the model has a drawback of not being modular and cannot interface with any system other than the one in which it is trained on. A waypoint output representation is adopted to produce mid level outputs. These can be adjusted individually in each vehicle by the control variables. Therefore, waypoint out representation is used in the state-of-the-art model. The waypoints are depicted in the architecture diagram in figure 3.5.

3.4.4 Multi-task Policy Model Architecture

The behavior planning model is composed of three parts. The multi-task network used in this thesis consists of CNN-based backbone model, a waypoint head and a prediction head. Among them, waypoint head is the primary task of the model and the prediction head is the auxiliary task. The backbone model is based on ResNet34 [22] and it takes the rasterized input data as represented in figure 3.4 to create a latent representation of features. These latent features are consumed by The features are then consumed by the waypoint head and prediction head. The waypoint head which is made of Multi-layer Perceptron (MLP) outputs a set of waypoints in x and y coordinates that is used to control the ego vehicle. A prediction is also co-trained using same latent features from the output of backbone. Prediction head consists of convolutional network suitable enough to

predict the future poses of dynamic objects. Here, prediction is used as an auxiliary task that helps in improving the generalization of the main planning task. This idea is inspired form ChauffeurNet [2]. The prediction task is only used during training of the behavior planning model and it has no effect at the test time. The details of each components of the network is illustrated in figure 3.5. This multi-task model is capable of behavior planning and shows the potential of imitation learning-based behavior planners.

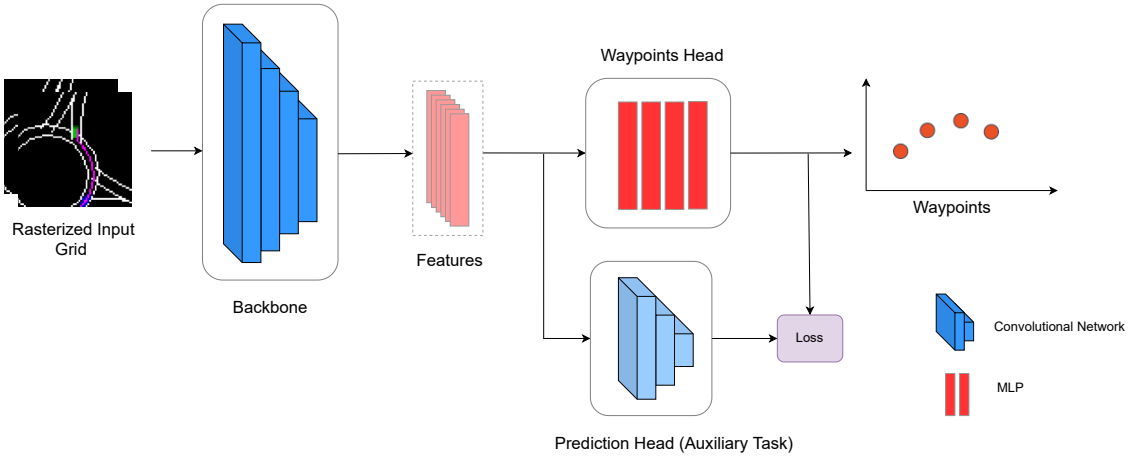


Figure 3.5: Illustration of Behavior Planning Network Architecture.

As shown in figure 3.5, a set of rasterized images are fed into the CNN-based backbone model. ResNet34 is the backbone in the baseline model and it is truncated after layer 1 to obtain a feature dimension compatible with the waypoint and prediction heads. The waypoints head outputs a list of waypoints where each point represents x and y positions in the ego coordinate system. The prediction head uses 1 × 1 convolutions with stride of 1 [27]. Predominantly ReLU [53] is used as the activation function between the hidden layers of the CNN-based networks. The policy network is trained using supervised learning technique with the help of a special combination of loss functions. The details of loss functions are discussed in section 3.4.5.

3.4.5 Loss Functions

In order to train the model to imitate the expert behavior, a specific loss function as specified in equation 3.3 is to be employed. They are discussed in detail in this section. The total imitation loss function is the combination of all the losses discussed as follows.

$$L_{total} = 0.5 \cdot L_{wl1} + 0.5 \cdot L_{path} + L_{wl2} + 10 \cdot L_C + 0.2 \cdot L_R \tag{3.3}$$

Waypoint L1 Loss

This is a robust loss function with fast convergence, that combines the properties of the absolute and the quadratic loss. A Huber loss is a piece-wise function used to combine the quadratic and absolute losses where they work the best. Waypoint L1 loss is calculated using huber loss function [20] with $\delta = 1$ and k time steps. Equation 3.4 describes the loss function used to computed waypoint L1 loss smooth.

$$L_{wl1} = \begin{cases} \frac{1}{2}a^2 & \text{for } |a| \leq \delta \\ \delta|a| - \frac{1}{2}\delta^2 & \text{otherwise} \end{cases} \tag{3.4}$$

Where, a is the absolute value of waypoint output p_k and the waypoint ground-truth labels p_k^{gt}.

Path L1 Loss

The path loss function is used to calculate the L1-norm between waypoints and their corresponding ground-truth waypoints and are matched to the path in Euclidean space. Consider x_k is the waypoint output, y_k is the ground-truth waypoints and p_k is the path polyline points. $point_k$ are calculated by taking orthogonal projection of waypoints onto the given path. Path L1 loss is calculated according to function specified in equation 3.5.

$$L_{path} = {}_{i=1}^{n} |x_k - point_k| \tag{3.5}$$

Waypoint Derivation Discount Loss

Waypoint derivation discount L_{wl2} loss is calculated using L2-norm of waypoint based on absolute derivation. It is mainly intended to calculate absolute velocity loss. A first order derivative of L2-norm is obtained and the loss value is calculated between output waypoints and their corresponding ground-truth.

Auxiliary Losses

By co-training a prediction head to predict the future positions of the dynamic agents by sharing the same feature representation F used by waypoint head. It is more likely to induce the backbone network to learn better features that are suited to enhance the waypoint prediction output. Losses calculated from prediction head are discussed as follows.

1. **Prediction Classification Loss**

   This is an object detection loss for the anchors on a grid image [27]. It is mainly used for classification only and hence it adds a cross-entropy loss to the predicted output $x^{`}$ to its ground-truth label $y^{`}$. Based on the research work from Lin et al. [47], a modulating factor

can be added to the cross-entropy loss to handle the imbalances during the training process. The focal loss is depicted in the equation 3.6. The prediction classification loss is calculated based on modified cross-entropy loss and it is named as L_C.

$$\mathrm{L_C}\left(p_\mathrm{t}\right) = -\left(1 - p_\mathrm{t}\right)^{\gamma} \log\left(p_\mathrm{t}\right) \tag{3.6}$$

2. **Prediction Regression Loss**

   For a regression task, the prediction loss [27] for the anchors on the grid images are calculated by absolute difference between the prediction output and the ground-truth labels and finally averaged over the ground-truth points. The prediction regression loss is termed as L_R.

3.4.6 Evaluation Pipeline

There are wide range of methods for evaluating the policy models in the literature. The evaluation can take place in different steps. The machine learning methods are predominantly evaluated with learning curves. In addition, the models can be evaluated on the basis of validation data. The evaluation pipeline illustrated in figure 3.2 is described here. As already mentioned, a closed-loop evaluation is employed using the simulation environment. The evaluation pipeline consists of various scenarios 3.4.1 and quantitative metrics specific to the roundabout driving use case.

Kothari et al. [40] presented an extensive closed-loop evaluation protocol for testing the learned policies. Within this pipeline, the user can access all the important simulation artifacts such as waypoints, maps, agents and log reply of ego vehicle. Counterfactual policy evaluation was introduced by Hart and Knoll [21] to review the generalization of the trained policy. Here, reflections on events that have not occurred are executed during evaluation. As a result different scenarios are developed and the generalization of the learned model can be ensured. The ego vehicle and the other road users are always initialized at the same position and at the same initial speed. What is changed is the driver model of one of the other Vehicles. For this purpose, the surrounding vehicles are assigned with different accelerations. The ideas of these previous are incorporated into the evaluation pipeline used in this thesis.

To evaluate the driving manoeuvres, different metrics are presented in the literature. Also in order to support en effective closed-loop evaluation suitable metrics are necessary based on the use case. Kothari et al. [40] proposed two sets of metrics: distance-based metrics specific to imitation learning and safety specific metrics to capture various collisions between the ego and the surrounding vehicles. Chen et al. [10] in 2019 used two-metrics for closed-loop evaluation. One is success rate and other measure the violations. By defining starting and ending points the success rate of the model is measured and can be determined across all scenarios. Whereas, Isele et al. [33] utilized four metrics, percentage of success, percentage of collision of ego, average time taken to reach the destination and average braking time for evaluating the Deep Q Network to ensure that it navigates through the occluded intersections safely. Inspired from the above mentioned literature work, ratio of rollouts ending in success and ratio of rollouts ending in collision are used as evaluation metrics in this thesis work. Additionally there are other metrics that are employed specific to the roundabout driving scenario. The appropriate metrics used for experimental evaluation are defined in detail in section 6.2.

3.5 Specific Research Questions

Based on the analysis of the state-of-the-art behavior planning framework, the possible research questions are formulated which needs to be explored and analysed in this thesis work. They are as follows:

RQ1 How to design occlusions in the roundabout environment?

RQ2 How can the baseline IDM be enhanced to handle the occlusions present in the environment?

RQ3 How to evaluate the performance of the extended IDM while navigating through the occluded roundabout?

RQ4 How to restore the performance of the policy model when supplied with a limited visibility dataset?

4 Methodology

The methodology used to handle occlusions in a roundabout environment with an imitation learning-based behavior planning is described in this chapter. The algorithm used to include occlusions in the roundabout, the creation of expert demonstrations, and ideas for feeding the additional information to the input data followed by various backbone model architectures are discussed. This chapter also includes the details on policy training and the corresponding training results. Finally a brief about the optimization algorithm used for tuning the hyperparameters are also discussed.

In this thesis, we assume that we already have a functioning perception module to process the raw sensor data. For example, we can use a localization system to estimate ego vehicle pose, an object detection system to detect the bounding box, position and heading of surrounding objects, and a traffic light detector to tell the states of traffic lights. Furthermore, we have access to the High-Definition (HD) map data for the area that the ego vehicle is operating, as well as the routing information which guides the ego vehicle to a specified goal position. This is a valid assumption because, they are not difficult to obtain with existing autonomous driving technology.

4.1 Baseline Model

The multi-task model detailed in section 3.4.4, is used as a baseline model for handling occlusions in behavior planner. Then potential enhancements are made on top of it. A static priority rule-based IDM model is used as expert model. IDM used to generate training dataset for the policy model from the highway-env simulation environment. The policy model as already stated consists of CNN backbone made of ResNet34 [22], MLP-based waypoint head to output future waypoints. The policy model is also equipped with a CNN-based prediction head [27] as an auxiliary task to enhance the waypoint prediction task. This baseline model was developed to implement the task of an automatic waypoint prediction for behavior planning. Figure 3.5 shows the selected model architecture. The input is processed by a backbone model and then passed to a set of fully connected layers and finally outputs the path points. Grid images as shown in figure 3.4 are used as input features. They contain the following information: Current ego position, history of ego position, current position of other objects, history of other objects, lane markings. The images have a resolution of 64 x 64 pixels. At the specified resolution of 1.0 m/pixel, this corresponds to a viewing range of approximately 64 m in the direction of travel and in the lateral direction. The ego vehicle is positioned at a distance of 21 pixels, i.e. 21 m, from the top edge of the image, with the direction of travel pointing downwards. This gives the vehicle a visual range of 43 m to the front. The history of the last two ego positions and the last two positions of other objects is scanned every 3 secs. This results in 10 binary images. Therefore each sample consists of an image with 10 binary channels. The corresponding labels are also created for each sample. These consist of the next 15 waypoints that the ego vehicle drives through. They are also specified every 3 secs as the input features. In other words, a trajectory is output for every 3 secs.

4.2 Proposed Approach

The methodology proposed to solve the risk in the roundabout environment due to the presence of occlusions is outlined in this section. Firstly, as described in section 3.4.1, a simulation environment is used for generating datasets for training the behavior planner. Hence, an occlusion region is included in the environment using the object filter algorithm. Once, the environment poses occluded zones, the existing baseline IDM has to be extended to handle the limited visibility of the occlusion environment. This is essential to ensure safe and comfortable driving in the occluded regions. The new extensions to the IDM model are evaluated by observing a potential difference in its driving behavior in an occluded environment. For this purpose, a new KPI metric is defined to perform a quantitative comparison of the extended IDM with its baseline model.

As a next step, the demonstration from extended IDM serves as a new data to training the policy model. In this case, the policy learns to imitate the driving behavior form expert demonstrations, but it may not be able to differentiate the occluded zones in the environment. Hence, it is necessary to feed the visibility region information to the CNN-backbone network of the policy model. Adding an additional channel to the input top-down grid images could be a viable solution to solve this case. A clear indication of the occluded and visible regions of the environment through a binary image can leverage the policy model to handle the invisible zones and provide a desired driving behavior. In the course of action, various CNN-based backbone model architectures used to train the policy, to achieve the desired results. Additionally, BOHP [16] optimization algorithm is used for tuning the hyperparameters for the policy training. Finally, each deep neural network policy model is evaluated by closed-loop simulation environment with and without occlusion zones under several driving scenarios and a comparative study is made and elaborated.

4.3 Occlusion Algorithm

Occlusions are common in urban traffic situations, particularly in the roundabout. The two types of occlusions as stated in [57] are considered in this work they are as follows:

- Object occlusion: Static or dynamic objects in the street scene are partially or entirely occluded. This makes it difficult to recognize and track such objects and poses a potential risk when driving.
- Free space occlusion: Some occluder in the surroundings environment blocks some region. As it is highly unknown if the occluded region is clear of impediments and suitable to drive, this is challenging scenario for prediction and planning.

For the purpose of occlusion handling in roundabout, both of the above types are considered. An algorithm is developed to include occlusion regions in the simulation environment as stated in section 4.2. A HD map data of roundabout environment is used to define the occlusion zones. Logically, the area inside the roundabout near the intersection could be an occluded zone for any ego vehicle approaching the roundabout. In terms of this assumption, a fixed area in the map is defined as occluded zone as depicted in figure 4.1 which has high probability of hindering the visibility of the vehicle and causes risk. The figure shows the regions of fixed occluded zones in a 4-way roundabout map. There are four occlusion zones O_k near the junctions of the roundabout. A four

way roundabout map is predominantly considered throughout this thesis. The path through which the ego travels via the roundabout is showed in the figure according to their map coordinates. The orange region is the starting position of the ego vehicle. The marked rectangles in the roundabout path are the fixed occlusion zones. As a sample, based on the current ego position (orange), the more relevant occlusion is the red color rectangle.

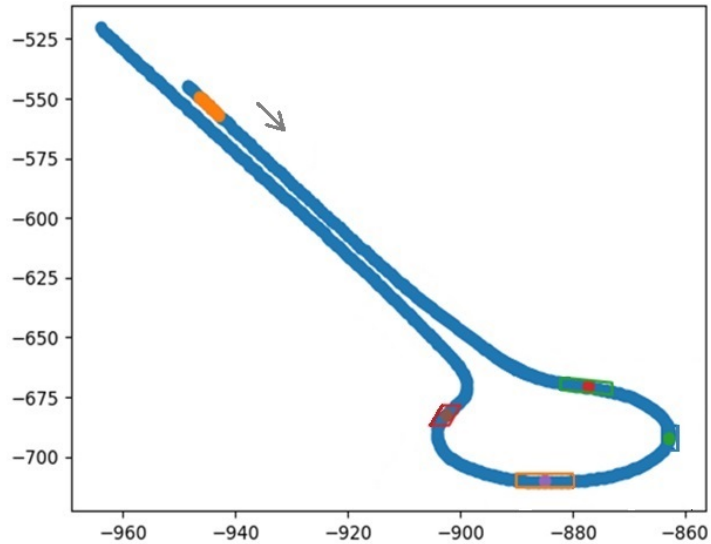


Figure 4.1: Fixed Occlusion Zones in the Roundabout.

Having known the location coordinates of the occlusion zone, an object filter algorithm is implemented to remove all the objects i.e., the vehicles that enter inside the occluded zone of the environment. In such a way, an occlusion is being included to the roundabout environment, where the ego vehicle will have access to only limited object lists. Once, the vehicle is out of the occlusion zone, it will suddenly be visible to the ego vehicle, and it will be hard for the ego to handle this situation effectively and quickly to avoid risks. The flow of the algorithm implementation is described is described in the flow chart shown in figure 4.3.

4.3.1 Dynamic Occlusion Zones

The simulation environment using in this work has only single ego vehicle, at any point of time t, hence it is viable to have only one of the four occluded zones at any point of time. The main reason is that considering the static driving rules, only a single zone near the ego's path has a potential risk of occluding the vision of the ego. Hence, the fixed zones are made to be dynamically active based on the ego's current position. To realize this, another set of zones called as trigger zones T_k are defined in the pathway towards the roundabout junction. Figure 4.2 depicts the set of trigger zones in green color defined on the four way roundabout HD map. Considering the trigger zone information, the object filter algorithm is modified to activate a single occlusion zone $i \in \mathrm{O}_k$ if the current ego position is inside the corresponding trigger zone $j \in \mathrm{T}_k$. In addition, to avoid ego

vehicle being stuck inside the trigger zone, it is important to fix the trigger zones at a reasonable distance from the roundabout junction. In this work, trigger zones are fixed at a distance 5 meters away which is one vehicle size in the environment.

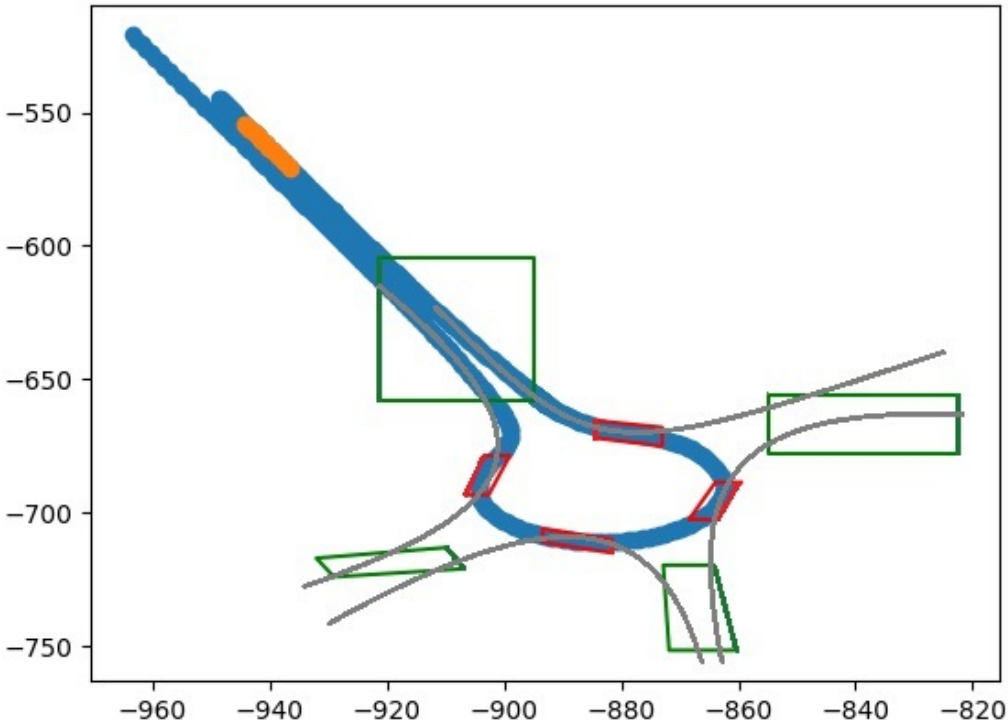


Figure 4.2: Roundabout with Trigger and Occlusions Zones.

Finally, the complete object filter algorithm used to induce occluded region in the highway-env simulation environment is depicted in the flow chart shown in figure 4.3. Accordingly, the rollouts R_i, are the sequence of actions generated per time steps in the simulation environment. Each rollout is a combination of series of frames f_i. And each frame contains the list of objects at a particular time instance. The rollouts also has more information such as ego ID, current ego speed, current position of ego and other vehicles etc. As per the flow chart in figure 4.3, the information about objects lists are acquired from the frames of the rollout. If the object is an ego vehicle it is retained. Otherwise, the object in the frame is retained only if it is not inside the occlusion zone. In case if the object is present inside the fixed occlusion zone, it is filtered. And hence this algorithm is termed as an object filter algorithm.

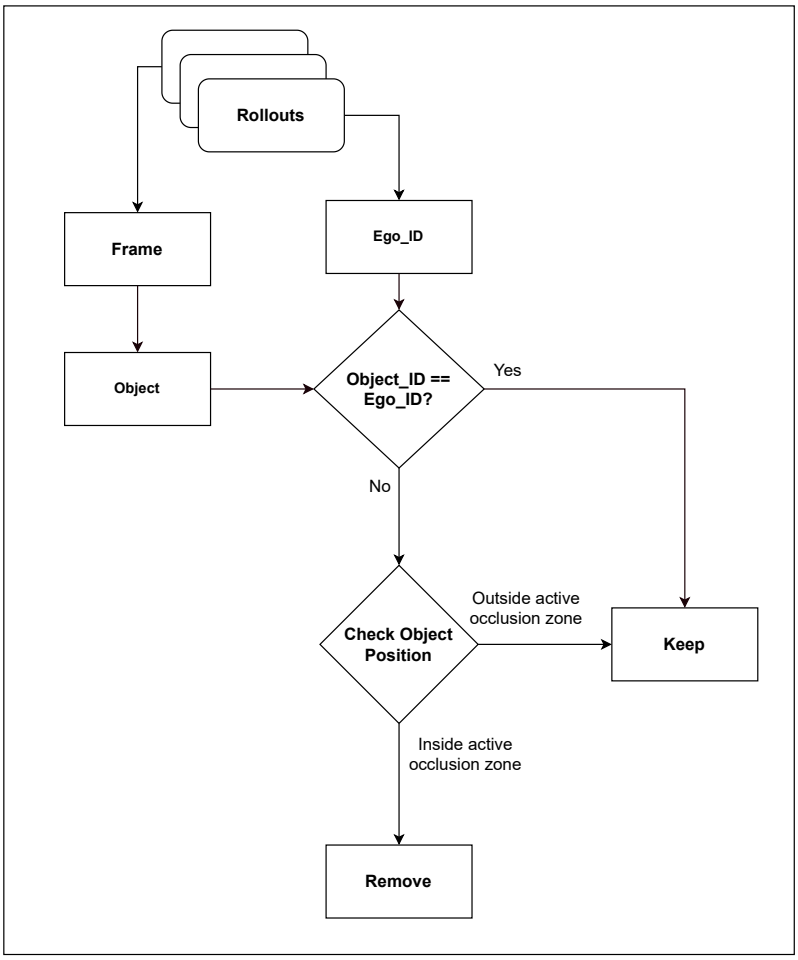


Figure 4.3: Occlusion Algorithm.

4.4 Expert Enhancement

The IDM described in section 3.4.2, is considered as the baseline IDM which is capable of handling the default environment with full visibility. When it gets exposed to occlusion environment it suffers from limited visibility. Minor tweaks are to be applied to the IDM to provide additional logic for handling occluded roundabouts. This is essential to obtain reasonable results without any collisions in case of situations with limited visibility. The list of enhancements added to the IDM model are:

- Spawning a vehicle inside the active occlusion zones featuring with a target velocity meanwhile ensuring minimum distance between the vehicles.
- Enable the ability of IDM to hallucinate about the hidden spawned vehicles.
- Yielding logic is extended for the hallucinated vehicles as well to respect the static priority rules .
- Crawling (or creeping) behavior is induced to handle when the target speed gets zero inside the trigger zone.

The above described extensions are applied to IDM to be usable for the behavior planning in the occluded roundabout scenarios and this enhanced expert is called as extended IDM. A complete flow of how the existing baseline IDM is used to adapt to the proposed extensions are described in figure 4.4.

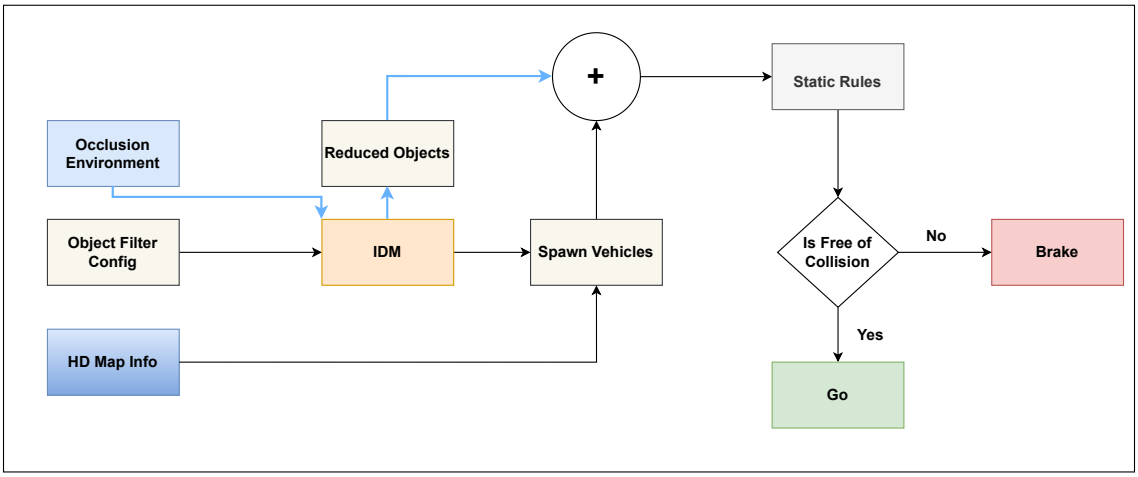


Figure 4.4: IDM Extension Workflow.

Accordingly, the IDM gets information about the HP map and object filter config as its inputs. It is able to calculate the reduced set of objects from the occlusion environment. Meanwhile, by using the map and object filter configurations, IDM spawns vehicles inside the active occlusion zones alone. The spawn is done based on the parameters of the frenet coordinate system with a desired speed and position. By doing do, IDM knows the positions of the spawned vehicles that are inside the hidden areas. It enables the IDM to hallucinate about their presence. Finally, as per the baseline IDM strategy, the new IDM also check for static priority rules. But as a change the new IDM uses both the hallucinated objects and reduced objects lists and check for the regulatory rules. In general, the static (or regulatory rules) checks the lane position and speed of the vehicles and verifies if there is a possibility of collision. If a collision possibility is found, it makes the ego (i.e, IDM in this case) to yield near the junction of the roundabout. The same rules are illustrated in the flowchart shown in figure 4.4. If no collision is found to happen, then the IDM is free to move forward. In this extended IDM, the hallucinated and reduced objects enables the IDM to check of rules and drive more carefully than the baseline IDM, which is the desired behavior.

4.4.1 Creeping Action

While testing the performance of the Extended IDM, it is observed that extended IDM is able to hallucinate about a hidden vehicle inside the occlusion zone base on algorithm in figure 4.4. There arises a situation during yielding scenario, when IDM is travelling through the trigger zone, it is able to see the hallucinated vehicle and react to it by yielding. Such yielding action results in a speed of 0, thus IDM gets stuck inside the trigger and cannot able to drive forward.

To overcome this issue, a creeping action is induced to the IDM model. For this purpose, a priority order is set for checking the static priority rules. Once the vehicles are spawned and added to the hallucinated vehicles list, a regulatory rules check is first made for the hallucinated vehicles alone. In this case, an yielding action of the IDM is monitored continuously and whenever the speed gets zero during yielding for the hallucinated vehicles, an external push is given to extended IDM by resetting the speed to a crawling velocity. At the end, the same regulatory rules are checked for reduced objects list which comprises of visible vehicles of the environment to ensure a safe and rule-obeying behavior.

4.5 KPI Definition

The baseline IDM when exposed to an occlusion environment, has visibility only to a reduced set of objects and this version of IDM is called as limited visibility IDM. There expected a collision with suddenly appearing vehicles at the junction. Surprisingly, the baseline IDM is good enough to handle such limited visibility without crashing to any vehicle by following the static priority rules.

As it did not crash as expected, there arises a need to define a new comfort metric based on the heuristics. In this case, the velocity profile is taken into consideration. Maximum deceleration during braking (m/s^2) KPI is defined based on the rate of braking.

A detailed analysis of experts: baseline, limited visibility and extended IDM, has resulted in an observation where the braking behavior of the models are different. A study is conducted to understand the braking behavior by deeply analysing the velocity profiles of the three experts. Figure 4.5 shows the velocity plot of the experts for each timestamp. In general, there are four regions in the velocity graph: initial acceleration, braking, crawling, post acceleration. It is observed that all the three models starts with a initial velocity and accelerates during initial driving. When the IDM is near to the roundabout junction, it yields by braking (or deceleration) to adhere to the priority rules. After yielding, IDM waits until the traffic is clear, it is reflected in the curve with a constant velocity values in the crawling region. Once, the roundabout is free to drive, IDM gets priority and it drives off. The drive off is visualised by shooting velocity values in the post acceleration phase.

For analysis, the braking region of the velocity profile is considered as the area of interest. From the graph shown in figure 4.5 it is evident that the limited visibility IDM has a hard braking behavior. It starts braking later than other two models and also it brakes sharp to a velocity close to zero. In the meantime, baseline IDM starts braking little earlier than limited visibility model and this also exhibits a steep curve in braking region causing discomfort. On the other hand, the extended IDM showcases a comfortable driving behavior by braking very early and slowly. The slope of extended IDM is not very steep, additionally the IDM poses a minimal crawling velocity and hence it is moving steadily towards the junction. Therefore, the above analysis paves way for a definition of new KPI metric to evaluate the IDM models.

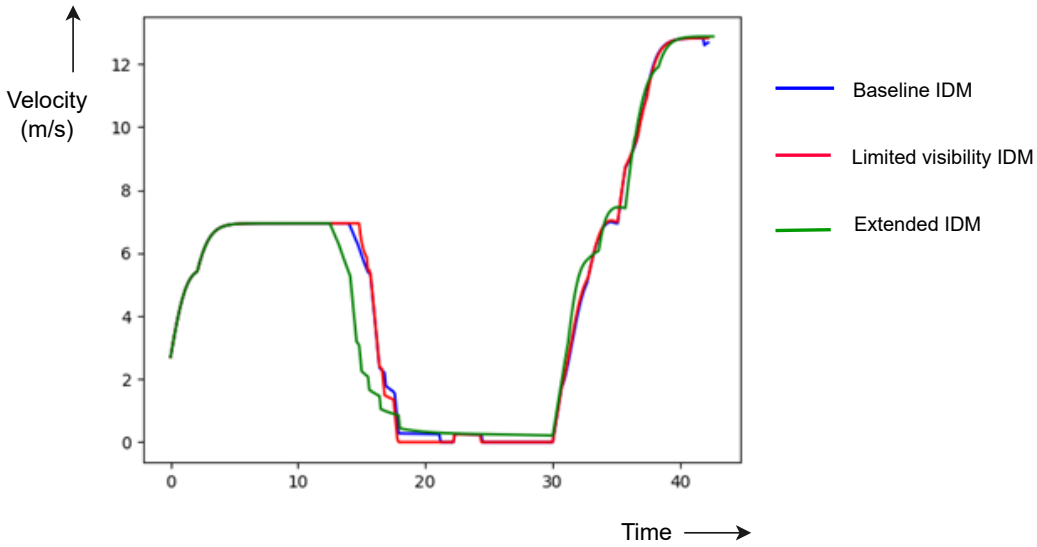


Figure 4.5: Velocity Profile of IDM Models.

4.5.1 Maximum Deceleration during Braking

Maximum deceleration during braking (m/s^2) KPI is defined based on the rate of braking as discussed in previous section. The algorithm is described in pseudocode below. Two thresholds are used in this algorithm namely velocity before stopping of 0.24 (m/s) and minimum velocity drop of 6 (m/s).

```
for i in timesteps:
for j in i:
if velocity[i] - velcoity[j] > minVelocityDrop_threshold:
braking_acceleration = (velocity[i] - velocity[j])/(j-i)
if(length(braking acceleration) >0):
maximum_Deceleration = max(braking_acceleration)
```

The figure 4.6 shows the evaluation results of three IDM experts corresponding to the new KPI. Here, the extended IDM model is referred as *full-idm* in the figure 4.6. Where, the lines are the confidence interval of the result and their mean value is indicated with a point. It can be verified that the extended IDM (or full-idm) has lower mean value and confidence interval which shows that the extended IDM is behavior in a more careful manner and started braking early and smoothly in the presence of occluded regions in the environment.

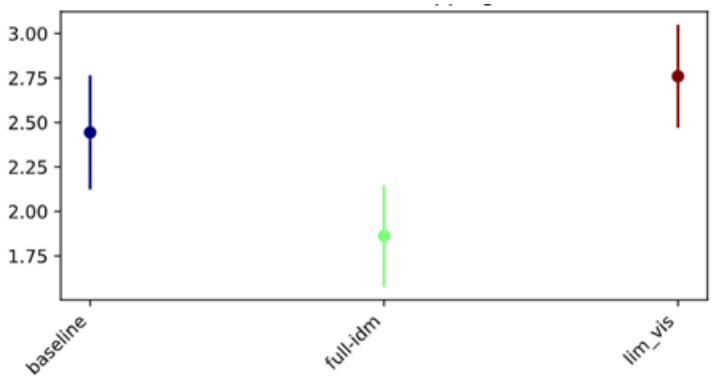


Figure 4.6: Result of Maximum Deceleration during Braking KPI.

4.6 Enhancement of Visibility Feature

The rollouts generated from extended IDM model has information of reduced object lists. Which means that certain objects are hidden due to occlusions. The ability of existing policy model is limited to understand the reduced list of objects. Hence, the existing heuristics planner can be enhanced to provide a reasonable behavior by added an additional channel to the input rasterized grid images. The new channel includes the information about occluded and non occluded regions of the map. By doing so it is possible to obtain a better feature representation and planner (policy) can be restored to execute the baseline behavior in presence of occluded regions.

The extended IDM evaluated in open loop with the simulation environment gives a series of rollouts. Each rollouts has number of frames representing sequence of actions. The frames can be filtered using the object filter algorithm and outputs a reduced set of frames with objects positioned outside the occlusion zones. The filtered frames can be used to extract the input features. Major features required to form the grid images are: ego pose, map information, traffic participants and positions of other vehicles. Each channel is represented using these collected features. As an enhancement, in order to represent the visibility information, object filter information like: occlusion and trigger zones, and current ego position is utilized.

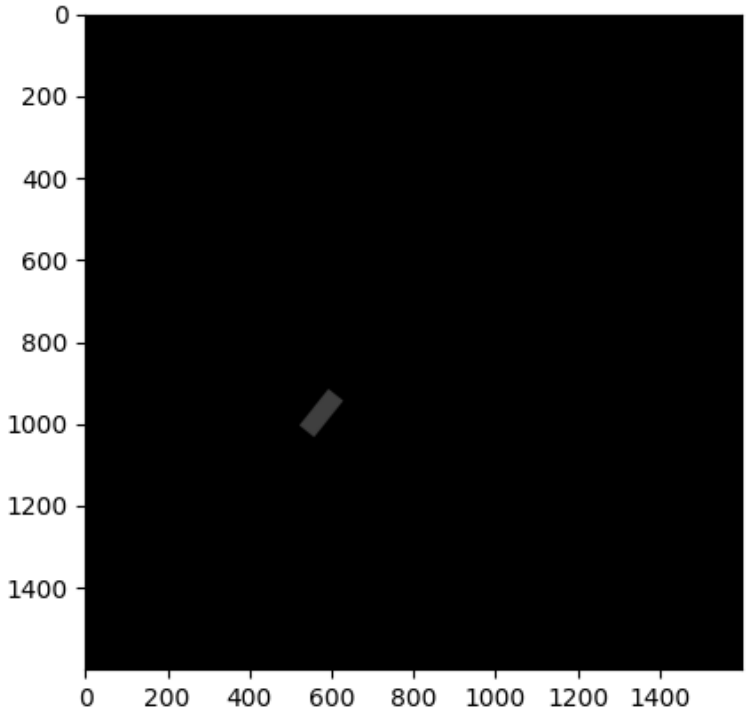
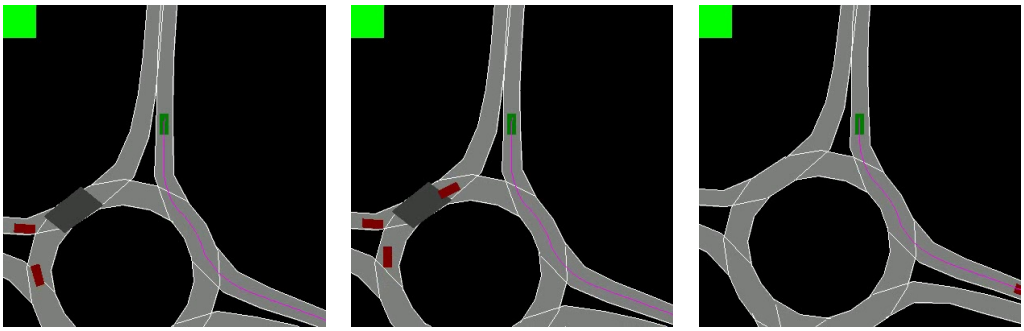


Figure 4.7: Binary Visibility Layer.

Using the object filter data extracted from rollouts and ego position, a set of active occlusion zones are found. Each zone in the active occlusions list is transformed into a frame of world coordinates. A polyline is formed using the transformed points with unit is metre and it is filled with gray shade. The filled polyline is added to the image frame of size H × W compatible with the grid data structure. Figure Figure 4.7 shows the binary image of visibility layer. The active occlusion region is masked with a grey shade polyline. As only one ego vehicle is considered in this thesis, only one of the four proposed occlusion zones can be active at any given moment. As a result, a single polyline indicating the active occlusion and its related position can be seen in the visibility layer.



(a) t = n secs (b) t = n+1 secs (c) t = n+10 secs

Figure 4.8: Visualization of Input Data with Visibility Layer.

The visibility channel added to the grid input data is visualized by generating grid video of the dataset. Sample frames are extracted from the grid video and are presented in figure 4.8. The figure shows ego vehicle as a green box and the visibility information color coded as grey area inside the roundabout region. The grey area in turn means that the area is occluded and the object inside them are hidden. The red boxes are the traffic participants. The intended route of the ego is shown as the pink line. Ego vehicle driving towards the left, sees an occluded region to its right side. Figure 4.8(a) shows that there two other vehicles in the environment. And the ego is yielding near the roundabout junction in reaction to the occlusion region visible to it at an initial time step t = n secs. Where as in figure 4.8(b) which is a subsequent frame, shows that there a third vehicle popping out from the grey occluded area. As a result it is inferred that at time = n sec in figure 4.8(a) one of the traffic participant is hidden inside the occlusion zone, which eventually made the ego vehicle to yield. After some time steps, the visibility layer of the occlusion zone disappears as the ego steps out of the trigger zone. It is shown in figure 4.8(c). Therefore, it is inferred that the visibility channel contains a mask to identify the active occlusion zone. Such masked channel information enables the policy model to understand the presence of invisible regions.

4.7 Policy Model Enhancement

In the baseline policy model ResNet34 [22] truncated at features layer 1 is used as a backbone model to extract a latent representation of the input grid images. The truncation is done to generate latent features of dimension compatible with the MLP and prediction heads. As the input data is enhanced with additional channels in this thesis work, an alternate backbone model is required to achieve the desired driving behavior either inline or beyond the performance of the existing baseline policy model.

4.7.1 Backbone Architectures

A deep look into the recent advances in the convolutional networks have given more insights on the new developed models. Four different CNN-based models are selected to experiment with policy training. They are ResNet50 [22], MobileNet-V2 [59], EficientNet-B0 [66] and ConvNeXt-Base [49] models. Among them ConvNeXt architecture is explained in detail in this section.

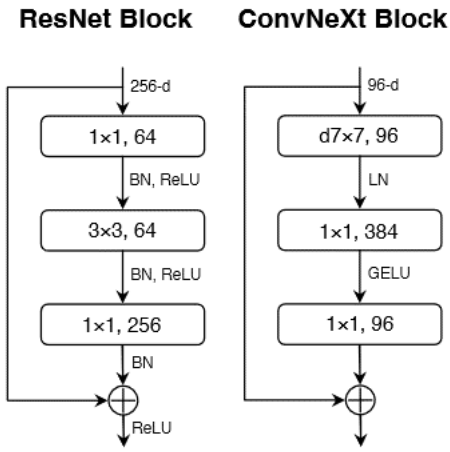


Figure 4.9: Architecture Comparison of ConvNeXt and ResNet Blocks [49].

In recent times, vision transformers plays a major role in image classification and surpassed convolutional networks. But vision transformer by itself faces difficulties when applied to general computer vision tasks such as object detection and semantic segmentation. A hybrid approach is introduced that makes Transformers a viable vision backbone. However, the effectiveness of such hybrid approaches is still largely credited to the intrinsic superiority of Transformers, rather than the inherent inductive biases of convolutions [49]. ConvNeXt are designed using ConvNets [23, 64] and have shown to have comparable performance as that of Transformers.

ConvNeXt

A detailed description of ConvNeXt are as follows, the idea started with resNet50 and enhanced the macro design patchifying the stem. It also adapted ResNeXt based structure with a combination of depth-wise convolution and 1×1 convolution that supports in aggregating the information spatially and across the channels respectively. The network width is increased as proposed in ResNeXt [70] and leads to increased FLOPS. The model is also designed to have a global receptive field by making use of large kernels of size 7×7. As ConvNets extensively used ReLU [29, 53] activation function, a more smoother version of ReLU which is Gaussian Error Linear Unit (GELU) [26] is utilized in ConvNeXt. More focus is given to making use of fewer activation's and normalization layers. Batch norm (BN) [31] in usual ConvNets as replaced with Layer Norm (LN) [1] in each residual block. Moreover, separate downsampling layers with 2×2 convolution with stride 2 are used. This modifications stabilizes with additional normalization layers whenever spatial resolution is changed. The architecture of ConvNeXt is shown in figure 4.9. Overall, ConvNeXt has benefits than ResNet34 in terms of accuracy and scalability, while maintaining the simplicity and efficiency of standard ConvNets.

Therefore, considering all the benefits of ConvNeXt, the base version is used as an enhancement to the backbone architecture of the policy model to generate feature representation in latent space. Additionally, the ConvNeXt-base is truncated after the first feature layer to generate features of suitable dimensions.

4.8 Summary

In summary, this chapter demonstrates the methodology adopted to handle the occluded roundabout environment. An occlusion is induced to the highway-env simulation environment using the newly defined object filter algorithm. The configurations of occlusion and triggered zones are depicted along their use cases. This thesis report proposes an enhancement to the expert model resulting in a new extended IDM that can handle the occluded zones by showing a more careful driving. The extended IDM is evaluated using newly defined KPI metric which showcases a comfortable driving behavior. In order to enable the policy model to handle the occluded roundabout more effectively, the input grid dataset is enhanced with a visibility information. This visibility feature is encoded to the training datasets as an additional channel. Moreover, in order to handle the datasets with enhanced the features a variant of convolutional network i.e, ConvNeXt is proposed to be used as the backbone of the policy model.

5 Experimental Setup and Results

The process of generating rollouts from extended IDM demonstrations are described in detail in this chapter. In the next step, the list of datasets generated for training the policy model are discussed. This chapter also covers the details of policy training procedure. The results of policy trainings are also presented with a detailed discussion. Finally, the hyperparameter tuning of the new policy model is discussed.

5.1 Rollouts Generation

Rollouts comprises of one scenario with nonzero duration. For example, a drive through a roundabout junction in this case. It contains perception output in the form of object lists including ego state.

Theoretically, the development of a neural network for behavior planning can only take place on the basis of real measurement data. However, it is helpful to have a simulation environment in which it is possible to simulate a road environment as well as the movement of other vehicles. This can then be used to simulate further training data and to evaluate the trained model in a closed-loop simulation. For this, an environment developed in-house based on the open-source simulator highway-env [43] is used. Highway-env makes it possible to perform driving simulations on various maps created in-house. In addition, a connection to the open source data set is possible. "highway-env"can be used to create synthetic training data or to test the trained model in a closed-loop simulation. The vehicles that are not controlled by the own planner are controlled by the IDM. The objects and map data generated in the simulation environment are features, which then serve as input for the expert driver model. The IDM, in turn generates waypoints that are returned to the simulation environment. This enables driving scenarios to be generated and the resulting features in conjunction with the waypoints are saved as a data record. Simulated data is also stored as a rollout.

In simple terms, one rollouts represents one sequence of planned action. The rollouts are generated from dataset or simulation environment by driving the expert as depicted in figure 5.1. In addition, rollouts can be generated for environment with various scenarios. Around 1000 episodes are generated for each scenario. Out of all the scenarios specified in section 3.4.1 only empty, general, yielding, yielding preceding are taken into consideration when extended IDM is used as the expert model to generate rollouts. The reason behind this due to the fact that, occlusion handling problem is mainly focused on the yielding and other that are relevant to it. Where as, the rollouts for all other scenarios such as stopping, drive off and off-road parked cars are generated using baseline IDM as it can handle these special topologies. These scenarios are generated using scenario catalogues.

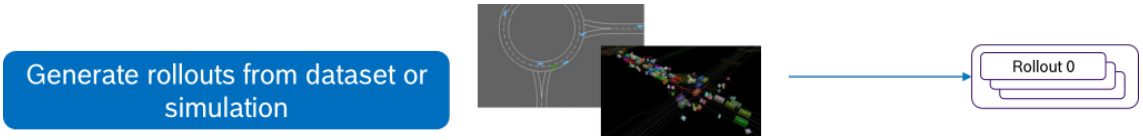


Figure 5.1: Rollouts Generation Process [41].

In this thesis, two sets of rollouts are generated. First set of rollouts are from the baseline IDM as expert in a default roundabout environment with no occlusions. This set of rollouts are named as *baselineIDM* rollouts. Other set is *extendedIDM* rollouts which are generated using new IDM as expert model over a roundabout environment with occlusions induced via object filter config.

5.2 Dataset Generation

The training data for imitation learning is primarily taken from simulated environment, with some real data thrown in for a good measure. The real data, is gathered from the rollouts that are generated by driving through a roundabout in the real world. The simulation data is generated from the rollouts created using highway-env [43] simulation framework.

After the rollouts have been generated, the next step is to standardize them by creating data records. Only one feature generator and label generator are required to create the training and validation data sets from all data. When training the models, i.e. in imitation learning, the model is trained using the features. The models outputs a set of waypoints, which are evaluated by means of the loss function in conjunction with the labels. A trained model can then be evaluated in the simulation environment. The scenarios generated are stored as a rollout and evaluation metrics are calculated.

Different versions of datasets are generated for the purpose of testing the model performance in handling occlusions. The proposed methodology is applied in steps to the dataset. The list of five datasets generated and their corresponding sources are listed:

1. **B0** - rollouts are from baseline IDM demonstration.
2. **B1** - rollouts from baseline IDM evaluated in default environment. Thus rollouts has full list of objects but certain objects in the specific regions are filtered or masked during dataset generation.
3. **B2** - same as the previous one, uses baseline IDM rollouts and applies filter during dataset creation. Additionally, adds visibility layer to the input features in dataset creation.
4. **E1** - rollouts are generated form extended IDM demonstrations in an occluded environment.
5. **E2** - rollouts from extended IDM are used along with that visibility information of occluded and non-occluded region is encoded into the grid features of the dataset.

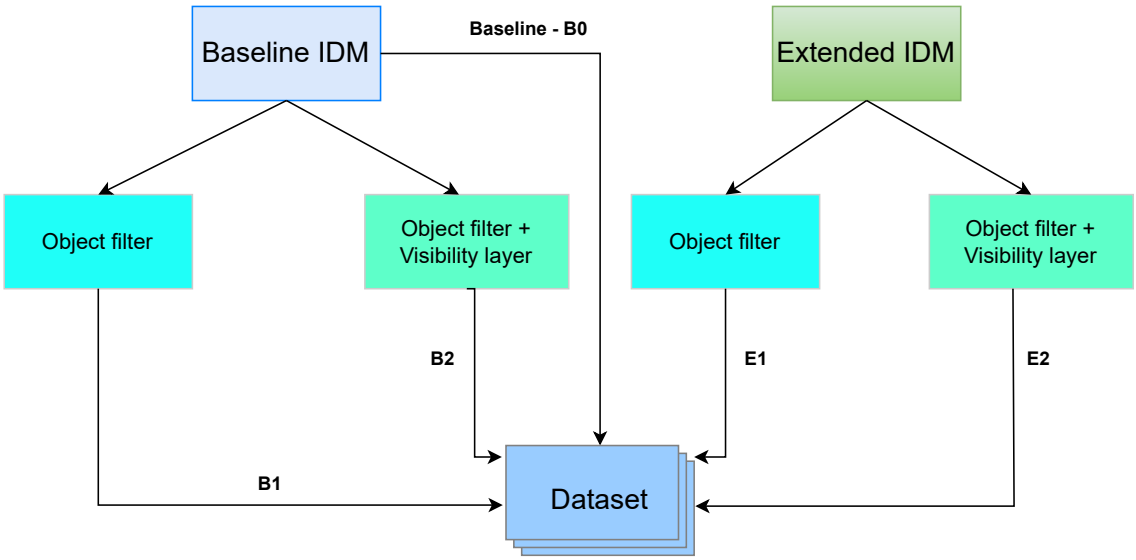


Figure 5.2: Illustration of Dataset Generation.

The process of datasets generation based on the source IDM is described in the figure 5.3. It could be easily inferred from the figure that B*x* series of datasets are generated from baseline IDM. Whereas, E*x* series of datasets are generated from extended IDM.

5.3 Policy Training

Trainings are carried out with different model configurations and the result is evaluated on the basis of different metrics. The main aim is to ultimately obtain a model configuration that makes it possible for the trained policy to handle occluded roundabout environment while preventing crash and discomfort in driving behavior. For this, the policy network with different backbone models are implemented and with same set of training datasets and Hyperparameters. By analysing the training results, it is possible to determine which model configuration is well suited to learn a policy.

The training datasets are generated from expert demonstration in simulation environment with and without occluded regions. The list of datasets used for training are listed in 5.2 along with the details of the features generator configuration. Each datasets consists of features and their corresponding labels. The labels serve as a ground truth for the supervised learning technique. Datasets from 8 different roundabout driving scenarios are used for training. The datasets are split into train and dev (also known as development or validation) subsets for training and validation processes respectively. Both train and dev datasets are encountered with augmentation using random rotation similar to [2]. With the given probability value, the input features and labels are rotated around the centre with respect to ego location. The rotation angle is uniformly sampled between [-degrees, degrees]. Gradient clipping with pre-determined gradient threshold is used during training the policy model. This prevents any gradient to have norm greater than the threshold and thus the gradients are clipped. With this, the problem of exploding gradients is prevented and aids in stable training.

A combination of loss functions as specified in section 3.4.5 are used for model training. As the datasets are represented as rasterized grid images, the shape of input grid images is H × W × 10 for datasets without visibility layer and H × W × 11 for datasets with visibility channel. The input images of size 64 × 64 is employed during training. Various training experiments are conducted for each dataset. We began by training the baseline policy model using ResNet34 as backbone inspired from [73] using IL technique. Later, enhanced the backbone to latest versions of ConvNets as specified in section 4.7.1 retrained. Each training is executed for 50 epochs with a batch size of 512. The learning rate is 0.001 and an exponential learning rate scheduler [46] is used as the optimizer with $\gamma = 0.94$. The clip gradient norm threshold is 1. with The training data is augmented with a degree of 5 and probability of 0.5. Finally, a regularization technique with weight decay of 0.0004884316427125545 is utilized in training. The output of the model is 15 waypoints as x and y coordinates. All the experiments are conduced on a NVIDIA Titan GPU that helps to accelerate the learning of the policy function.

5.3.1 Training Procedure

Besides our the proposed model with ConvNeXt backbone using extended IDM rollouts along with visibility layer as input data, we also train and test the models without visibility layer, also with and without object filter (i.e, occluding objects) for a detailed comparison. The datasets used for training are already illustrated in figure 5.3. The training procedure with dataset and model configurations are described in figure 5.3 for better understanding. Accordingly, the policy model receives five types of datasets namely B0, B1, B2, E1 and E2 as inputs. Next, the backbone models are varied as mentioned in section 4.7.1. Accordingly, the arguments and discussions in this thesis are only based on ResNet34 and ConvNeXt based policy model. All the models are trained in an end-to-end manner with same set of datasets under same conditions.

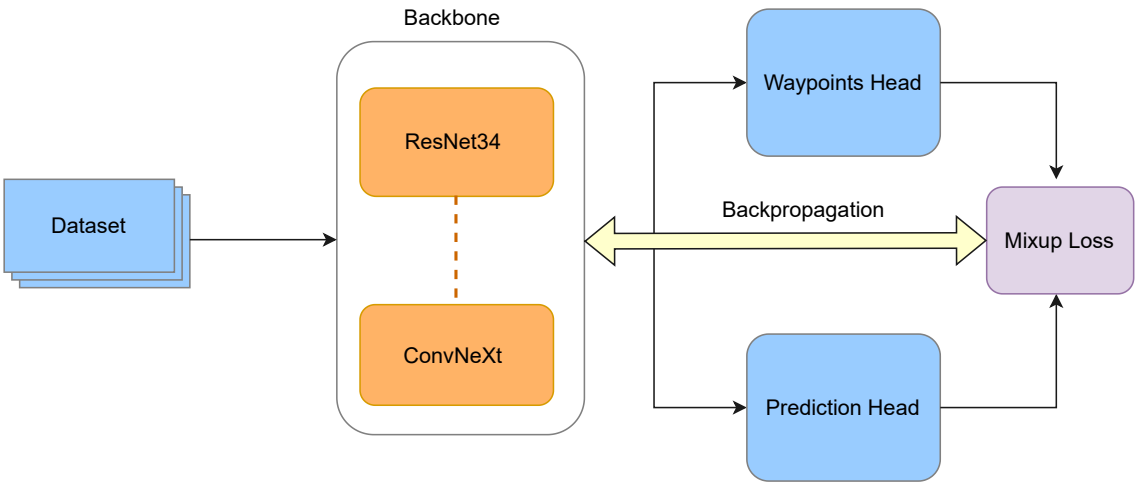


Figure 5.3: Training Procedure of Policy Models.

5.3.2 Training Results

During training the policy models are varied with different configurations of datasets and backbone model, resulting in five training courses for each backbone. The metrics with which the models are evaluated are the training loss and the development (or validation) loss. These are averaged and compared for each model. The training results are shown in table 5.1.

The details of training and development losses are specified for each training configuration. On analysis of the training results concerning the datasets, B0 is the baseline dataset generated from the baseline IDM model. It is observed that the models using datasets from baseline IDM i.e., B0, B1, and B2 have almost the same loss values for a configured backbone model. In which, B1 and B2 datasets are created from baseline IDM rollouts with object filter and visibility layer configurations. Whereas, the training and development losses of extended IDM based datasets (E1 and E2) are marginally low when compared to the B*x* datasets. The reason behind such observation is that the B1 and B2 datasets are created from the baseline IDM model that has complete visibility of the environment. So the policy is unable to get the complete information that the expert already has seen. On the other hand, E*x* datasets are easier for the policy model to imitate because the datasets are created from by new expert which also has limited visibility same as what is known to the policy model. A contributing factor for E*x* datasets is that a part of the datasets scenarios are generated from baseline IDM.

Table 5.1: Results of Policy Training.

Dataset	ResNet34		ConvNeXt-Base	
	Training Loss	Development Loss	Training Loss	Development Loss
B0	0.371	0.392	0.417	0.42
B1	0.385	0.401	0.42	0.42
B2	0.383	0.418	0.399	0.411
E1	0.281	0.288	0.296	0.295
E2	0.282	0.286	0.3	0.289

In regards to the backbone models, both ResNet34 and ConvNeXt have comparable performance. But still, when takeing a deep look into the loss values of the ResNet34 backbone model it is observed that validation loss is above the training loss. This is a sign of overfitting. Whereas, the ConvNeXt-base model does not have a high difference between the training and validation losses. The training curves of the E1 dataset for ResNet34 and ConvNeXt-base are shown in figure 5.4 as a sample. The training curves are create with MLFlow framework. It is evident that for the dataset E1, the overall validation loss is at the end above the training loss for ResNet34, and ConvNeXt validation loss settles slightly below than train loss. This effect is observed in all the loss components used. The result is a model that can be trained well based on the of the new features generated from extended IDM. Thus, ConvNeXt is selected as the suitable backbone which can handle the feature from the expert that has limited visibility.

Overall, when using the datasets generated from Extended IDM with object filter and additional visibility layer has shown a good improvement in the policy in handling the occlusions of the environment. It is highly evident from the corresponding loss values. Additionally, the training

losses of ConvNeXt-base and ResNet34 are comparable. ConvNeXt has shown a stable training when fitted into the multi-task policy model. This selection is justified in detail with the evaluation results discussed in chapter 6.

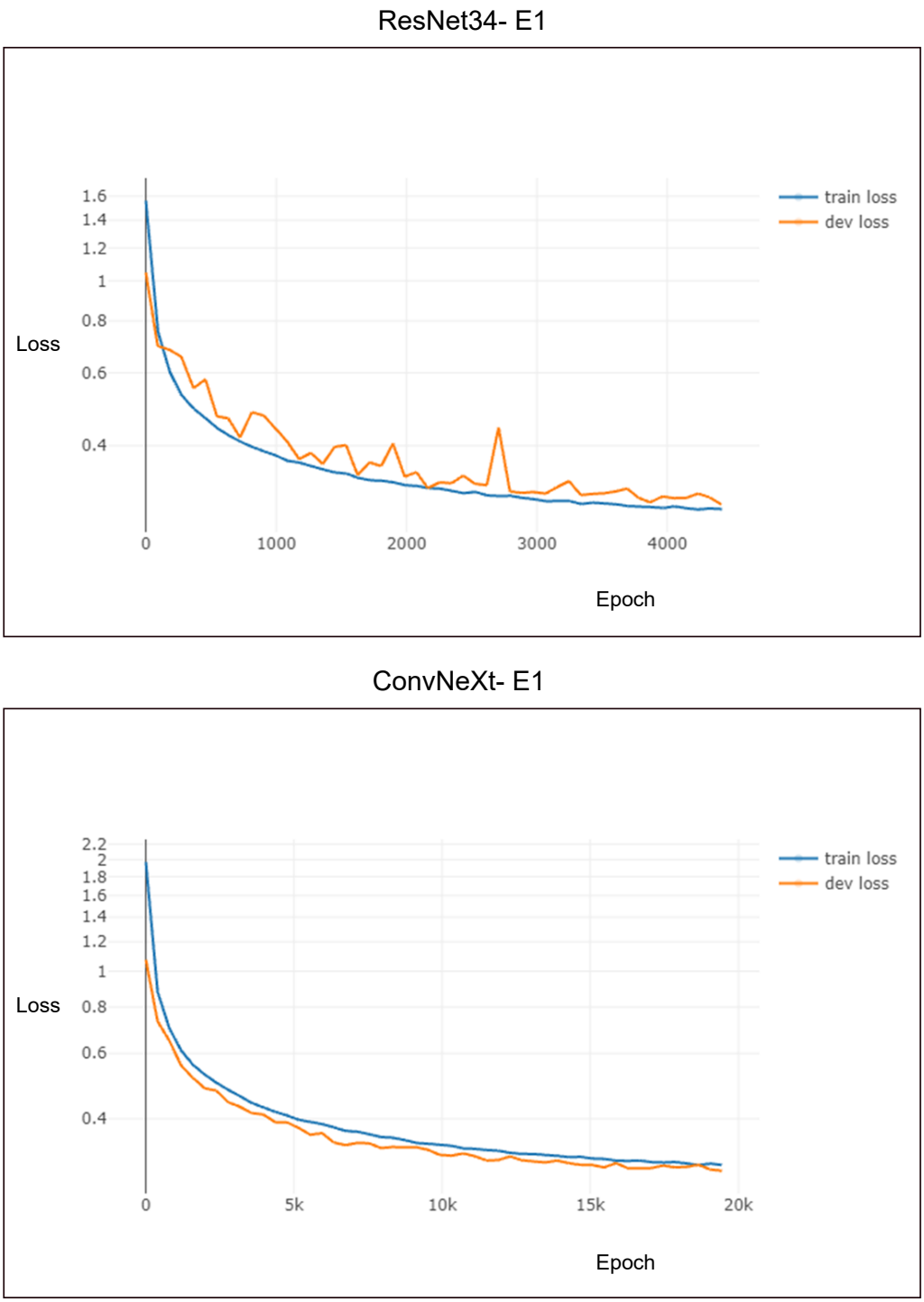


Figure 5.4: Loss Curves of ResNet34 and ConvNeXt-Base Policy Models Trained with E1 Dataset.

5.3.3 Hyperparameter Optimization

The model performance strongly depends on the optimal setting of internal hyperparameters [25, 50]. In our case, the policy model is a multi-task network where manual tuning of the parameters is not desirable. Hyperparameter optimization (HPO) plays a major role in automatic tuning of the model parameters. The validation of the performance of machine learning algorithms can be modelled as a function of their hyperparameters. The HPO problem is then defined as finding the hyperparameter setting that minimizes this objective function. In this thesis, Bayesian Optimization and Hyperband (BOHB) algorithm is used for optimizing the hyperparameters. The BOHB algorithm is described in detail as follows.

BOHB [16] is a well-known hyperparameter optimization approach. It incorporates Hyperband [45] and Bayesian Optimization [63] concepts. According to Falkner et al. [38], BOHB makes progress more quickly than vanilla Bayesian optimization and random search approaches. It has been shown to produce superior results than Hyperband and random search. The components of the BOHB technique is discussed as follows.

Bayesian Optimization (BO) models the objective function using a probabilistic model based on a set of previously observed data points [16]. BO identifies improved configurations using an acquisition function depending on the present model. The acquisition function can be used to adjust the trade-off between exploration and exploitation. BO takes time to establish a good model in the beginning. As the budget grows, the model acquires more and more knowledge about the search space. In general, BO requires high budgets to find optimal parameters. On smaller budgets, Hyperband (HB) is the best suitable HPO approach that uses cheap-to-evaluate approximations of the objective function. HB uses SuccessiveHalving (SH) [36] to find the optimal configuration. SH examines n randomly sampled configurations with limited budget and retains the best half while doubling the other half.

BOHB combines the advantages of BO and HB in an efficient way. It follows HB's method of selecting budgets and continues to employ SH but utilizes a BO component to perform the random sampling to guide the search. In a nutshell, BOHB uses HB to build a model and BO to choose a new configuration based on previously evaluated configurations. In the beginning, BOHB uses HB's method of swiftly assessing many configurations on a little budget. This enables BOHB to quickly find and refine the selected configurations, resulting in a strong final performance of BOHB. BOHB evaluates many configurations in each iteration, each of which can run separately on multiple workers, allowing BOHB to make efficient and effective use of parallel resources. This can help to accelerate the optimization process. As a result, in addition to the desiderata that is satisfied by BO and HB, BOHB follows the simplicity and computational efficiency making it a robust, flexible and scalable HO method. Thus, BOHB is used for optimizing our behavior planning policy model.

The hyperparameters of the baseline policy model is already optimized using BOHB method. The internal hyperparameters like: learning rate, weight decay, learning rate scheduler are tuned. The optimization is executed for the new policy model with ConvNeXt-base backbone as well. As a results, it is inferred that the default hyperparameters which are tuned for ResNet34 backbone, fits the best for ConvNeXt backbone as well. Hence, the policies are trained using default hyperparameters that are specified in section 5.3.

5.4 Summary

In summary, this chapter elaborates on the procedure for generating rollouts from IDM model using simulation simulation. Subsequently, the dataset generation process is also explained. Five types of trainings datasets are generated from baseline and extended IDM rollouts. As a next step, using the generated datasets as inputs, the two versions of policy models are trained, one with ResNet34 and other with ConvNeXt-base as backbone. The performances of policy models are compared based on their training results. It turns out that the behavior planning architecture is more efficient with ConvNeXt backbone especially for handling the datasets from extended IDM that has limited visibility. The statement on ConvNeXt is further justified with even more results during experimental evaluation in an occlusion environment which is discussed in the next chapter.

6 Evaluation

Various evaluation studies are conducted to verify the performances of extended IDM as well as the new policy model. The research questions [RQ3] and [RQ6] are addressed through detailed experimental evaluation of the IDM and the policy model using various datasets. Initially, the scenarios used for closed-loop evaluation are described. Followed by that, the evaluation metrics and their corresponding importance are specified. At first an ablation study is conducted with the expert IDM models. Next, the influence of training datasets over the policy model is closely examined by evaluating the policy model with default and occlusion environment. Finally, the chapter ends with presenting the visual results.

6.1 Evaluation Scenarios

The evaluation experiments are performed using highway-env simulation environment in a closed loop setup. All the experiments are conducted using uniform scenarios. The scenarios are defined in the simulation environment using the scenario generator accompanied with the simulation environment. The roundabout road network, the initial and target speeds and number of vehicles to be simulated are configured via scenario generator. Out of the various driving scenarios the one's that are mainly considered for evaluating the occlusion handling ability are categorized into three types:

6.1.1 Scenario Empty

Second scenario used to test the drivability of the ego vehicle is empty. For this purpose, an empty roundabout highway scenario is generated in which no other vehicles are available. The initial position of the ego vehicle is randomly selected on one of the four possible pathways of the roundabout.

6.1.2 Scenario Yielding

Yielding is the scenario, in which the roundabout has ego along with other controlled vehicles in it. In this case, the ego approaching the roundabout might have same route as the other vehicles and it must yield before the junction to avoid collisions and to obey the traffic rules. Here, the initial position of the ego vehicle is selected on any of the four pathways.

6.1.3 Scenario General

First scenario is general, which simulates the traffic situation on a busy roundabout, where the ego's behavior depends on the other traffic participants. Here, interaction with other road users is particularly important. The distribution of general scenario is a combination of all other scenarios for example, empty, yielding and yielding-preceding. The topology again consists of a four way roundabout, where both the initial position of the ego vehicle as well other vehicles are selected randomly. The other traffic participants are controller by IDM.

6.2 Evaluation Metrics

The models both IDM and policy are evaluated on the basis of four different evaluation points (KPIs). The KPIs are used based on there performance categories: success, safety and comfort metrics. Safety is ensured with collision ratio and off-road driving ratio metrics. The KPIs represent the mean value of the metric on a single rollout episode, where the mean is taken over the distribution of scenarios. The confidence interval of the results are represented with underlying distributions are normal, binomial and poisson distributions. Normal distribution is the default case and it is used for continuous observations. Binomial distributions is supported in case of binary events that may or may not happen in a rollout (i.e., only values 0 or 1 are taken). Whereas, poisson is mainly used for counting the number of discrete events in rollouts. Furthermore, KPI distributions are calculated with 95% confidence interval. Hence, the resultant metrics will have a mean value along with the confidence interval.

In this thesis various experiments are conduced for evaluation. There are two category of evaluations performed: (1) Expert evaluation and (2) Policy Evaluation. All these experiments are evaluated using four KPI metrics of different categories mentioned earlier.

6.2.1 Success Ratio

Success ratio determines the percentage of the runs where the ego (car) successfully reached the goal. This metric takes into account both collisions and time-outs. This is a metric specific to special driving cases like roundabout. To calculate the success rate, a start and end points are defined for each case, such as start at several meters before entering a roundabout, and end at several meters after passing through it. We compares all the versions of expert models 4.4 and policy models under success ratio metric.

Here, the success ratio is measured based on certain criteria that defines the objective. Major objectives in the evaluations are: reach the destination point. For each experiment the proportion of rollouts i.e., trials that satisfied the objectives are measured against total number of rollouts. Additionally, success ratio is also affected by the safety criteria like crash (or collision) and off-road driving behaviors. Success ratio is measured using binomial distribution of whether the objective is satisfied or not.

6.2.2 Collision Ratio

Collision ratio is a measure of safety of the method. In this case, the definition is little different with the metric defined in [14], where they classify the collision events with respect to different kinds of objects such as vehicles, bicycles and pedestrians. In this thesis, a collision event means collision to other vehicles. Since we do not have pedestrians and bicycles in our environment. Collision ratio is also measured based on binomial distribution. If a crash or collision with other vehicle is observed then collision rate is incremented. The resultant metric indicates proportion of rollouts ended in collision.

6.2.3 Off-Road Ratio

This KPI measures the number of times ego goes off the road from the driving lane. It also comes under safety metric and follows binomial distribution by nature.

6.2.4 Maximum Deceleration during Braking

The fourth KPI falls under the category of comfort metric that is maximum deceleration during braking which is already discussed in detail 4.5.1. This metric can be seen as a measure of how hard the ego vehicle brakes before stopping in response to an observed traffic. This proposed metric measures the maximum of the braking accelerations calculated with two thresholds.

6.3 Expert Ablation Study

The IDM is the expert model used to produce set of demonstrations for training the policy model. In this work of handling occlusions in the roundabout environment, a set of enhancements are proposed to the IDM as stated in section 4.4 to enable the IDM to handle the limited visibility in the environment created by the presence of occlusions. Whereas, the default expert i.e., baseline IDM has full visibility of the environment and the objects present in it. The capability of baseline and Extended IDM to handle the limited visibility is analysed by an extensive ablation study. For the purpose of ablation study, a third version of IDM is used called as limited-visibility IDM. The characteristic of limited-visibility IDM is that it has the visibility to a limited set of vehicles in the environment. In simple terms, the limited-visibility IDM is evaluated in the occlusion environment. Overall, three IDM models are taken into consideration for evaluation in which only the baseline IDM is evaluated in the default fully visible environment. Whereas, the extended and limited-visibility IDM's are exposed to occlusion environment.

For a fair comparison, all three IDM models are evaluated in closed loop simulation using same KPIs. Evaluations are performed for three test scenarios mentioned in section 6.1 using the metrics specified in section 6.2. Out of the three scenarios yielding is the main focus when dealing with occluded environment. As we are aiming to achieve the human driver behavior of slowing down and crawling in case of occlusions, yielding scenario deals with the similar behavior as desired.

Figure 6.1 shows the results of success and comfort metrics of three expert models corresponding to each test scenario. As none of the three expert models faced collisions or off-road driving in any of the scenarios, safety metric category is ignored as they remains zero for all three models.

The x and y axis of the plots are model type and KPI values respectively. Each plot describes the output distribution with a confidence interval of 95% with the mean value indicated as a point on the line. By analysis, the mean value of success ratio of all three expert models are the same in all the test scenarios but with varying confidence interval. This result on success ratio demanded the need for a new proposed metric called Maximum deceleration during braking (m/s^2) as defined in 4.5.1. On analysing this new metric, it is evident that the Extended IDM in the yielding scenario has a lower maximum deceleration value as it brakes smoothly when it knows about the limited visibility zone while traveling through the trigger zone. Meanwhile, baseline and limited-visibility experts brakes hard causing driving discomfort in the yielding scenario. In case of empty scenario when there are no traffic participants in the roundabout environment, baseline and limited-visibility experts has lower deceleration, as empty scenario doesn't make a difference in default and occlusion environment. But, the extended-IDM has a higher metric value comparatively. The main reason is that the expert is being more careful while approaching the roundabout with occlusions irrespective of the number of vehicles in the environment. Due to such alertness of the extended-IDM expert, it slows down inside the trigger zone while approaching the roundabout as it realizes that limited visibility of the environment. Hence, the performance of the extended-IDM is tested using the maximum deceleration during braking KPI and it is proven to handle the occluded environment similar to the human driver which is the desired behavior.

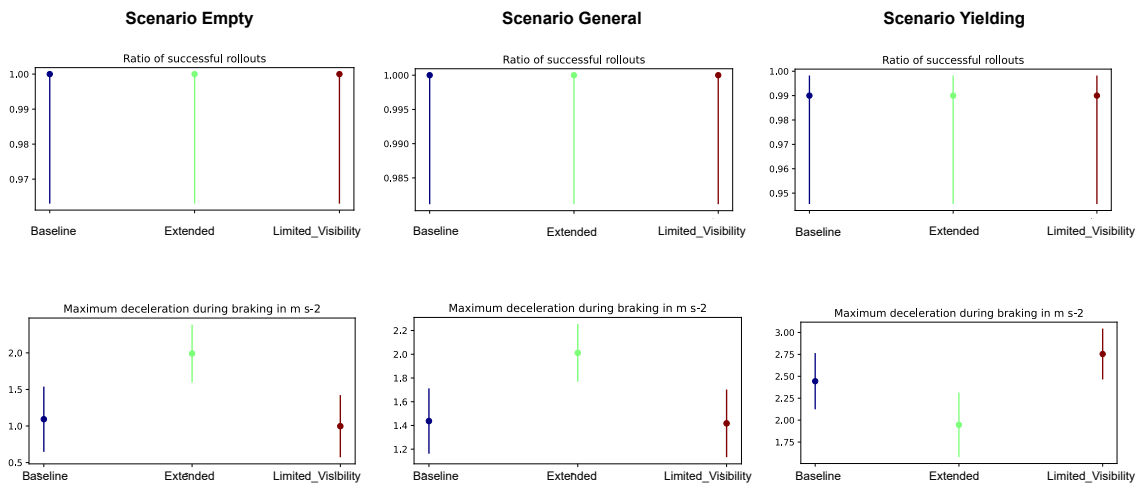


Figure 6.1: Results of Expert Ablation Study.

6.4 Policy Model Evaluation

The policy model trained using demonstrations from expert model are evaluated using closed loop simulation technique. Here the policy model is driving the agent forward, hence this is a closed-loop evaluation setting.The evaluations are carried out for three test scenarios as discussed already. During the simulation, we assess the success, safety and comfort specific metrics as mentioned in section 6.2. When it comes to occlusion handling, yielding is the most important test scenario. The

main goal is to create the best performing policy model that can handle occlusions in the simulation environment. The techniques proposed in this thesis, is expected to perform as good as the baseline policy model, which is trained with a complete visibility of the environment.

Various evaluation experiments are conducted using different backbone models like: ResNet34 (baseline), ResNet50, MobileNet-v2, EfficientNet-B0 and CovNeXt-base. Among them, the ConvNeXt showed a better performance based on the results from section 5.3.2. Hence, the discussions are narrowed down to baseline policy model using ResNet34 backbone and enhanced policy model using ConvNeXt-base backbone.

6.4.1 Comparison of ResNet34 and ConvNeXt

The baseline policy model uses the ResNet34 as an important module in the model architecture, which makes it possible to process top-down grid images. The advantage of ConvNeXt is that it can make use of images with more channels and produce a transformed one-dimensional image in the latent space with more accuracy. The following shows that this advantage also has a positive effect on the learned policy. For this purpose, two network architectures trained using three datasets (B0, E1 and E2) under same conditions are compared with each other.

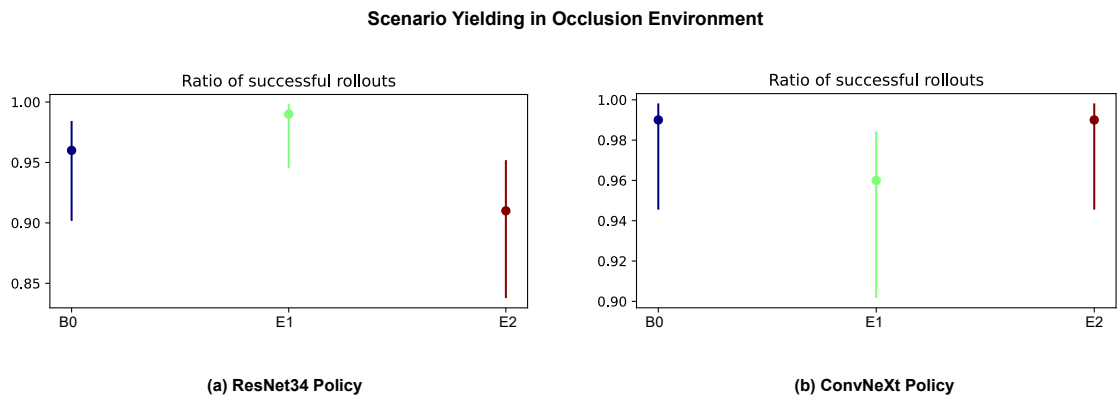


Figure 6.2: ResNet34 vs ConvNeXt-Base: Evaluation Results.

Figure 6.2 shows the evaluation of the two architectures in a closed loop simulation. The figure illustrates the success ratio of two versions of policy models for three types of datasets i.e., B0, E1 and E2. Here, the evaluation under occlusion environment with yielding scenario is illustrated. In general, the baseline of ConvNeXt is better than the ResNet34 baseline, which means ConvNeXt has more ability to handle the occlusion environment even without any careful demonstration from the expert.

With respect to the datasets based on Extended IDM (E1 and E2), though ConvNeXt failed to produce good results with E1, it is able to restore its baseline performance with E2 when a visibility layer is added to the new expert rollouts. Meanwhile, with ResNet34 backbone, only object filter (E1) has enhanced the model performance against its baseline. But when an additional visibility channel is added to its data (E2), the performance drops, which shows it cannot perform the task of occlusion handling more successfully.

Overall, the architecture with the ConvNeXt on an average results in more points on success ratio. This shows that the architecture generally works better with the ConvNeXt. In addition, it can be concluded that the selection of the architecture, as concluded in section 5.3.2, is justified. The reason behind the conclusion is that the ConvNeXt can process the grid features as input even better then ResNet34. The performance difference between both the backbone is not very large, but still in this case it makes sense to use a CovNeXt in combination with the grid images for a better efficiency.

6.4.2 Influence of Training Data

The policy model learns from the demonstration data in imitation learning. They have a significant impact on the model's output in this regard. The demonstration data in this scenario consists of simulated data and a few percentage of real data from roundabout environment.

The policy model with ConvNeXt backbone is trained with different compositions of training data, as discussed in section 5.3. Polices are evaluated in default and occlusion environment under three different test scenarios. Evaluations are compared using four different metrics. The goal of this ablation study is to better understand how different components of training data affect policy model performance. The target of policy evaluation is to see, if an enhanced dataset with more useful information can produce the same quantitative results as its expert model. Initially the evaluations with default environment is discussed and then observations with occlusion environment is elaborated.

Evaluation in Default Environment

At first the policy model with ConvNeXt backbone is evaluated in default environment which has no occlusions in the roundabout. As the distribution of general scenario in the highway-env is the combination of interactive and non-interactive scenarios, the discussion starts with evaluation in general scenario. Figure 6.3 shows the evaluation results with general scenario for three KPIs. The metrics of all five datasets are presented. It is observed that, there is no rollouts ending in off road driving. Also only reasons for unsuccessful rollouts are collisions and policy being getting stuck. The success ratio is dropped by few percentage for B0, B1 and B2 due to collisions. While the policy trained on E1 got stuck and has less success percentage. Whereas, the policy trained with E2 gives a full success ratio in general. In regards to the driving comfort, E1 shows hard braking with high metric value, it is mainly due to the policy getting stuck. Based on the results form B2 and E2, it is also observed that by adding visibility layer to the dataset gives more comfort to driving compared to B1 and E2 respectively.

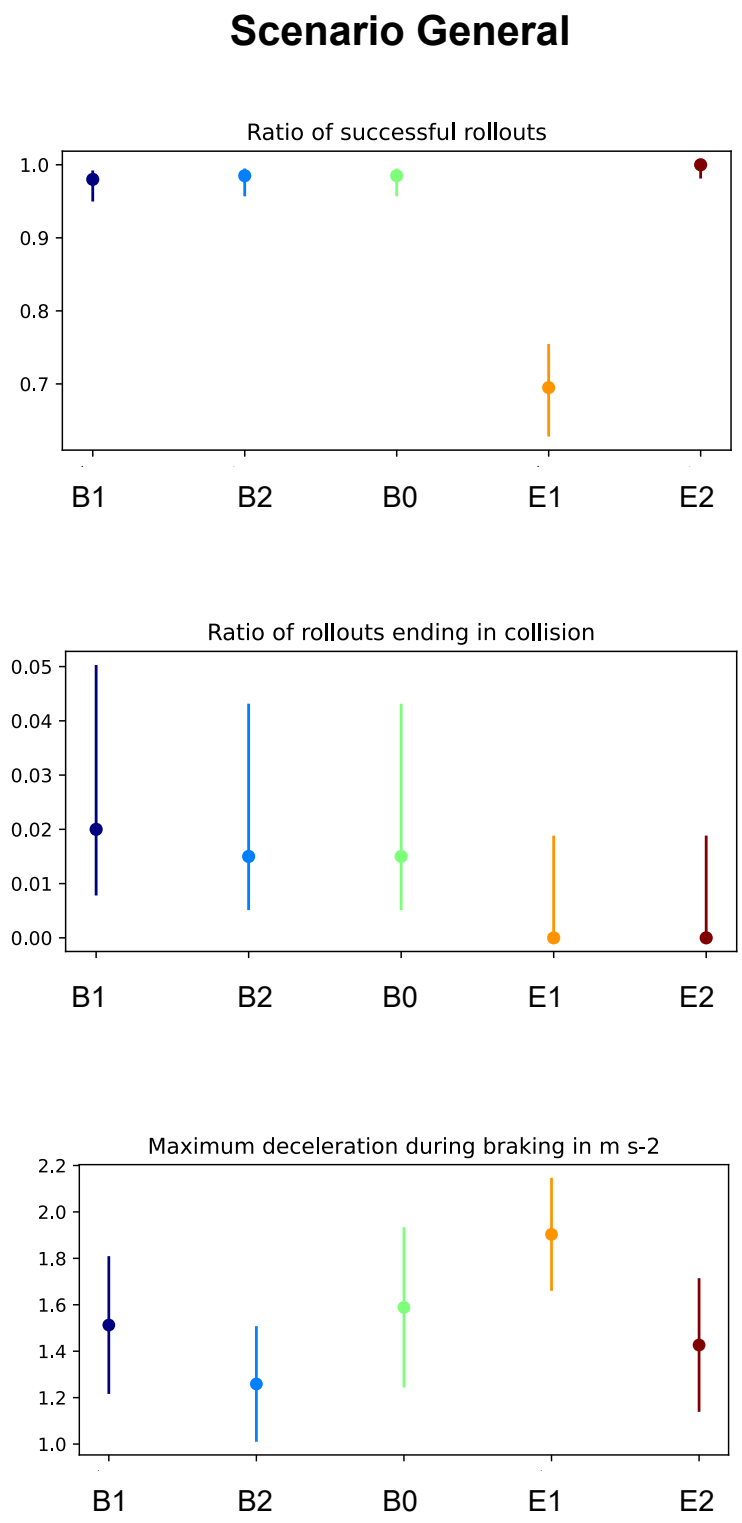


Figure 6.3: ConvNeXt Policy Evaluation with General Scenario in Default Environment.

With empty scenario, where are no traffic participants, except the model trained with E1, all other configurations enables the policy model to drive towards the destination successfully for all the trials. With E1 again the model got stuck and stands still resulting in timeout of the simulation episode. This timeout behavior also reflected in its maximum deceleration value showing a very hard braking behavior. Whereas, other dataset configurations of the policy yields a comparable metric values. This is illustrated in figure 6.4.

In case of yielding test scenario, B0, B1 and B2 faces a failure due to collision. Only the extended IDM based datasets showcases a good performance in terms of success ratio. Hence, the focus is mainly on E1 and E2 in this case. The maximum deceleration of the E1 is same as its corresponding extended IDM as described in figure 6.1. By using a visibility layer (E2), the driving gets even more smoother during braking. This shows the importance of visibility layer added to the grid images in improving the comfort of driving. Overall, though the extended IDM based policy models loses a little with success ratio, gives more comfort to drive.

After analysing the influence of each dataset for the three test scenario. We could conclude that the Bx series of datasets are generated from baseline IDM rollouts has more privileged information about the environment than what the policy have at the test time. Whereas, the extended IDM based datasets (E1 and E2) ensure that the policy has same field of view as that of its expert model. This is more similar to the realistic data and makes sense to use these datasets for training the model. Also they ensures that the policy imitates the expert better. Hence, it is more helpful to focus on the realistic datasets as there is no domain gap between environment from which the data is collected and the environment that is used for testing.

On the whole, in case of default environment evaluation, the results are comparable expect for E1. In most cases, the performance of the baseline model is almost restored.

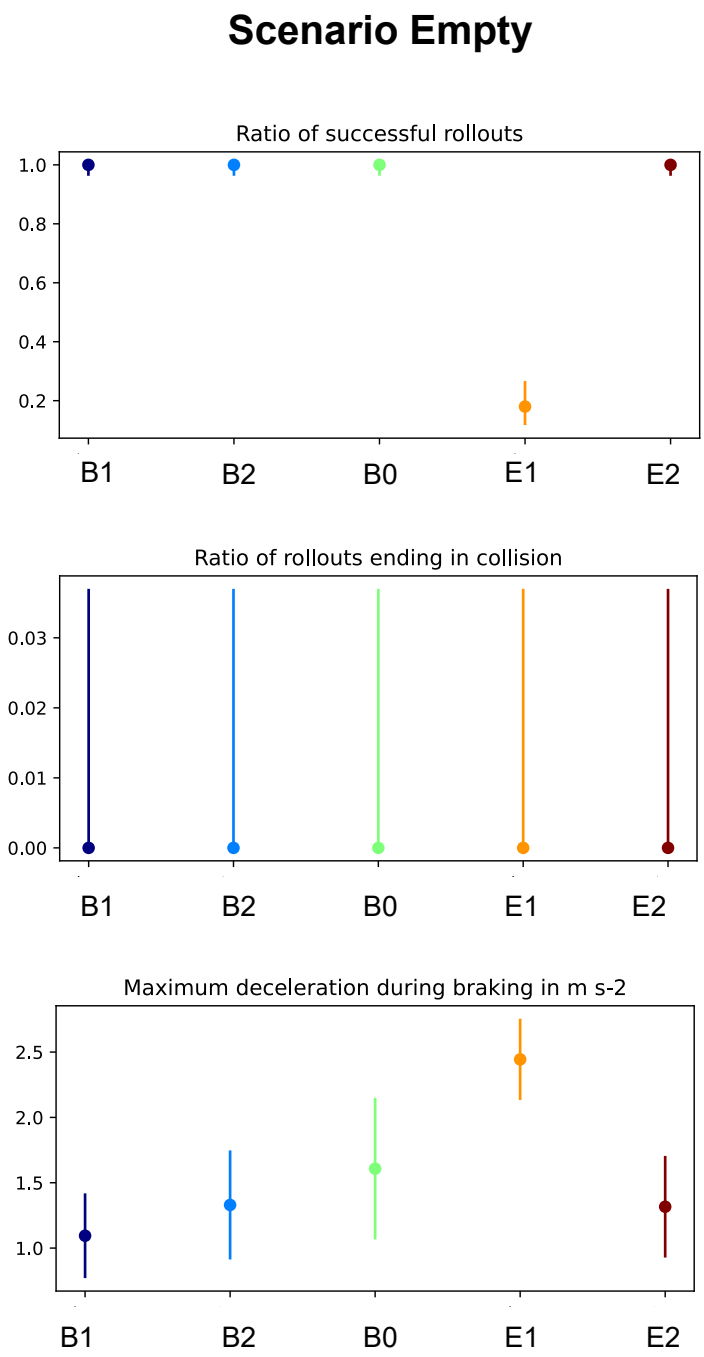


Figure 6.4: ConvNeXt Policy Evaluation with Empty Scenario in Default Environment.

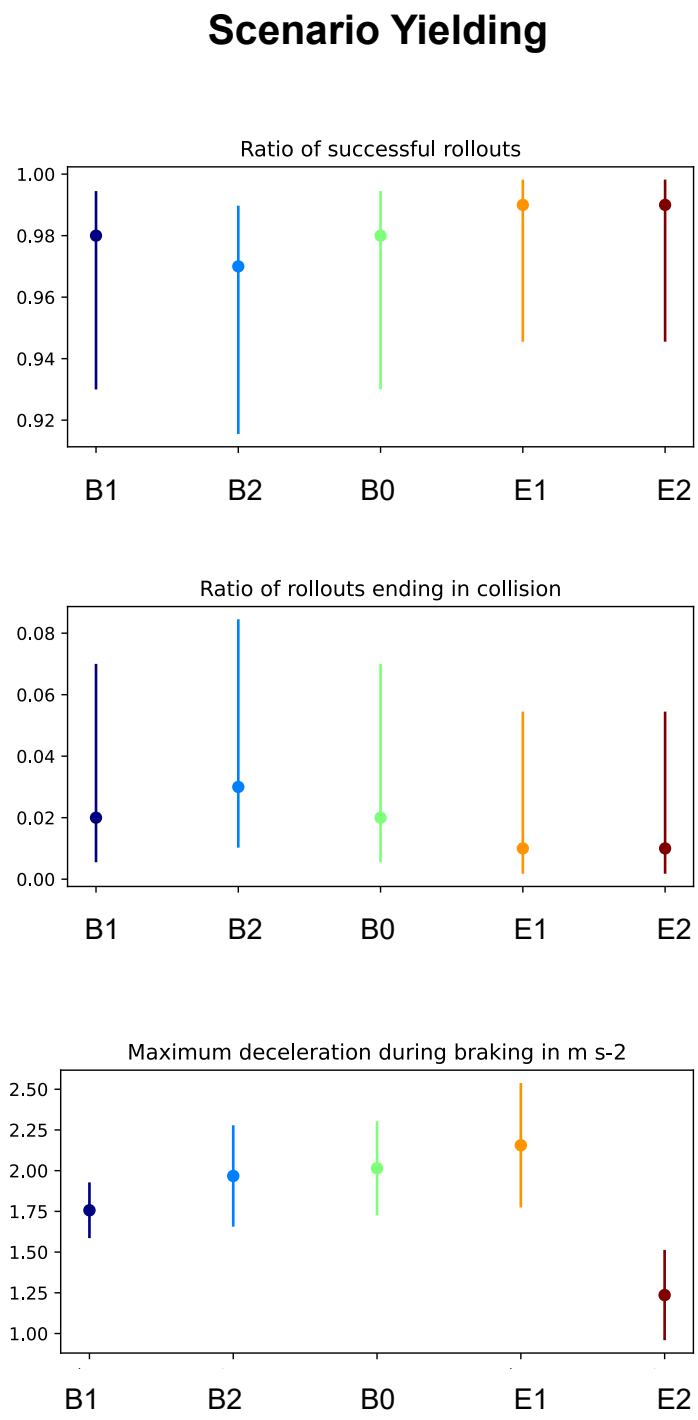


Figure 6.5: ConvNeXt Policy Evaluation with Yielding Scenario in Default Environment.

Evaluation in Occlusion Environment

As a next step the policy model with ConvNeXt backbone is evaluated using occlusion environment. Here, the environment uses the occlusion algorithm defined in section 4.3 to induce limited visibility zones in the roundabout. First the general test scenario is described in figure 6.6. Based on the results, policy trained with Bx datasets shows good performance in reaching the objective. While the policy with E1 and E2 dataset configurations fails resulting in reduced success ratio. The reason is that the policy with datasets that generated from extended IDM model gets stuck inside occlusion zone. Another observation is that except B2 all others shows comparable results with respect to maximum deceleration during braking KPI. In this case, B2 with baseline IDM rollouts and masked objects lists shows slow braking behavior.

With empty scenario as seen in figure 6.7, where there is no other vehicles in roundabout, all the policy configurations shows good performance except E1. Again in this case, the reason behind E1's failure is not due to collision or off-road driving but it gets stuck and stands still until timeout. This also affects its maximum deceleration value during braking. While, other policy versions retain a similar metric value during braking.

In yielding test case, baseline (B0) and E2 has good performance in terms of success. Here, all the models shows 1% of collision. And also there is no off road driving at any case. From the comfort metric also it is observed that the extended IDM with visibility layer config showcases more comfort than baseline in yielding. It is to be noted that the model trained with E2 can restore the same performance as its baseline model in all the scenarios, by strictly following its expert demonstrations. The results are described in figure 6.8.

Overall, in case of evaluation with occlusion environment, the collisions are reduced when compared to default environment. After observing the results from both the environment BX versions of datasets are not recommended in handling a driving behavior in occluded environment. As they have a strong domain gap that the policy faces during test time. There are also some failure case, that is the success ratio of E1 is low in all the test scenarios. This failure cases are elaborated in section 6.6.

Therefore, the policy model with ConvNeXt backbone, is shown to have reproduced the extended IDM's behavior is most of the cases when visibility layer configuration is added. It is also proven that the model gives more comfort in driving through an occluded environment. It justifies the previously stated fact that the policy with E2 dataset has enough information about the environment same as what its expert has during generating demonstrations for training dataset. Thus E2 seems to be a more realistic dataset for the occlusion environment.

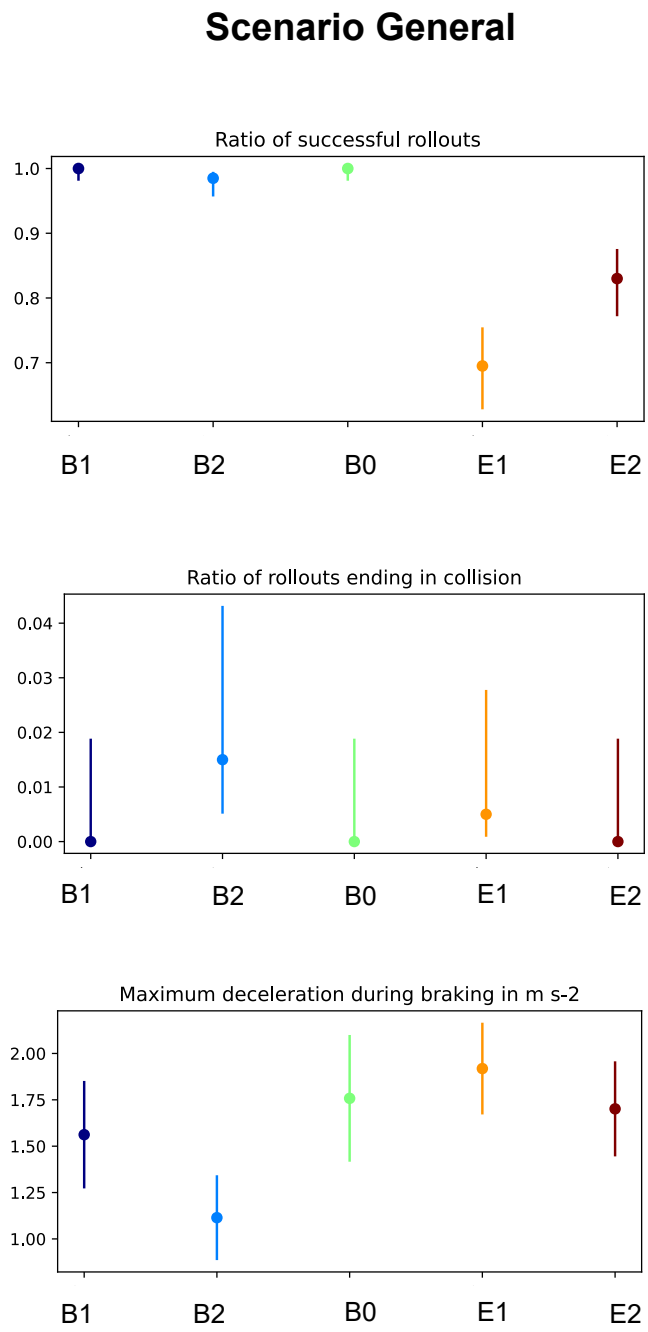


Figure 6.6: ConvNeXt Policy Evaluation with General Scenario in Occlusion Environment.

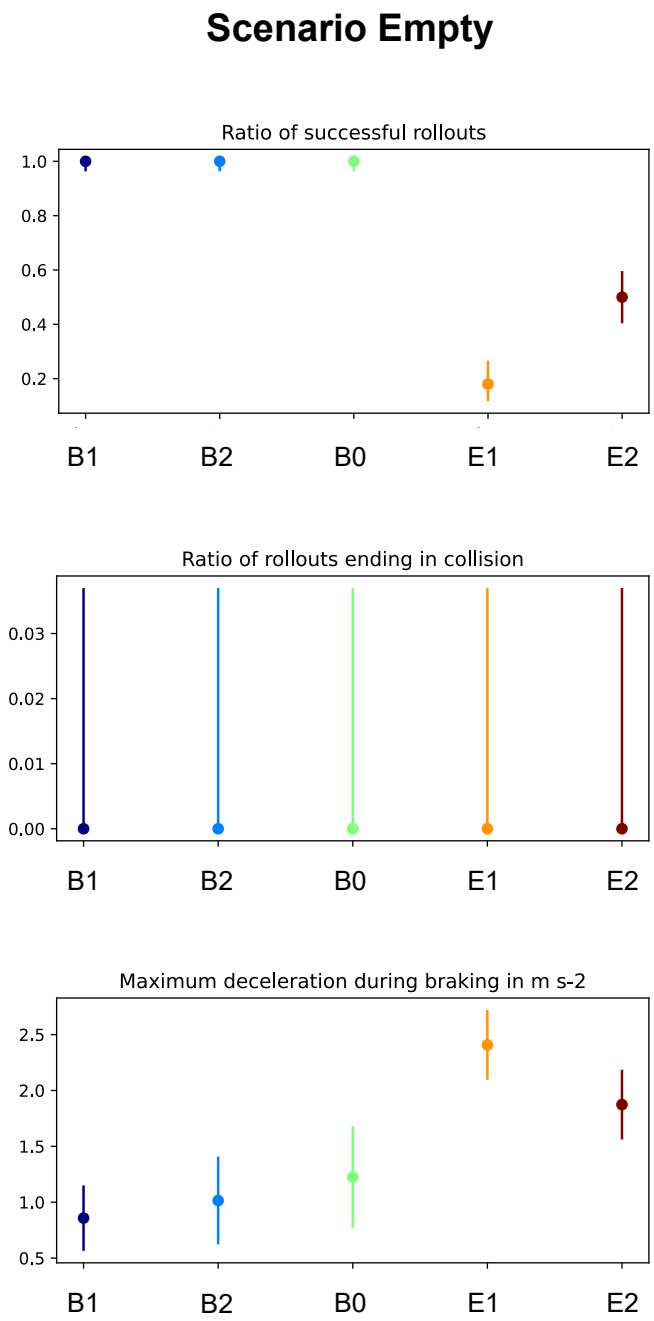


Figure 6.7: ConvNeXt Policy Evaluation with Empty Scenario in Occlusion Environment.

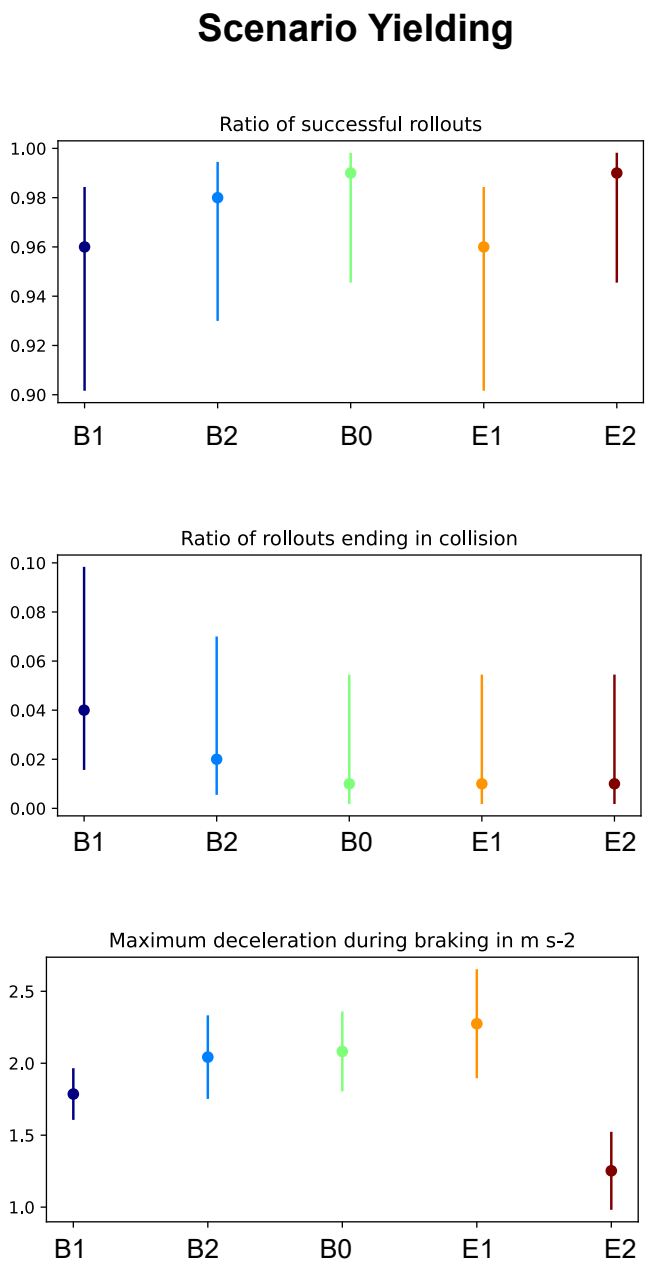


Figure 6.8: ConvNeXt Policy Evaluation with Yielding Scenario in Occlusion Environment.

6.5 Visual Evaluation

In qualitative examination, the enhanced dataset with visibility layer are to be examined closely. For this purpose, a yielding scenario is generated as a sample with the help of which the imitation learning-based behavior planning model is examined in more detail.

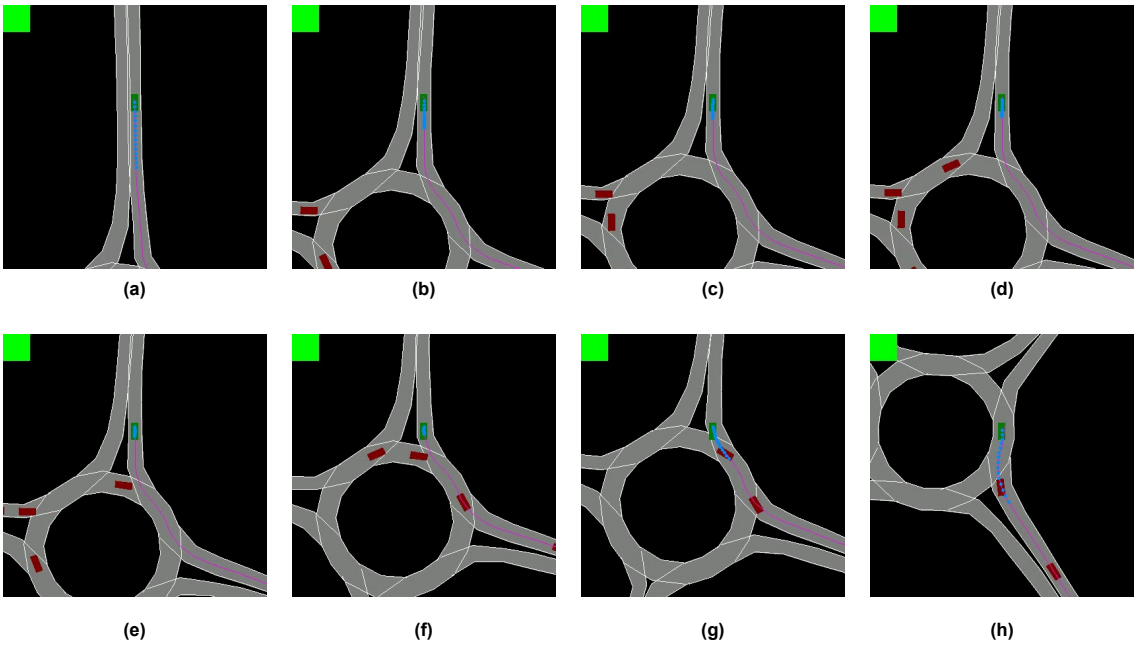


Figure 6.9: Visual Results of Policy Evaluation in Occlusion Environment.

Figure 6.9 shows the visual results of the ConvNeXt based policy model evaluation in occlusion environment. The images are generated from the grid video of the environment. A sample of frames are included in the figure to show the behavior of policy model. Figure 6.9(a) shows that the policy (referred as ego) is driving towards the roundabout junction with high speed. The pink line indicates the intended route for the ego to travel while, the blue dots are the predicted waypoints output of the model. In figure 6.9(b), the policy model approaching the roundabout is slowing down. It is very well observed from the changes in the waypoints pattern. This is mainly because the ego realized the presence of occlusion region towards its right that hinders its visibility to the environment. In case of figure 6.9(c), the ego vehicle decelerates having known about the limited visibility region. In the image (d), a third red vehicles comes out of the occlusion region which further causes the ego to yield in order to obey the static priority rules. In the image (e), the ego almost reaches a zero velocity state and stands still inside the trigger zone, until it gets a full visibility of the environment. In figure 6.9(f), ego gets a full visibility of the environment as the occlusion zone disappears. In this case, ego is out of the trigger zone and is waiting near the junction for the priority vehicles inside roundabout to pass-by. Finally, once the ego finds that its path is free to drive, it accelerates and drives through the roundabout by following the intended path. This scenario is illustrated in figures 6.9(g) and (h).

On the whole, it is inferred that the policy model trained with more realistic E2 dataset is able to handle the occlusion environment more carefully while retaining the driving comfort. In other words, the policy model is very much able to imitate the expert's behavior which is more similar to a human driver.

6.6 Failure Cases

The proposed methodology is not perfect, there are still collisions and timeout behavior observed during evaluations, especially with E1 version of the dataset. Some of the failure cases are analyzed and the causes are discussed here.

The model trained with extended IDM dataset (E1) gets stuck inside the trigger zone when it sees an occluded region present in the roundabout. Due to this the ego stands still a few meters away from the roundabout, as a result the simulation gets timeout. Such behavior from the ego can also be considered as a side effect of careful driving that is demonstrated by the extended IDM. The main root cause of the timeout issue is that there is a distribution mismatch in the data. The policy is facing a situation in which the ego vehicle is standing still for no reason even without any traffic. It is unable to handle this situation. This is because the expert model i.e., extended IDM which is used to generate the training data has never faced a scenario where the ego has to stop for no reason even when there is no other vehicle inside the roundabout. As a result of this distribution mismatch, the policy was unable to showcase a good performance. So it is recommended to induce such exceptional scenarios in the simulation environment and generate a new dataset. Such new datasets generated with necessary perturbations has enough knowledge to mitigate the issue of timeout. Additionally, steps has to be taken to mitigate the smaller percentage of collisions that is observed in certain cases during evaluations.

6.7 Summary

The results of evaluation experiments are summarized here. The scenarios used for closed evaluation of expert and policy models are detailed. The KPIs essential for quantitative evaluation of the models are explained in detailed. Especially, the new KPI metric on maximum deceleration during braking has served its purpose during detailed evaluation of expert and policy models. First, an ablation study is performed with three versions of expert models. As a result, the extended IDM is proved to have more comfort than the baseline IDM. Next, the policy models trained using five datasets are evaluated with default and occlusion environments. In both the environment, extended IDM based datasets shows a performances comparable to their corresponding baseline models in yielding scenario. Though there are certain performance drops in general and empty scenarios, they do not contribute much to the occlusion handling behavior. Thus, the policy with ConvNeXt is shown to have good performance by restoring the baseline model's performance even with the presence of occlusion zones in the roundabout environment. Overall, the comfort metric played a critical role in narrowing down to the good policy model that can handle the occlusions better. Finally, we observed the following the failure case of timeout behavior in the policy model. The root cause of this failure are analysed and possible solutions to resolve the failure case are discussed.

7 Conclusion

A detailed summary of the contributions to this thesis are discussed in this chapter. The conclusions drawn from this thesis are elaborated, which answers the research questions mentioned in section 3.5. In addition, suggestions of some possible future extensions of this topic are specified.

7.1 Summary

Handling occlusions caused by static and dynamic objects are crucial for a behavior planner to navigate through the roundabout environment. In this thesis work, the importance and complexity of the occluded regions in the roundabout driving scenario is elaborated. The benefits of end-to-end planning method in enabling future self-driving vehicles were discussed. Initially, a detailed overview about the existing behavior planning framework is explained. It is observed that the expert and policy models in the baseline has complete visibility to the roundabout in the simulation environment. This is because the underlying highway-env simulation which is used in generating training data and testing of the learned models has no occlusions present in it.

The thesis formulates a methodology to extend the baseline behavior planning framework to handle the roundabout environment with occlusions. As a part of the proposed methodology, an idea is devised to include the occlusion regions in the roundabout simulation environment. A strategic approach is used in narrowing down the possible regions that could be occluded when an ego vehicle tries to enter into the roundabout. It is also verified that by applying the newly devised algorithm, the simulation environment is induced with occluded regions near to the junctions inside the roundabout similar to a real world scenario. With this occlusion algorithm, the first research question [RQ1] is answered.

In a attempt to solve the second question [RQ2], the expert model in the baseline framework is evaluated to verify if it has any performance degradation. For this purpose, a new KPI called as maximum deceleration during braking is developed to evaluate the performance based on driving comfort of the expert. When evaluated with this new KPI, the baseline expert is found to produce a stronger braking behavior in the presence of occlusions which leads to a severe driving discomfort. In order to solve this problem, a set of enhancements are made to the baseline expert resulting in a new extended IDM. The extended IDM is enable to hallucinate about the spawned vehicles inside the active occlusion zones and it checks for the static priority rules for the hallucinated and reduced vehicles lists. Additionally, the extended IDM is also made to creep forward whenever it stands inside the trigger zone for no valid reason. With this enhancements the extended IDM is again validated using the new KPI metric and it is found to produce smoother braking behavior and gives a better comfort while driving through an environment with occlusions. Such proposed extensions solves the second research question.

The new devised comfort metric plays a predominant role in evaluating the performance of extended IDM. The answer to question [RQ3] is obtained by performing the ablation study on the expert model using KPIs. For this purpose, three IDM models are taken: baseline, limited visibility and extended IDM. Where, only the baseline IDM is tested with a fully visible environment. The results of the ablation study clearly proves that extended IDM has an upper hand in driving through the occluded roundabout with more comfort when compared to other expert models.

The baseline and extended IDM models are used to generate the dataset for training the planner model. As the policy model does not have enough knowledge to understand the reduced set of object lists present in the new dataset generated from extended IDM, an additional information is encoded to it. This information is provided by adding a feature channel consisting of the positions of visible and invisible regions of the roundabout environment. This new feature channel is rendered along with the grid images and serves as the input data to the policy model. Different versions of datasets are generated using configurations such as, with and without object filter and/or visibility layer. Eventually a new backbone with ConvNeXt is adapted to handle the enhanced features of the input dataset. An extensive training of the policy models are performed. The trained policies are then evaluated in both default and occlusion environments. The evaluations are performed using metrics like, success ratio, collision ratio and maximum deceleration during braking. The evaluation experiments are executed mainly with general, empty and yielding scenarios. As a result, a detailed analysis is made to understand the influence of training data in enabling the policy to imitate the expert. The evaluation results in the occlusion environment proved that the ConvNeXt-based architecture is able to restore the performance of the baseline framework when trained with a dataset from extended IDM along with encoded visibility information i.e., E2. The main reason behind this results is that, the E2 dataset is generated from the extended expert has limited visibility in the environment. This is more similar to the realistic datasets generated from real world occluded roundabout environment. Thus, in this case the extended IDM has limited field of view to the environment, whereas the policy model also has same limited visibility during test time. Therefore, there is no domain gap between the data used for training and the information available during testing of the planner. With this finding, the final research question [RQ4] is successfully answered. The resultant work is of course not free from failure cases. The failure cases observed during policy evaluations are due to time outs. The root cause of this time out problem is discussed and possible solutions are also specified.

7.2 Future Work

The possible improvements and extensions of the thesis work could be as follows:

- The trained behavior planning policy model faces some form of failure cases due to distribution mismatch with the data. A solution to resolve this failure case could be to synthesize relevant perturbations in the simulation environment and generate an enhanced dataset with more knowledge to train the behavior planning model.
- The complete training and evaluations are performed using simulation environment. The trained planner is directly suitable to be plugged into any existing perception and control system in the vehicle. But a dedicated vehicle testing has to be done for a more detailed analysis in order to verify if the policy can generalize well to handle the real world situations.

This also helps in testing robustness of the planner by exposing to the dynamically estimated occlusions. As the real-time occlusions can be bit more noisy, it is essential to verify if the behavior planner can handle such noise.

- In this thesis work, only random rotation is used as an augmentation technique. Though it achieves an impressive performance improvement, it cannot handle the bias in probability distributions. Hence a special data augmentation methods could be adapted to enhance policy behavior of imitating the expert in real world scenarios.
- When the proposed behavior planning framework in used in the vehicle system. There are chances where the real-time datasets like drone dataset are used for training. Such datasets are collected from an expert which as a complete visibility of the environment form top-down view. Here the expert has more privileged information. Whereas, the policy model trained with such dataset can be evaluated only by driving through the environment. In such case, the policy has a different field of view during test time. Therefore, an attempt could be made to propose an approach that can handle this domain difference between training and testing information in a generic manner.
- Further enhancements could be performed in the direction of proposing a method that can enable the prediction head to handle the invisible zones in the environment.

Bibliography

[1] J. L. Ba, J. R. Kiros, G. E. Hinton. "Layer normalization". In: *arXiv preprint arXiv:1607.06450* (2016) (cit. on p. 44).

[2] M. Bansal, A. Krizhevsky, A. Ogale. "Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst". In: *arXiv preprint arXiv:1812.03079* (2018) (cit. on pp. 14, 15, 23, 24, 27, 29, 49).

[3] H. Bey, M. Sackmann, A. Lange, J. Thielecke. "POMDP Planning at Roundabouts". In: *2021 IEEE Intelligent Vehicles Symposium Workshops (IV Workshops)*. IEEE. 2021, pp. 264–271 (cit. on pp. 14, 15).

[4] R. Bhattacharyya, S. Jung, L. A. Kruse, R. Senanayake, M. J. Kochenderfer. "A Hybrid Rule-Based and Data-Driven Approach to Driver Modeling Through Particle Filtering". In: *IEEE Transactions on Intelligent Transportation Systems* (2021) (cit. on p. 26).

[5] 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". In: *arXiv preprint arXiv:1604.07316* (2016) (cit. on p. 22).

[6] M. Bojarski, P. Yeres, A. Choromanska, K. Choromanski, B. Firner, L. Jackel, U. Muller. "Explaining how a deep neural network trained with end-to-end learning steers a car". In: *arXiv preprint arXiv:1704.07911* (2017) (cit. on p. 22).

[7] M. Bouton, A. Nakhaei, K. Fujimura, M. J. Kochenderfer. "Scalable decision making with sensor occlusions for autonomous driving". In: *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2018, pp. 2076–2081 (cit. on p. 21).

[8] S. Brechtel, T. Gindele, R. Dillmann. "Probabilistic decision-making under uncertainty for autonomous driving using continuous POMDPs". In: *17th international IEEE conference on intelligent transportation systems (ITSC)*. IEEE. 2014, pp. 392–399 (cit. on p. 21).

[9] M. Buehler, K. Iagnemma, S. Singh. *The DARPA urban challenge: autonomous vehicles in city traffic*. Vol. 56. springer, 2009 (cit. on p. 14).

[10] J. Chen, B. Yuan, M. Tomizuka. "Deep imitation learning for autonomous driving in generic urban scenarios with enhanced safety". In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2019, pp. 2884–2890 (cit. on pp. 22, 23, 26, 31).

[11] F. Codevilla, M. Müller, A. López, V. Koltun, 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 (cit. on pp. 14, 23).

[12] H. Cui, V. Radosavljevic, F.-C. Chou, T.-H. Lin, T. Nguyen, T.-K. Huang, J. Schneider, N. Djuric. "Multimodal trajectory predictions for autonomous driving using deep convolutional networks". In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 2090–2096 (cit. on p. 23).

[13] N. Djuric, V. Radosavljevic, H. Cui, T. Nguyen, F.-C. Chou, T.-H. Lin, J. Schneider. "Short-term motion prediction of traffic actors for autonomous driving using deep convolutional networks". In: (2018) (cit. on p. 23).

[14] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, V. Koltun. "CARLA: An open urban driving simulator". In: *Conference on robot learning*. PMLR. 2017, pp. 1–16 (cit. on pp. 14, 22, 57).

[15] J. Eggert, F. Damerow, S. Klingelschmitt. "The foresighted driver model". In: *2015 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2015, pp. 322–329 (cit. on p. 26).

[16] S. Falkner, A. Klein, F. Hutter. "BOHB: Robust and efficient hyperparameter optimization at scale". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 1437–1446 (cit. on pp. 34, 53).

[17] L. Fu, A. Yazici, U. Ozguner. "Route planning for OSU-ACT autonomous vehicle in DARPA urban challenge". In: *2008 IEEE Intelligent Vehicles Symposium*. IEEE. 2008, pp. 781–786 (cit. on p. 13).

[18] I. Georgievski, M. Aiello. "Automated planning for ubiquitous computing". In: *ACM Computing Surveys (CSUR)* 49.4 (2016), pp. 1–46 (cit. on p. 22).

[19] I. Georgievski, M. Aiello. "HTN planning: Overview, comparison, and beyond". In: *Artificial Intelligence* 222 (2015), pp. 124–156 (cit. on p. 22).

[20] K. Gokcesu, H. Gokcesu. "Generalized huber loss for robust learning and its efficient minimization for a robust statistics". In: *arXiv preprint arXiv:2108.12627* (2021) (cit. on p. 30).

[21] P. Hart, A. Knoll. "Counterfactual Policy Evaluation for Decision-Making in Autonomous Driving". In: *arXiv preprint arXiv:2003.11919* (2020) (cit. on p. 31).

[22] K. He, X. Zhang, S. Ren, J. Sun. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778 (cit. on pp. 28, 33, 43).

[23] K. He, X. Zhang, S. Ren, J. Sun. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778 (cit. on p. 44).

[24] S. Hecker, D. Dai, L. Van Gool. "Learning driving models with a surround-view camera system and a route planner". In: *ArXiv e-prints* (2018) (cit. on p. 22).

[25] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, D. Meger. "Deep reinforcement learning that matters". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 32. 1. 2018 (cit. on p. 53).

[26] D. Hendrycks, K. Gimpel. "Gaussian error linear units (gelus)". In: *arXiv preprint arXiv:1606.08415* (2016) (cit. on p. 44).

[27] R. Huang, J. Pedoeem, C. Chen. "YOLO-LITE: a real-time object detection algorithm optimized for non-GPU computers". In: *2018 IEEE International Conference on Big Data (Big Data)*. IEEE. 2018, pp. 2503–2510 (cit. on pp. 29–31, 33).

[28] C. Hubmann, N. Quetschlich, J. Schulz, J. Bernhard, D. Althoff, C. Stiller. "A pomdp maneuver planner for occlusions in urban scenarios". In: *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2019, pp. 2172–2179 (cit. on p. 21).

[29] H. Ide, T. Kurita. "Improvement of learning for CNN with ReLU activation by sparse regularization". In: *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2017, pp. 2684–2691 (cit. on p. 44).

[30] M. Ilievski, S. Sedwards, A. Gaurav, A. Balakrishnan, A. Sarkar, J. Lee, F. Bouchard, R. De Iaco, K. Czarnecki. "Design space of behaviour planning for autonomous driving". In: *arXiv preprint arXiv:1908.07931* (2019) (cit. on p. 18).

[31] S. Ioffe. "Batch renormalization: Towards reducing minibatch dependence in batch-normalized models". In: *Advances in neural information processing systems* 30 (2017) (cit. on p. 44).

[32] D. Isele, A. Cosgun, K. Fujimura. "Analyzing knowledge transfer in deep q-networks for autonomously handling multiple intersections". In: *arXiv preprint arXiv:1705.01197* (2017) (cit. on p. 22).

[33] D. Isele, R. Rahimi, A. Cosgun, K. Subramanian, K. Fujimura. "Navigating occluded intersections with autonomous vehicles using deep reinforcement learning". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 2034–2039 (cit. on pp. 14, 22, 31).

[34] D. Isele, R. Rahimi, A. Cosgun, K. Subramanian, K. Fujimura. "Navigating occluded intersections with autonomous vehicles using deep reinforcement learning". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 2034–2039 (cit. on p. 22).

[35] A. Jain, L. Del Pero, H. Grimmett, P. Ondruska. "Autonomy 2.0: Why is self-driving always 5 years away?" In: *arXiv preprint arXiv:2107.08142* (2021) (cit. on p. 13).

[36] K. Jamieson, A. Talwalkar. "Non-stochastic best arm identification and hyperparameter optimization". In: *Artificial intelligence and statistics*. PMLR. 2016, pp. 240–248 (cit. on p. 53).

[37] E. Kaldeli, A. Lazovik, M. Aiello. "Domain-independent planning for services in uncertain and dynamic environments". In: *Artificial Intelligence* 236 (2016), pp. 30–64 (cit. on p. 22).

[38] A. Klein, S. Falkner, S. Bartels, P. Hennig, F. Hutter. "Fast bayesian hyperparameter optimization on large datasets". In: *Electronic Journal of Statistics* 11.2 (2017), pp. 4945–4968 (cit. on p. 53).

[39] M. Klimke, B. Völz, M. Buchholz. "Cooperative Behavioral Planning for Automated Driving using Graph Neural Networks". In: *arXiv preprint arXiv:2202.11376* (2022) (cit. on p. 27).

[40] P. Kothari, C. Perone, L. Bergamini, A. Alahi, P. Ondruska. "Drivergym: Democratising reinforcement learning for autonomous driving". In: *arXiv preprint arXiv:2111.06889* (2021) (cit. on p. 31).

[41] R. Krajewski, J. Bock, L. Kloeker, L. Eckstein. "The highD Dataset: A Drone Dataset of Naturalistic Vehicle Trajectories on German Highways for Validation of Highly Automated Driving Systems". In: *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. 2018, pp. 2118–2125. DOI: 10.1109/ITSC.2018.8569552 (cit. on pp. 24, 48).

[42] A. Kuefler, J. Morton, T. Wheeler, M. Kochenderfer. "Imitating driver behavior with generative adversarial networks". In: *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2017, pp. 204–211 (cit. on p. 22).

[43] E. Leurent. *An Environment for Autonomous Driving Decision-Making*. https://github.com/eleurent/highway-env. 2018 (cit. on pp. 24, 47, 48).

[44] E. Leurent. *An Environment for Autonomous Driving Decision-Making*. https://github.com/eleurent/highway-env. 2018 (cit. on p. 25).

[45] L. Li, K. G. Jamieson, G. DeSalvo, A. Rostamizadeh, A. Talwalkar. "Hyperband: Bandit-based configuration evaluation for hyperparameter optimization". In: *ICLR (Poster)*. 2017 (cit. on p. 53).

[46] Z. Li, S. Arora. "An exponential learning rate schedule for deep learning". In: *arXiv preprint arXiv:1910.07454* (2019) (cit. on p. 50).

[47] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár. "Focal loss for dense object detection". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2980–2988 (cit. on p. 30).

[48] X. Lin, J. Zhang, J. Shang, Y. Wang, H. Yu, X. Zhang. "Decision making through occluded intersections for autonomous driving". In: *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE. 2019, pp. 2449–2455 (cit. on p. 13).

[49] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, S. Xie. "A ConvNet for the 2020s". In: *arXiv preprint arXiv:2201.03545* (2022) (cit. on pp. 43, 44).

[50] G. Melis, C. Dyer, P. Blunsom. "On the state of the art of evaluation in neural language models". In: *arXiv preprint arXiv:1707.05589* (2017) (cit. on p. 53).

[51] J. Müller, J. Strohbeck, M. Herrmann, M. Buchholz. "Motion planning for connected automated vehicles at occluded intersections with infrastructure sensors". In: *IEEE Transactions on Intelligent Transportation Systems* (2022) (cit. on p. 14).

[52] M. Müller, A. Dosovitskiy, B. Ghanem, V. Koltun. "Driving policy transfer via modularity and abstraction". In: *arXiv preprint arXiv:1804.09364* (2018) (cit. on p. 23).

[53] V. Nair, G. E. Hinton. "Rectified linear units improve restricted boltzmann machines". In: *Icml*. 2010 (cit. on pp. 29, 44).

[54] J. E. Naranjo, F. Jiménez, O. Gómez, J. G. Zato. "Low level control layer definition for autonomous vehicles based on fuzzy logic". In: *Intelligent Automation & Soft Computing* 18.4 (2012), pp. 333–348 (cit. on p. 28).

[55] U. Ozguner, T. Acarman, K. A. Redmill. *Autonomous ground vehicles*. Artech House, 2011 (cit. on p. 14).

[56] S. D. Pendleton, H. Andersen, X. Du, X. Shen, M. Meghjani, Y. H. Eng, D. Rus, M. H. Ang. "Perception, planning, control, and coordination for autonomous vehicles". In: *Machines* 5.1 (2017), p. 6 (cit. on pp. 17, 18, 22).

[57] A. Philipp. "Perception and Prediction of Urban Traffic Scenarios for Autonomous Driving". PhD thesis. 2021 (cit. on p. 34).

[58] D. A. Pomerleau. "Alvinn: An autonomous land vehicle in a neural network". In: *Advances in neural information processing systems* 1 (1988) (cit. on p. 22).

[59] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen. "Mobilenetv2: Inverted residuals and linear bottlenecks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 4510–4520 (cit. on p. 43).

[60] A. Sauer, N. Savinov, A. Geiger. "Conditional affordance learning for driving in urban environments". In: *Conference on Robot Learning*. PMLR. 2018, pp. 237–252 (cit. on p. 14).

[61] A. Sauer, N. Savinov, A. Geiger. “Conditional affordance learning for driving in urban environments”. In: *Conference on Robot Learning*. PMLR. 2018, pp. 237–252 (cit. on p. 23).

[62] J. Schulz, C. Hubmann, N. Morin, J. Löchner, D. Burschka. “Learning interaction-aware probabilistic driver behavior models from urban scenarios”. In: *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2019, pp. 1326–1333 (cit. on p. 15).

[63] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, N. De Freitas. “Taking the human out of the loop: A review of Bayesian optimization”. In: *Proceedings of the IEEE* 104.1 (2015), pp. 148–175 (cit. on p. 53).

[64] K. Simonyan, A. Zisserman. “Very deep convolutional networks for large-scale image recognition”. In: *arXiv preprint arXiv:1409.1556* (2014) (cit. on p. 44).

[65] L. Sun, W. Zhan, C.-Y. Chan, M. Tomizuka. “Behavior planning of autonomous cars with social perception”. In: *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2019, pp. 207–213 (cit. on p. 13).

[66] M. Tan, Q. Le. “Efficientnet: Rethinking model scaling for convolutional neural networks”. In: *International conference on machine learning*. PMLR. 2019, pp. 6105–6114 (cit. on p. 43).

[67] Y. Tang. “Towards learning multi-agent negotiations via self-play”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*. 2019, pp. 0–0 (cit. on pp. 15, 27).

[68] S. Thornton. “Autonomous vehicle speed control for safe navigation of occluded pedestrian crosswalk”. In: *arXiv preprint arXiv:1802.06314* (2018) (cit. on p. 21).

[69] M. Treiber, A. Hennecke, D. Helbing. “Congested traffic states in empirical observations and microscopic simulations”. In: *Physical review E* 62.2 (2000), p. 1805 (cit. on p. 26).

[70] S. Xie, R. Girshick, P. Dollár, Z. Tu, K. He. “Aggregated residual transformations for deep neural networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1492–1500 (cit. on p. 44).

[71] H. Xu, Y. Gao, F. Yu, T. Darrell. “End-to-end learning of driving models from large-scale video datasets”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2174–2182 (cit. on p. 22).

[72] M.-Y. Yu, R. Vasudevan, M. Johnson-Roberson. “Occlusion-aware risk assessment for autonomous driving in urban environments”. In: *IEEE Robotics and Automation Letters* 4.2 (2019), pp. 2235–2241 (cit. on p. 14).

[73] J. Zhou, R. Wang, X. Liu, Y. Jiang, S. Jiang, J. Tao, J. Miao, S. Song. “Exploring imitation learning for autonomous driving with feedback synthesizer and differentiable rasterization”. In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2021, pp. 1450–1457 (cit. on p. 50).

[74] Z. Zhu, H. Zhao. “A survey of deep rl and il for autonomous driving policy learning”. In: *IEEE Transactions on Intelligent Transportation Systems* (2021) (cit. on pp. 19, 23).

Declaration

I hereby declare that the work presented in this thesis is entirely my own and that I did not use any other sources and references than the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations. Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before. The electronic copy is consistent with all submitted copies.

Stuttgart, 17.05.2022

place, date, signature