WiseBench: A Motion Planning Benchmarking Framework for Autonomous Vehicles

by

Marko Ilievski

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Masters in Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2020

Authors Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Rapid advances in every sphere of autonomous driving technology have intensified the need to be able to benchmark and compare different approaches. While many benchmarking tools tailored to different sub-systems of an autonomous vehicle, such as perception, already exist, certain aspects of autonomous driving still lack the necessary depth and diversity of coverage in suitable benchmarking approaches – autonomous vehicle motion planning is one such aspect. While motion planning benchmarking tools are abundant in the robotics community in general, they largely tend to lack the specificity and scope required to rigorously compare algorithms that are tailored to the autonomous vehicle domain. Furthermore, approaches that *are* targeted at autonomous vehicle motion planning are generally either not sensitive enough to distinguish subtle differences between different approaches, or not able to scale across problems and operational design domains of varying complexity. This work aims to address these issues by proposing *WiseBench*, an autonomous vehicle motion planning benchmark framework aimed at comprehensively uncovering fine and coarse-grained differences in motion planners across a wide range of operational design domains.

WiseBench outlines a robust set of requirements for a suitable autonomous vehicle motion planner. These include *simulation requirements* that determine the environmental representation and physics models used by the simulator, *scenario-suite requirements* that govern the type and complexity of interactions with the environment and other traffic agents, and *comparison metrics requirements* that are geared towards distinguishing the behavioral capabilities and decision making processes of different motion planners. *WiseBench* is implemented using a carefully crafted set of scenarios and robust comparison metrics that operate within an in-house simulation environment, all of which satisfy these requirements. The benchmark proved to be successful in comparing and contrasting two different autonomous vehicle motion planners, and was shown to be an effective measure of passenger comfort and safety in a real-life experiment. The main contributions of our work on *WiseBench* thus include: a scenario creation methodology for the representative scenario suite, a comparison methodology to evaluate different motion planning algorithms, and a proof-of-concept implementation of the *WiseBench* framework as a whole.

Acknowledgements

I would like to thank my supervisor, Prof. Krzysztof Czarnecki for his tremendous support and understanding throughout this long and difficult journey. Thank you, Krzysztof, for providing me with constant feedback and continued guidance on this project along the road. Thank you for not losing faith in me even when the road ahead looked bleak.

A big thank you to many members of the WISE lab for your collaboration. In particular, thank you to Frederic Bouchard, Michal Antkiewicz, Maximilian Kahn, Rodrigo Queiroz, and many more too numerous to name. Without your help and amazing work on the simulation environment, and many other aspects of the autonomy stack, this thesis and many of the experiments within in it would have been impossible.

Thank you, also, to my parents Zaneta and Goran and my little sister Marija, who have been so supportive despite the numerous challenges that have arisen. Thank you for always being by my side even when not physically present.

Finally, a *massive* thanks to Nicole Dillen. Without you, and your continued and unwavering support and help in every aspect, I would not be where I am today. Thank you for everything.

Dedication

This is dedicated to my parents, Zaneta Ilievska and Goran Ilievski, who sacrificed all they could to make me who I am today.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Autonomous Vehicles (AVs), in recent years, have made incredible progress towards achieving full autonomy. The Society of Automotive Engineers (SAE) have described 5 different Levels of autonomy ranging from driver assistance and partial automation at Levels 1 and 2, to high and full automation at Levels 4 and 5. Rapid developments from SAE Level 3 (conditional automation in the presence of a human safety driver) towards Levels 4 and 5, have been accompanied by equally swift expansions in the environmental complexities that such vehicles can robustly handle. Waymo, for example, has already begun testing its Level 4 passenger fleet on human riders with no safety driver present [123]. Such advancements were largely fueled by the DARPA Grand Challenge in 2007 [27], in which 11 AV teams competed against each other to complete an urban area course within a specified time limit, while simultaneously interacting with each other and obeying the rules of the road. The complex nature of the challenge tested, primarily, the intelligent decision making capabilities of the competing AVs, and kickstarted the race to full autonomy.

While the DARPA challenge served as an initial benchmark for AV systems, it failed to scale with respect to the complexity required for true autonomous driving: the results are neither reproducible nor testable, and testing extreme edge cases of the ODD is not feasible in the real world due to the risk of accidents and physical damages. A scientific approach is instead required to formulate a robust set of benchmarks capable of handling diverse environments and complexities. Such benchmarks already exist for many sub-systems of an AV. Perception, in particular, has seen an explosion in the number of available datasets (such as KITTI [44], Cityscapes [30], BDD100K [129], Apolloscape [50], and Waymo's Open Dataset [115]) that facilitate benchmark comparisons between existing and new algorithms. AV motion planning, on the other hand, does not share this diversity, as it requires extremely large datasets in order to properly train and test algorithms.

The robotics community, in general, has presented several methods to benchmark motion planning algorithms across a variety of domains: while some of these are more straightforward and focus on general robot movement from one location to another [88, 69], others are more specific in nature – such as those targeted towards grasping-related problems [86]. Many of these benchmarks have been run in MoveIt! [29], a benchmarking framework for decomposing path generation properties of motion planning algorithms, which has been integrated into the widely-used ROS (Robot Operating System) framework.

However, while existing benchmarks may suffice for trajectory generation problems, these benchmarks neither account for interactions with other agents, nor do they require any knowledge about the rules of the road. As such, they are not easily transferable to the domain of automated driving systems (ADS) simply because they do not cover the full complexity and range of motion planning tasks required for the safe operation of AVs. That said, however, there have been recent attempts to adapt existing benchmarking approaches from other domains to the specific problem of AV motion planning: CommonRoad [12], the 2019 (and 2020) CARLA Autonomous Driving (AD) Challenge [7], and Voyage Deepdrive [6] are examples of three such efforts.

CommonRoad, a set of "composable benchmarks for motion planning on roads" [12], aims to solve the benchmarking problem through a collection of scenarios, vehicle models, and cost functions that are representative of the AV motion planning problem. However, due to the rudimentary nature of the simulator used, and a lack of suitable comparison metrics, this set of benchmarks is not adequate to cover the full complexity of AV motion planning. The CARLA AD Challenge, on the other hand, *does* provide a more comprehensive set of evaluation metrics, and, in addition, aims to benchmark algorithms across a variety of realistic traffic, environmental, and weather conditions. It is, however, limited by the low number of scenarios used as well as by reproducibility issues caused by the random nature of its traffic agents. Finally, Voyage Deepdrive provides a high fidelity simulation environment along with evaluation metrics but, at the time of writing, contains only a single scenario available for benchmarking.

Thus, while most attempts at benchmarking AV motion planners have taken a step in the right direction, they tend to suffer from a lack of breadth and depth in the combination of both scenarios employed and comparison metrics used to evaluate different algorithms. Hence, they typically do not cover the entire scope of AV motion planning. A strong benchmark framework for this problem should thus consist of a *high fidelity simulation environment* that can simulate real-world driving as closely as possible using a *representative suite of scenarios*, while also providing *robust comparison metrics* that can be used to evaluate different algorithms. These set of requirements are further elaborated in Chapter 3.

To address the issues prevalent in state-of-the-art AV motion planning benchmarks, this thesis proposes *WiseBench*, a motion planning benchmark framework targeted at AVs, that incorporates a robust set of simulation, scenario suite, and comparison metrics requirements. It describes, in detail, the simulation environment used to implement our framework, and presents a method to select and represent a diverse set of scenarios with scalable difficulty levels. It further provides a concrete design for our benchmark interface, along with a robust set of metrics that can appropriately evaluate performance across a wide range of motion planning algorithms.

The following 8 chapters propose, describe, and evaluate *WiseBench*: Chapter 2 serves as a background for the reader and reviews existing motion planning algorithms and benchmarks prevalent throughout robotics in general, as well as within the AV domain. Chapter 3 proposes a set of broad as well as specific requirements necessary to realize a successful AV motion planning benchmark, while Chapter 4 describes in detail the specifics of the simulation environment that we used to realize the WiseBench framework. Chapter 5 outlines the methodology and design decisions used to create hand-made scenario features for a given (or shared) operational design domain (ODD), and goes on to discuss the specific use case of the Autonomoose, an autonomous driving research platform at the University of Waterloo. Chapter 6 discusses the comparison metrics used to compare different motion planning algorithms and identify their behavioral and driving characteristic differences.

Chapters 7 through 9 focus on evaluating the proposed benchmark. Chapter 7 presents the results obtained over two distinct cases: one in which in which driving parameter differences were introduced, and another in which misbehavior was introduced on the part of the ego (the vehicle equipped with the ADS); the 2 cases were considered to provide insight into how the proposed scenarios and comparison methods work together to highlight differences between algorithms. Meanwhile, Chapter 8 describes a real world experiment [34] that uses a subset of these comparison metrics along with a set of simple scenarios to further evaluate the feasibility of applying the metrics to closed course autonomous driving environments. Finally, Chapter 9 provides an in-depth discussion of the results obtained, and delineates possible directions for future work.

The thesis highlights three major contributions, namely:

1. A scenario creation methodology for the scenario suite, that allows benchmarking in any given ODD (when evaluating a single AV motion planner) or shared ODD (when comparing multiple planners).

2. A comparison methodology that can be used for benchmarking AV motion planning algorithms – and for comparing 2 or more algorithms – by evaluating (or identifying

differences in) behavioral capabilities and decision making trade-offs employed across the given (or shared) ODD.

3. An implementation of this approach and its application to the Autonomoose project, using the simulation environment created by the University of Waterloo's WISE laboratory to compare differences in driving characteristics and evaluate behavioral correctness.

Chapter 2

Related Work

This chapter provides an overview of the various approaches towards motion planning for autonomous driving in order to achieve a sufficient level of context and understanding of the problem space we will be evaluating. In addition, we capture the start-of-the art of available motion planning benchmarks both in general robotics domains and, more specifically, autonomous driving.

2.1 Motion Planning for Autonomous Vehicles

The motion planning problem for autonomous vehicles (AVs) can be summarized as: the *generation* of a *trajectory* (path and velocity), towards a *goal* within a continuously *evolving environment* constrained by **safety** of all traffic agents, **comfort** of passengers, and continued **progression** towards a goal. Additionally, all generated trajectories must meet a constraint of feasibility: meaning they can be feasibly executed by the vehicle hardware while taking into account all of the vehicle's dynamics. In this thesis, we refer to the vehicle equipped with an automated driving system (ADS) as the "ego vehicle".

Due to the complexity of the motion planning problem, the motion planning task is typically decomposed into three sub-problems:

- **Route Planning**, the highest level decisions which plan a sequence of lanes from the current position of the ego vehicle, to its final destination
- **Decision Making**, which takes into consideration the planned route and the environment around the ego vehicle to devise a sequence of high level maneuvers to navigate through the environment.

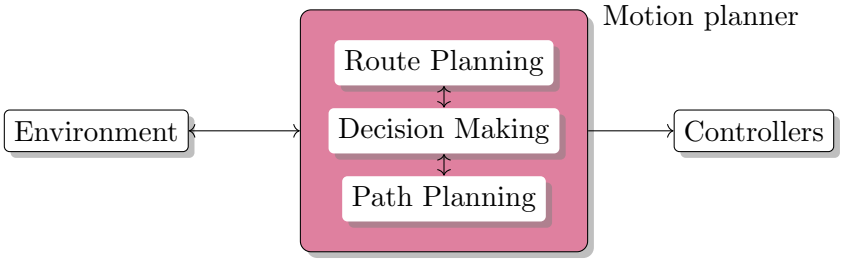


Figure 2.1: General architecture of motion planning. White boxes denote the scope of the current work.

- **Path Planning,** the lowest-level planner which, given the route and the set of maneuvers, generates a safe and smooth trajectory for the AV.

The architecture used to complete these sub-problems within the motion planning literature fall within two categories: A logically *Separated* Architecture, and fully *Integrated* Architecture.

Separated architectures are ones where there is logical separation between each sub-problem. Early approaches [120, 14] mainly followed a top-down architecture design, where high level decisions constrain the problem for low-level decisions; the route planning module constrains the problem for the decision making module which constrains the problem for the path planning module as illustrated by Figure 2.1. Many early approaches provided single decisions which, in some situations, constrain the problem into infeasibility. Alternatively, due to the separated nature of the modules incurring significant re-computation, Wei et al. [124] propose a solution by inverting the planning problem. Wei et al. first planned a set of possible trajectories which were then individually evaluated by the Decision Making module. While this approach still suffered from re-computation, it solved the issue of over-constraining the problem into infeasibility.

Integrated architectures, on the other hand, do not enforce hard boundaries between the sub-problems. Baidu's open source autonomy stack, Apollo, [39] has demonstrated this by encapsulating the motion planning problem into a single pipeline. This encapsulation of the motion planning problem into a single pipeline allows maneuver selection to be performed implicitly during the planning process. This style of architecture has become significantly more popular with the emergence of learning methods. One method of solving the learning problem is to solve the entire motion planning problem [68, 98], and perhaps even the entire autonomous driving problem, in an end-to-end fashion [126, 25, 37]. To achieve this, the learned system must rely on rewards or cost functions to inform its behavioral decision-making; this can be done either with [98] or without the use of explicitly defined maneuvers

[16]. While these approaches show great promise, they rely heavily on large labelled data sets and sophisticated simulation environments, both of which may not be readily available to academic researchers.

Regardless of the specific architectural decision, in this thesis we will look at each sub-problem of the motion planning task individually; each task will be discussed independently so as to provide greater detail in an organized manner.

2.1.1 Route Planning

Route planning is the process of selecting a sequence of lanes through the road network, starting from the ego vehicle's current location and passing through all requested points towards the final destination.

The topological relationships of different lanes often result in road networks being represented as weighted directed graphs [96]. The weights of the network will vary largely depending on the use-case. Common factors which influence the weight of a given road include a combination of distances, the user's preferences, and a priori assumptions about road conditions including safety assumptions (for instance, some roads may be less prone to accidents then others).

The nature of the problem (i.e., finding the most optimal path to the destination) and the underlying graph representation of the road network allow route planning approaches to employ traditional graph search algorithms such as Dijkstra and A* Search to find the best route. Other more optimized approaches, such as pre-computed network optimization algorithms, have also been utilized to solve this task [17].

2.1.2 Decision Making

One of the major sub-problems of the motion planning process is the selection of discrete, high-level maneuvers – or decisions – to navigate the ego vehicle through the complex environment. The set of maneuvers must navigate all road-rules, and handle reactions and interactions with static and dynamic traffic agents. High-level maneuvers might include such basic manoeuvres as *speed up*, *slow down* and *come to a stop*, as well as more complex and detailed maneuvers like *overtake a parked vehicle* .

Ilievski et. al. [52] have broadly classified prior approaches in decision making for AVs into one of two approaches: classical programmed decision makers, and learning based approaches.

Programming Based Approaches

A significant amount of work has been done on classical methods, and have largely originated from the early 2007 DARPA challenge [120, 14]. Programming often relies on either a decomposition of the problem into finite state machines (FSMs) [80, 84, 70], a set of carefully hand-crafted rules stored in rule-engines [13], or optimization problems relying on techniques such as partially observable Markov decision process (POMDP) solvers [79, 128, 43, 51, 26]. Programmed logic systems have the advantage of interpretability which allows traditional software engineering methods, such as inspections and walkthroughs, to uncover problems in implementation. However, such methods have several large faults, namely, (1) the inability to adapt to unexpected situations, (2) over generalization, which often leads to overly conservative behavior, (3) hard to (manually) tune parameters, (4) constraint monitoring and adaptation of hyper parameters based on environmental changes, and finally (5) unsolvability even in small Operational Design Domains (ODDs).

Learning Based Approaches

Learned methods are an alternative to classically programmed methods. These can be further subdivided into algorithms which learn by example, and algorithms that learn by interaction, with some approaches relying on both strategies.

Learning by Example is a set of algorithms which use recorded and labeled data from human driver demonstrations to learn how to drive in various situations. Many subvariations of such algorithms exist such as end-to-end techniques [102, 90, 25], imitation learning [134, 106, 114], and inverse reinforcement learning [68]. While all the aforementioned approaches have their own individual strengths and weaknesses, most of the major strengths and weakness are shared by all. The major strength of such learning techniques is their ability to accurately mirror human-style driving, which encodes both the rules of the road as well as human driving conventions. However, the major weaknesses include: (1) requiring a huge amount of highly accurately gathered and labeled training data, (2) the difficult challenge of postulating highly accurate reward functions able to capture all scenarios in the ODD, and (3) hard to train edge cases such as driving scenarios that are difficult to record.

Learning by Interaction, on the other hand, is a set of algorithms that learn the task of driving by attempting different strategies (policies) in a simulation and optimizing for the most successful policy through a set of reward functions. Such algorithms are also

known as Reinforcement Learning (RL). While many models of RL have been tried to tackle the problem of self driving (traditional RL [87, 130], model-based RL [56, 116], hierarchical RL [124, 111, 73], and neural networks techniques [62, 75, 76, 98, 108]), the underlying concept relies on the same principles. The agent receives feedback from the simulated environment in the form of a reward and adjusts its behaviour to maximize the expected long term future reward. Although such approaches have seen some success in academic settings [116], they often don't work as well when transferred to real vehicles. This is due to the over-reliance on simulators which tend to be more simplistic and unable to capture all the complexities of the real world; while some fail at the ability to transfer the correct vehicle dynamic, others are unable to model other agents' behavior (such as the unpredictability of human driving behavior). Finally, some simulators do not take into account non-driving environmental factors, such as weather, road conditions, construction, and dynamically shifting map networks.

Learning from Example and Interaction is the final category of learning methods which has seen the least amount of exploration within academia as a whole. Imitation Learning is an example of this set of methods. For example, in [111] an initial behaviour is obtained through imitation learning (learning from example), but improvements are made through interaction with a simulated environment (learning from interaction). Biologically inspired methods such as Hebbian learning and genetic algorithms could also be combined with RL (learning from interaction) to learn a driving behaviour [93, 94, 65]. For these methods to be effective, it is imperative to appropriately design the fitness function and spend the requisite amount of time needed for parameter tuning which, many times, is the downfall of such approaches.

2.1.3 Path Planning

Path planning is the final and lowest level sub-problem in the motion planning task. It is responsible for using the planned route and high-level decisions from the previous two steps to generate a final trajectory for the ego vehicle. This trajectory includes the path that the vehicle should take along with the velocity that it should follow along that path. While many categorisations of this domain exists [96, 47] one of the most popular of these categorizations groups path planning into the following three categories: sampling based approaches, interpolating curve approaches, and finally grid-based search approaches [47].

Sampling Based

Sampling based motion planners continuously sample random points within the state space towards the ego's final goal. These points are concatenated and smoothed to form the full trajectory of the ego [**?**, 47]. The most common sampling method is the Rapidly-exploring Random Tree (RRT) [41] approach, which has been explored in the greater robotics domain in contexts ranging from navigational tasks all the way to flight path planning [71, 59, 58]. This method has relatively low computational cost and as such was heavily utilised in early AV development through a wide range of implementations[**?**]. Most notably, Karaman et. al. showed that the original RRT algorithm sub-optimally converges to a final path, and proposed an asymptotically optimal adaptation which they named *RRT**[59]. *RRT** has since become one of the most popular techniques in path planning due to its versatility[97, 8, 83, 122].

Sampling methods, however, are not without their disadvantages. The planned trajectories that these methods produce are sub-optimal and far jerkier than other approaches [**?**]. At the same time, the primary advantage in using them is that they perform significantly better in high dimensional domains due to their low computational costs and guaranteed ability to find a path given enough time.

Interpolating Curve Approaches

AVs have an internal map representation of the environment around them. Often these maps include a lane center which the vehicle should try to closely follow. Interpolation approaches use these lane center points to generate a new and smoother path, ensuring trajectory continuity. Besides providing a continuous trajectory, these approaches take into consideration the vehicle constraints as well as the dynamic environment to generate an accurate trajectory for every environment [47].

The *Lines and Circles* approach is one of the simplest methods for generating a smooth trajectory [105]. This method uses straight and circular shapes to interpolate between known points and, hence, produce a more dense trajectory. Due to its relative simplicity, this set of techniques is computationally efficient; however, the trajectory is not continuous across multiple planning iterations, causing the overall trajectory to become jerky. On the other hand, *Polynomial Curves* of varying complexity can also be used to fit the center line points in order to generate a continuous and smooth path. The overall performance, however, largely varies with the order of the polynomial function being fitted [61]. High-order polynomials provide smoother trajectories but jeopardize computational time and efficiency. Low-order polynomials, meanwhile, output far more jerky trajectories which only loosely follow the center line but are still significantly more computationally efficient.

Grid Based Optimization

Grid based planners, also known as graph search planners, represent the drivable area around the ego vehicle as a set of discrete interconnected points. Once the drivable area is represented as a grid, a connected and directed graph can be created[47].

To solve the path planning problem, traditional graph based approaches can be used: Dijkstra was originally one of the post popular approaches[57], with A*[74], and D*[112] showing additional promise as the heuristic nature of these algorithms allows for more computationally efficiency. Alternatively, due to the 3-dimensional nature of the trajectory generation algorithm, State Lattice algorithms have also been used[135, 132, 85].

2.2 Motion Planning Benchmarks

We will now explore the related literature on motion planning benchmarking, dividing it into two sections: motion planning benchmarks in the general robotics domain, and motion planning benchmarks targeted specifically at autonomous driving.

2.2.1 Benchmarks in the Robotics Domain

The robotics community, in general, has presented several methods to benchmark motion planning algorithms across a variety of domains.

A large number of these benchmarks focus on general robot movement from one location to another; one of the most commonly used of these benchmarks has been integrated into the *Open Motion Planning Library* (OMPL) [113]. *OMPL* implements many of the most important path planning approaches with an emphasis on sampling-based planning methods. Along with these implementations, the OMPL library also offers a standalone benchmark [54] comparing the various approaches across a variety of robotics problems. Another popular benchmarking tool is *MoveIt!* [29], a benchmarking framework for decomposing path generation properties of motion planning algorithms, that has been integrated into the widely-used ROS (Robot Operating System) framework. Moll et. al. [88] have presented a set of interfaces which combine both OMPL and *MoveIt!* within a single API, alongside an array of benchmarking problems, and a visualization tool, thus making the process of benchmarking new robotics frameworks significantly easier.

The task specific domain of grasping-related motion planning problems is another area in which a large variety of benchmarks has been employed. Meijer [86] adapted the OMPL

library along with the *MoveIt!* framework to survey and compare multiple grasping algorithms. Due to the platform dependent nature of grasping problems, implementing such benchmarks was a largely difficult task. To address this issue, a platform-independent evaluation method for task and motion planning (TAMP) [69] was introduced. Much like the work presented in this thesis, TAMP introduced a common set of metrics, formats, and problem applications to make comparisons across the field more uniform. Finally, Iversen provided 3 different scenarios to benchmark motion planning algorithms in the task of bin packing[55].

2.2.2 Benchmarks in the Autonomous Vehicle Domain

AV research was largely kick-started by a pair of Defense Advanced Research Projects Agency (DARPA) challenges in 2004 and 2005 [119]. This early era initiative was also the first example of real-world test beds designed to benchmark motion planning as part of the overall autonomy stack. While revolutionary, the first set of challenges was rather limited both by the number of real driving scenarios and by the comparison metrics that were used. These limitations were addressed in the third 2007 DARPA Urban Challenge [27], which introduced more realistic and representative urban scenario suites. However, the set of metrics used to evaluate the different motion planning approaches of the competing teams was limited exclusively to progress metrics. In the same vein, many companies, such as Google [123], Kodiak [64], Apollo [127], and Lyft, have continued to use physical test-beds to test and benchmark internally developed motion planning approaches. Important to note, however, is the fact that these real-world benchmarks have not been available for academic or cross-organizational comparisons.

To combat some of the limitations with real-world benchmarking approaches, researchers have turned to simulation environments. CommonRoad, developed by Althoff et. al. [12] aims to solve the benchmarking problem through a collection of scenarios, vehicle models, and cost functions that are representative of the AV motion planning problem. Pek et. al. [99] offered a extension which adds significant better metrics of comparison to the metrics framed as real-time drivability checker. However there are a number of issues with this proposed benchmark. The set of metrics are rudimentary, focusing significantly more on path generation while neglecting to capture the decision making aspects the planning problem. Another issue with this benchmark is the limited nature of the simulator used, which overly simplifies interaction with other traffic agents and thus fails to transfer results to real world applications.

CARLA [35] is another popular open source simulation environment targeted towards AV research. This simulator is based on the Unreal engine. To combat some of the is-

sues highlighted above push the motion planning community forward, the 2019 (and 2020) CARLA Autonomous Driving (AD) Challenge [7] was unveiled. The CARLA challenges provide a more comprehensive set of evaluation metrics, and also aim to benchmark algorithms across a variety of realistic traffic, environmental, and weather conditions. Despite these improvements over other benchmarking attempts, however, the challenges are still limited by the low number of scenarios used and the random nature of the traffic agents that the ego vehicle encounters. This makes the challenge problematic in terms of reproducibility and usability.

Finally, similar to CARLA, Voyage introduced a Deepdrive [6] challenge. This challenge uses a high fidelity simulation environment along with evaluation metrics but is by far the most limited benchmarking tool discussed so far. Deepdrive, at the time of writing, contains only a single scenario that is available for benchmarking.

A table outlining a more detailed comparison of all motion planning benchmarks currently available can be found in Table 2.1.

	Simulation Environment	Scenario Suite	Comparison Methodology (Metrics)
CommonRoad [12, 99]	Low Fidelity	***Large Amount of Scenario* Data Generated Scenarios Hand Crafted Scenarios***	Simple Scenario Completion Comfort Progress
Voyage Deepdrive [6]	***High * Fidelity***	1 Scenario Single Unprotected Left	Simple Scenario Completion Comfort
CARLA Autonomous Driving (AD) Challenge [7]	***High * Fidelity***	10 Scenarios 10 NHTSA Critical Scenarios	Simple Scenario Completion Rule
WiseBench Framework	Medium Fidelity	***174 Scenarios * 78 NHTSA Critical Scenarios 96 ODD Specific Scenarios***	***Comprehensive * Scenario Completion Comfort Safety Rule Progress***

Table 2.1: Comparison between state-of-the-art motion planning benchmarks across 3 dimensions: simulation environment, representative scenario suite, and comparison methodology. Bold and starred (*) text indicates the best performing benchmark within a given dimension.

Chapter 3

Requirements

A Motion Planning benchmark may be defined as:

a method that allows for the evaluation of the performance of a single Motion Planning algorithm within its **operational design domain** (ODD), or for the **comparison** of the **performances** of two or more Motion Planning algorithms within a **shared, common ODD**.

To serve as a tool for performance evaluation of a single algorithm or for comparison between multiple algorithms, any motion planning benchmark must be able to realize a well-defined set of requirements. These requirements can be grouped into three broad categories: *Simulation Environment Requirements* for the simulators serving as virtual test beds, *Scenario Suite Requirements* representative of the given ODD or shared ODD, and a robust *Comparison Methodology* to highlight performance or differences in performance along a wide range of axes.

3.1 Simulation Environment

The race to full autonomy may be attributed to the first three major DARPA Grand challenges: the DARPA Grand Challenges of 2004 and 2005 [119], and the DARPA Urban Challenge of 2007 [27]. These challenges served as the first set of benchmarking opportunities for many fields of autonomous driving. Besides these challenges, however, no other publically available physical autonomous vehicle test beds have since been established.

In any case, real-life physical test beds may not always be feasible. For one, they may be unable to scale with respect to the number of scenarios required to benchmark

larger ODDs; secondly, they may not be entirely capable of testing all edge cases. For example, in order to test extreme edge-case scenarios that would essentially break the normal functioning of the automated driving system, a real-life test bed might actually result in costly damages to and destruction of the vehicle(s) being tested. Thus, in these situations, simulation environments that represent virtual test beds are more tenable to the use case.

The overarching requirement for any virtual test bed, is to **bridge the gap** between the **simulated** and **real world**.

Just as with any other virtual test bed, the primary requirement for simulated environments is to create simulations that represent the real-world as closely as possible. This is often difficult to measure and quantify. Thus, the primary simulation requirement can be further divided into a set of sub-requirements measured as dimensions of simulation *fidelity*.

Simulation fidelity may be defined as:

the degree to which a simulator is able to replicate or reproduce reality.

Simulation fidelity has been a focus in many other domains, aircraft simulation being one of the most common [78, 45], and is usually defined independently for each individual component of the simulator used. As an example, flight simulator fidelity comprises a number of dimensions, namely, visual input, control and kinesthetic feedback, motion cues, and environmental factors, with higher fidelity in certain dimensions – particularly motion cues – being more important than others [45]. Similarly, for AV motion planning as well, the highest level of fidelity does not need to be achieved for each and every dimension; rather, there should be a holistically high level of fidelity across the most important factors pertaining to the benchmarking task, so as to achieve the best real-life representation feasibly possible with reasonable cost.

For the purpose of autonomous vehicle motion planning, the simulation environment should be designed to achieve a specific set of requirements pertaining to the vehicle model, environment representation, traffic agent fidelity, input noise model, and weather model, see Figure 3.1. An example in the literature where this has been significantly explored is *WISE Drive* [33]. These requirements each have their own levels and dimensions of fidelity, which we now go on to explain below, drawing from examples of 4 different simulators: SUMO (Simulation of Urban MObility) [66], Applied Intuition [1], *WiseSim* [4], and NVIDIA DRIVE Constellation [5] (as seen in Figure 3.2).

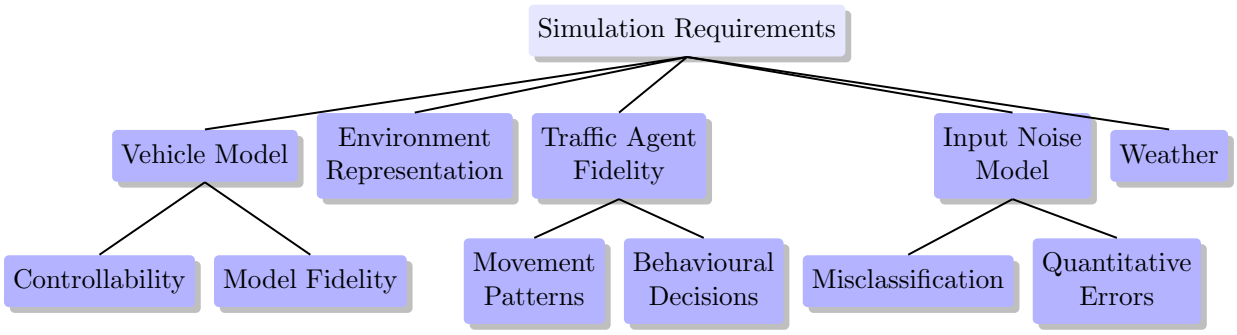
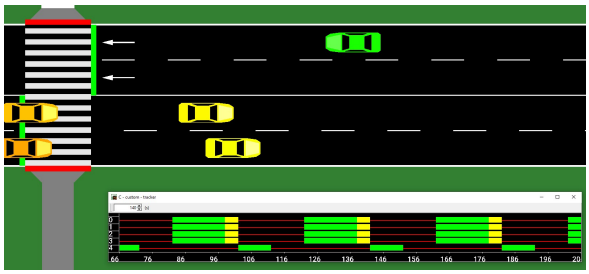
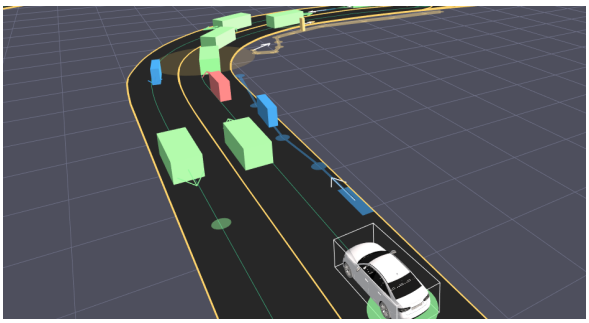


Figure 3.1: The simulation environment used should satisfy a set of requirements, each pertaining to a different aspect of autonomous driving.



(a) SUMO



(b) Applied Intuition

(c) *WiseSim*

(d) NVIDIA DRIVE Constellation

Figure 3.2: Four simulators with different levels of fidelity: (a) SUMO (Simulation of Urban MObility), (b) Applied Intuition, (c) *WiseSim*, and (d) NVIDIA DRIVE Constellation.

3.1.1 Vehicle Model

Requirement: The vehicle model for the ego should be controllable and the vehicle model should be able to execute the trajectory provided by the motion planner in a physically feasible manner.

The vehicle model used has multiple dimensions of fidelity:

1. **Type of model used:** Vehicle dynamics may be associated with numerous degrees of freedom, resulting in a wide range of vehicle models that may be applied to model the ego. These models include the simplest two-degree-of-freedom bicycle model that neglects the longitudinal direction, the three-degree-of-freedom model which adds the longitudinal component, and higher fidelity models that also account for the rotational degrees of freedom at the wheels, either in pairs (front wheels and rear wheels) or individually, as well as suspension models which consider the roll and pitch of the vehicle. Furthermore, kinematic models (with the appropriate degrees of freedom) are used in the absence of external forces, usually below speeds of 50 km/h, while dynamic models are used otherwise. The simulator should use the correct level of fidelity suitable for the task at hand: for example, lower fidelity models would work for simpler studies that require analysis only within the X-Y plane, while higher fidelity models would be required when analyzing traction and braking forces.

2. **Interaction with environment:** The vehicle model should account, also, for the interaction of the vehicle with its environment. In particular, it should account for the friction between the wheels and road surface, taking into consideration the type of surface and any precipitation that may be present, such as rain, snow, or black ice. Fidelity can range from simple models that have a constant coefficient of friction regardless of road surface conditions, to complex models that account for changes in friction depending on precipitation and road surface texture.

Examples: Applied Intuition and *WiseSim* are examples of simulators that incorporate high fidelity vehicle models with multiple degrees of freedom. SUMO, on the other hand, uses a very low fidelity vehicle model that is not suitable for the purpose of motion planning benchmarking.

3.1.2 Environment Representation

Requirement: The simulation environment should closely represent the real world as far as possible.

The simulator should be able to handle different kinds of representation formats for different components of the environment. For example, it should be able to handle bounding box representations for static and dynamic object (traffic agent) tracking, and map representations of the road network. Due to the significant differences in the objects represented, the method of measuring the environment representation requirement varies. For instance, bounding box representations should be measured on the accuracy of the bounding boxes used for modeling the static and dynamic objects they represent (for example a polygon will be higher fidelity than a box). Map representations can be measured as a standard error or offset from field measurements.

In addition, the environment, and the static and dynamic objects in particular, should exhibit a realistic appearance (with respect to the sensors used) so as to engage the perception layer to detect and track objects as it would in real world circumstances. The environment should also be able to realistically model different road surfaces and conditions such as road closures and construction, along with lane markings and regulatory elements such as traffic lights and signs.

Examples: The NVIDIA DRIVE Constellation simulator employs a very high fidelity environment representation that is realistic enough for perception algorithms to be employed directly on the environment itself. Applied Intuition and *WiseSim*, on the other hand, have a lower fidelity representation in terms of traffic agents, which are represented using bounding boxes. This representation, however, is still suitable for the purpose of motion planning.

3.1.3 Traffic Agent Movement

Requirement: The simulator should have the ability to generate naturalistic human-like trajectories and behavior for traffic agents.

To accurately represent real world conditions, all traffic agents in the simulated environment should have two dimensions of fidelity:

1. **Movement patterns:** Movement patterns determine how closely the simulated trajectory is able to reproduce the full range of naturalistic human driving, in terms of both the speed profile used as well as the actual path followed. Lower fidelity simulations would typically generate fixed robotic trajectories with no natural variation, while high fidelity simulations would typically be based on human-like models trained on extensive data sets sourced from naturalistic human driving in real-world conditions.

2. **Behavior patterns:** Behavior patterns determine how closely the simulated traffic agents are able to mimic the full range of maneuver decisions made by real traffic agents. These maneuver decision patterns can be defensive or aggressive in nature and include decisions such as stopping – or even running – a red light, overtaking other vehicles, and changing lanes. Behavior patterns can range from low fidelity simulations where all traffic agents perfectly obey traffic rules and drive in harmony with each other, to higher fidelity simulations where driving decisions are slightly more erratic, and occasionally, unpredictable.

Besides being able to achieve different levels of fidelity across movement and behavior patterns, the simulator should also provide the ability to set trigger points to achieve certain desired behaviors that can be used to evaluate specific scenarios.

Examples: Applied Intuition simulates, with high fidelity, both traffic agent movement patterns and behavior, while SUMO is able to achieve this only at a low fidelity level.

3.1.4 Input Noise

Requirement: The simulation environment should be able to adequately model input perception noise in order to mimic the real world perception errors observed when using the given set of perception algorithms in a given ODD.

While perception algorithms have significantly improved in the recent past, no set of algorithms is 100% accurate and errors in perception still exist in real-world driving situations. The simulation environment should be able to simulate these errors in two dimensions of fidelity:

1. **Misclassification errors:** These are of three types, namely, **False Negative** errors or failed detections, **False Positive** errors that involve false alarms, and **Incorrect Labeling** errors that detect an object but assign it the wrong class label. Fidelity here ranges from an assumption of zero perception noise, thus being able to perfectly detect and classify all objects in the scenario, to assuming a realistic noise model (based on real-world perception algorithms) that is able to sample different kinds of perception errors in a realistic fashion.

2. **Quantitative errors:** Sensor noise combined with inaccuracies in the perception algorithm used can result in quantitative errors such as errors in the current and future predictions of the position and velocity of traffic agents. Lower fidelity simulations would assume zero perception noise and would always perfectly know the location

and future path of surrounding traffic agents. High fidelity simulations, on the other hand, would be able to profile the distributions of noise and error for the trajectory positions and sample from these distributions in order to obtain close but imperfect predictions.

Examples: NVIDIA DRIVE Constellation and Applied Intuition both employ high fidelity input noise: perception models – with their associated error and noise – can be used directly on the environment in the NVIDIA Drive Constellation simulator, hence obviating the need for a separate noise model, while Applied Intuition applies the noise model directly to the traffic agent bounding boxes. *WiseSim*, meanwhile, which is still in development at the time of writing, can account for possible occlusions by only static objects, and does not otherwise use a suitable noise model to simulate perception errors.

3.1.5 Weather

Requirement: The simulation environment should be able to model all weather conditions within the specified ODD or shared ODD.

Weather conditions should be modeled in two dimensions of fidelity:

1. **Weather type:** These can range from sunny and clear skies to sub-optimal conditions such snow, rain, and fog. The simulation environment should be able to account for different types of weather and their impact on vehicle dynamics. Low fidelity simulators would be unable to simulate different weather types and would always assume optimal conditions, while higher fidelity simulations would take into account visibility effects due to fog and precipitation, as well as control effects such as changing coefficients of friction on road surfaces affected by precipitation (such as wetness from rain, snow buildup, and black ice).

2. **Intensity:** Weather intensity can range from mild or moderate to heavy for a given type of weather, and should be another dimension of fidelity for the simulator.

Examples: Applied Intuition and NVIDIA DRIVE Constellation are examples of simulators that use high fidelity weather models, although Applied Intuition does not simulate the weather visually. SUMO and *WiseSim*, however, do not account for a weather model at all. *WiseSim* assumes all scenarios to take place in clear, bright conditions.

3.2 Representative scenario suite

Reliable comparisons on motion planning algorithms can be achieved only in identical conditions. Thus:

The scenario suite used must consist of an extensive set of **representative**, **naturalistic** scenarios.

In order to achieve this requirement, the raw data used to generate each scenario should be as naturalistic as possible. Scenario environments can be crafted by constructing detailed paths and specifying parameters such as the type of road, regulatory elements, start and goal points, as well as environment constraints. Difficulty could be varied both by tuning scenario settings (such as the velocity profiles and positioning of agents involved, or the speed of the ego) as well as through the scenario environment itself. For example, highway driving in clear weather is inherently easier than driving through a 4-way intersection in foggy conditions. Finally, natural perception noise should also be present in order to support traditional perception algorithms and can be incorporated in one of two ways: either as a separate attribute of the scenario suite for each traffic agent, or as part of the simulation itself, generated from the ground truth of each scenario.

At the same time, the scenarios should cover hard to replicate cases, such as rare events like crashes and misbehaviour. While many methods of sampling rare events exist [109, 9], an alternative method is the combination of tailored, expert-crafted scenarios as well as through probabilistic simulations.

The scenario suite should cover a **range** of scenarios (that can be handled within the given ODD or shared ODD) and **difficulty levels** that are diverse enough to uncover algorithmic **differences**.

To satisfy this second requirement, the scenario suite should be scalable across multiple difficulty levels, testing not only routine events (such as lead vehicle following or overtaking) [36], but also edge cases (such as an oncoming vehicle drifting into the ego's lane). These edge cases rigorously challenge the motion planning algorithms being benchmarked, and uncover behavioral differences in the manner in which these algorithms respond to such events. The difficulty levels and ODD specifications should be able to be adjusted by the user according to their motion planning requirements.

It should be noted that a variety of difficulty levels should be used as each level would aid in individually teasing apart subtle behavioral differences which might not be apparent in a complex scenario that involves a combination of several challenges to the ADS.

At the same time, the set of scenarios used does *not* need to be the complete set of all possible scenarios covering the entire ODD: while regular and frequently occurring scenarios should be represented, difficult and rarer edge cases are more important for inclusion.

Scenario creation can be decomposed into 5 components which we go on to discuss (Table 3.1).

Road Structure & Geometry
Traffic Interactions
Reactions to Static Objects
Weather Conditions
Occlusion

Table 3.1: Scenario creation consists of 5 principal axis of decomposition components.

3.2.1 Road structure and geometry

These represent the geometry of the road, i.e., whether it is curved or straight, as well as the kind of connections that constitute it, i.e., if it contains multiple lanes, an intersection, biking or parking lanes, or a curb. In addition, the road structure describes, also, the kind of regulatory elements that are present throughout or at the end of the road, for example, speed limits, stop signs, sidewalks, crossings, and traffic lights. The road structures used should be able to represent the set of road environments that the ego is expected to handle within the bounds of its ODD. For example, if the ego's ODD includes 4-way intersections, roads with protected as well as unprotected 4-way intersections should be included in the representative set of scenarios. Further context is provided by *WISEDrive* [33] which offers a detailed description of road structure and geometry for autonomous vehicles.

3.2.2 Traffic interactions

These represent the interactions that the ego can have with other traffic agents such as vehicles, pedestrians, and other traffic agents. The traffic interactions should represent a wide range of realistic events that involve zero, single, and even multiple agents. This range should include, also, rare interactions such as misbehavior on the part of the traffic agents involved, and crashes that may or may not be rule abiding (in terms of traffic laws) in nature. Furthermore, these interactions should account for both the presence or absence of an occluding body (such as a large truck) (see Section 3.2.5).

3.2.3 Reactions to static objects

These involve ego vehicle maneuvers that occur as a reaction to static objects (for example, a pylon or a barricade) 5.12. Static reactions should cover, also, a variety of cases including no reaction at all as well as major reactions that require the ego to move into an adjacent lane to avoid the object. Again, these reactions may either be in the presence or absence of an occluding body (Section 3.2.5).

3.2.4 Weather conditions

These consist of atmospheric, lighting, and weather-related road surface conditions. Weather conditions typically introduce two types of difficulties: visibility challenges, and slip between the tires and road surface. Visibility challenges are posed by atmospheric conditions such as mist and fog, precipitation such as rain and snow, as well as lighting conditions such as daylight, dusk, or night. Slipping, meanwhile, is a result of lower coefficients of friction between the tires and road surface and is caused by precipitation such as rain, snow, or black ice. Scenarios should be created taking into account all possible weather conditions that can be handled within a given or shared ODD.

3.2.5 Occlusion

Object occlusion occurs when an object (or parts of it) are hidden by another object (for example, a small car being occluded by a large truck behind it, both of which are ahead of the ego). Occluding bodies serve as modifiers for each of the previous requirements. Occlusion can occur as a result of *road geometry* such as sharp turns that cut off visibility in the direction of the turn, *large traffic agents* such as trucks in front of or to the side of the ego, *static objects* such as a tree blocking the road ahead or large buildings to the side of the road, as well as *weather conditions* such as severe rain, snow, or fog.

Scenarios should be created taking into account any kind of occlusion that might occur as a result of road geometries, static objects and traffic agents, and weather conditions that fall within the ODD. For example, if the ego is able to handle traffic agents within its ODD then scenarios that involve occlusion by trucks should be included in the representative set of scenarios. Similarly, if the ego is able to handle precipitation, occlusion by precipitation should also be included.

3.3 Robust comparison method

A robust set of comparison metrics should be used to thoroughly and accurately compare the motion planning algorithms being benchmarked and tease apart their individual capabilities in multiple dimensions. In order to achieve this goal, the metrics used should be able to identify two distinct cases of the motion planning problem: **breaking points** in the motion planning algorithms that hit the boundaries of their behavioural capabilities, and **trade-offs** that each algorithm employs with respect to different aspects of the motion planning problem.

3.3.1 Behavioral capabilities

The comparison method used should be robust enough to distinguish between **behavioural capabilities** of different planners within a shared ODD, and should be capable of identifying **breaking points** of the algorithms being evaluated.

This thesis defines a breaking point as:

the point at which the logic of the given motion planning algorithm fails to appropriately handle a required scenario.

In order to perform exhaustive comparisons between different motion planning algorithms, a set of metrics should be devised that are capable of comparing two different motion planning algorithms while isolating their strengths and weaknesses within the shared ODD. These metrics should be diverse and strong enough to identify each planner's breaking points, especially in edge case scenarios.

Furthermore, the metrics should take into consideration aspects of the motion planning problem that *can* be controlled such as how different road surfaces are handled by the vehicle controller, as well as distinguish between behavioral decisions in situations that *cannot* be controlled such as reactions to misbehaving traffic agents.

3.3.2 Decisions and trade-offs

The comparison method used should be robust enough to identify motion planning decisions and **trade-offs** with respect to **different aspects** of the motion planning problem.

Different motion planning approaches may exercise different levels of trade-off depending on their goal and use-case. For example, some planners may heavily favor safety over progress and consequently may be very restrictive in nature, while others may prioritize occupant comfort over progress. These features should be teased apart and compared using different classes of metrics, which would each be capable of isolating the trade-off mechanisms employed in order to provide a more comprehensive comparison between different approaches.

Chapter 4

Simulation Environment

This chapter describes WiseSim, the simulation test bed used by the *WiseBench* framework. It covers the necessary components of the Autonomoose Platform, which serves as the necessary backbone for the correct execution of the entire autonomy stack, and includes a detailed description of the motion planning algorithm to be benchmarked. Furthermore, each component of the simulation environment is related back to the set of requirements previously outlined in Chapter 3.

4.1 WiseSim

In order to demonstrate our proposed motion planning benchmark framework, we use WiseSim (Figure 4.1), a simulator built on top of Unreal Engine 4 (UE4) [2]. Developed in C++ by Epic Games, UE4 is a popular video game engine that provides a rich set of well-documented features for creating complex and realistic 3D environments. The UE4 simulator interfaces with the Autonomoose stack to apply control actions (generated by the stack) on a high-fidelity vehicle dynamics model, modelled from real vehicle data [121, 49]. The resulting response from the simulated environment is then fed back into the stack ecosystem.

4.1.1 Vehicle Model

The Autonomoose uses a high-fidelity vehicle model that is fully controllable and able to safely execute any given trajectory.

The vehicle model used is based on that of a 2017 Lincoln MKZ vehicle, as seen in Figure 4.2. This is a high-fidelity rigid body model with 14 degrees of freedom, modeled from real vehicle data [121, 49]. The 14 degrees of freedom are composed of:

1. The chassis, considered as a single rigid body with 6 degrees of freedom
2. The front and rear wheels, with one degree of freedom each that allows each wheel to spin
3. The suspension system with 4 degrees of freedom that enable compression and decompression.

The model takes in 5 inputs consisting of the steering wheel angle, and torques for each of the four wheels. Model outputs consist of the state of the vehicle (position, velocity, acceleration, orientation, and angular velocity of the chassis) and each of its wheels (position, orientation and spin rate).

Due to its complexity and modeling on real-world data, this is a high fidelity model as per our requirements.

4.1.2 Environment Representation

The ego's environment uses a combination of High Definition road network maps, bounding box representations for traffic agents, and a realistic appearance to simulate the real world as closely as possible.

The ego's environment is composed of a High Definition (HD) Lanelet map [23] that describes the road network structure and associated regulatory elements, and a set of bounding boxes that describe the position and size of traffic agents present on the road. A rich, realistic, 3D representation of the environment is used, complete with a set of regulatory elements and textures for road surfaces.

Environment Map

The environment map is represented as an HD map of "lanelets". Lanelets are "atomic, interconnected drivable road segments" consisting of polylines (polygonal lines) of GPS waypoints [23]. These polylines represent the left and right lane boundaries and also include the center-line along with regulatory elements, as seen in Figure 4.3.

A lanelet consists of four main components:

1. **Boundaries:** polylines (an ordered sequence of GPS waypoints) that demarcate the lanelet's left and right lane boundaries, and the center line, as well as physical restrictions, such as curbs, that inhibit driving. The distance between points can be of varying levels of granularity ranging from a few centimeters to a few meters. The granularity of this distance depends on the smoothness of the polyline in question. The ordered sequence of waypoints serves as an indicator for the direction of travel and heading for the lanelet, and can be used to compute the curvature of the road or to interpolate the centerline for the desired path of the ego.

2. **Regulatory elements:** elements such as stop lines and static signs present *at the end* of the lanelet. These are represented as lines defined by a set of collinear points and require a specific action or decision to be made. For example, a traffic light is a regulatory element whose state (i.e., color) determines the course of action for the vehicle.

3. **Regulatory attributes:** these are like regulatory elements but persist *for the entirety* of the lanelet, such as speed limits, or attributes indicating whether the current lanelet crosses or merges with another.

4. **Associated intersection:** a reference to the intersection to which the lanelet is connected, if any. The intersection itself is represented as collection of lanelets that cross each other.

A lanelet is associated, also, with a type, representing the nature of the lane to which it belongs, for instance, a bicycle lane, driving lane, shoulder, parking lane, or bus stop.

An individual lanelet has four possible connections; these connections could be to other lanelets directly to its left and right, as well as to the lanelets immediately following or preceding it. In Figure 4.4, the indicated lanelet element is connected to two other lanelets on its right and left respectively.

It should be noted that there could be more than one lanelet preceding or following any given lanelet, as in the case of an intersection. The connections between lanelets, as stored in the map, allow for easy traversal and calculations for the ego vehicle.

A directed graph is used to represent all lanelets and their connections, and forms the base structure of the HD map. This map thus consists of three main components:

1. The **set of all lanelets** in the area.

2. The **set of all the connections** between different lanelets.

3. The **set of all intersections** with their constituent lanelets.

The HD map is, thus, a highly involved representation of the road network along with its regulatory elements, and serves as a high fidelity representation of the static environment.

Traffic agent Representation

A traffic agent is any non-stationary entity external to the ego vehicle, such as a moving car, truck, bicycle, pedestrian, or animal. With respect to *WiseSim*, the visual on-screen representation of these traffic agents uses only two models: one for pedestrians, and one for vehicles, Figure 4.1a. However, for the requirement of benchmarking motion planning algorithms, these are represented as a set of bounding boxes, as seen in Figure 4.1b.

Each traffic agent is represented as a bounding box which specifies its length, breadth, and height, and is associated with a class label, i.e., the category to which the object belongs (pedestrian or car). Since each traffic agent is created within the simulator itself, its size and location are known as ground truths and are preformed into the bounding box representation provided to the vehicle's tracker.

While this representation of traffic agents is indeed of low fidelity, it is still sufficient for the purpose of benchmarking motion planning algorithms and, hence, satisfies our requirements.

4.1.3 Traffic Agent Movement Fidelity

Traffic agents follow a set of predefined paths. These paths may either be derived from naturalistic driving data [22], or manually crafted using a set of GUI tools. While the simulator supports high fidelity movement patterns for each traffic agent, behaviour patterns inside the simulator are controlled by a set of trigger points, which can be set off based on location, time, or a pre-defined metric. Thus, as it does not support human-like interactions, the behaviour pattern fidelity is fairly low, although, for our purpose of benchmarking, it still provides enough support to model most levels of interactions with traffic agents.

4.1.4 Input Noise Model

The noise model employed uses a ray tracing LIDAR system to detect possible occlusion of objects for the ego as well as only detecting objects which are correctly within sensor range. Besides this, no other input noise models are present from the WiseSim simulator.

While the simulator has no additional noise model due to the Traffic Agent Tracking step (discussed below) errors were introduced both in the location of the objects as well as in the history for these objects.

However, due to misclassification and quantitative errors not being represented in the simulator, this model does not completely satisfy the input noise model requirements.

4.1.5 Weather Model

A simple weather model is used, restricted to daylight and clear conditions, which satisfies the constraints of the ODD.

Due to the ODD constraints of the Autonomoose platform, a high fidelity weather model is not implemented. This can be seen by the weather demonstrated in Figure 4.1a with clear blue skies.

4.2 The Autonomoose Vehicle Platform

The Autonomoose vehicle platform consists of four major subsystems: the **localizer**, which determines the ego vehicle state, **perception and tracking**, which detects and tracks traffic agents in the vicinity of the ego, the **motion planner** which computes a safe and smooth trajectory for the ego to follow, and the **vehicle control** system which executes the trajectory produced by the motion planner.

4.2.1 Localization

The localizer is responsible for calculating and updating the ego vehicle state, as well as for publishing transforms between different frames of reference. The vehicle state is a representation of the current position and velocity of the ego vehicle and is computed by means of sensor fusion, in which speed measurements of each of the four wheels are combined with LIDAR point cloud data and GPS measurements. The fused data is then passed as input to a Kalman Filter to obtain accurate estimates of the vehicle state. The localizer employs a constant velocity model and is restricted to mapped areas with no precipitation.

The vehicle state is published at a rate of 50 Hz, the average latency of which depends on the system load.

4.2.2 Traffic Agent Tracking and Prediction

The tracker is responsible for tracking the location, orientation, and position of each traffic agent over time and produces a history of its past and current positions and orientations, along with its next predicted state and collisions metrics (position and time of possible future collisions with the ego).

While, in real life, the tracker is typically fed with input from the perception subsystem, in the simulator, true bounding box representations and orientations for traffic agents are already known as ground truths. These bounding boxes and orientations are based on the true location of all visible objects in the scene and are provided directly by the simulation environment itself.

The bounding box representation and the orientation are fused to represent the tracked traffic agent. Objects are tracked continuously over time and are all simultaneously and continuously fed into a multi-objects Kalman Filter which produces a continuous estimation of their speed and position. Tracking is limited to a maximum of 30 objects at a time, and missing objects are tracked upto a maximum of 8 frames. Besides bounding boxes, additional inputs to the tracker include the vehicle state of the ego, and the lanelet map.

The average and worst case latency for the tracking module is 60 and 70 ms respectively, with an output frequency of 20 Hz.

4.2.3 Motion Planner

The motion planning subsystem consists of three main components: the mission planner, the behaviour planner, and the local planner.

Mission Planner

The mission planner is responsible for making high level decisions about the overall journey or mission of the vehicle. These decisions involve choosing the best route that optimizes a set of objectives such as time taken and energy spent, given inputs such as the starting and goal points, current state of the ego, and road conditions.

The Autonomoose's mission planner produces the route for the ego by continuously querying the HD map server with the current state and goal point to obtain the sequence of lanelets for the ego to follow. Due to the large size of the original lanelet map, a parameter is used to specify the radius of the desired queried area. For a lanelet to be considered, the distance between the ego vehicle and at least one of the waypoints within

the lanelet must fall within this specified radius. If this criteria is not satisfied, the lanelet will not be considered for any further computation.

Behaviour Planner

The goal of the behaviour planner is to take in the sequence of lanelets provided by the mission planner, and produce high level behavioural decisions or maneuvers, such as *stop*, *yield to traffic*, or *track speed*. These high level maneuvers allow the ego to navigate safely through various traffic and road conditions in order to reach its goal.

The behaviour planner realizes its objective by first taking in mapping and perception information and decomposing this information into a set of abstract predicates. It then passes these predicates to its rule engine, which uses an extensive set of rules based on traffic laws and regulations to process the predicates and generate a maneuver. Predicates are divided into 3 types:

1. **Ego predicates:** such as *"ego velocity"* and *"ego acceleration"*, which describe the state of the ego.
2. **Travel predicates:** such as *"is ego approaching a stop sign?"*, which describe how the environment map relates to the ego.
3. **Traffic agent predicates:** such as *"is there a lead vehicle present?"* or *"distance to traffic agent"*, which describe how the world of traffic agents relates to the ego.

The primary use for these predicates is to discretize the rich and complex representation of the environment into a set of premises that the rule engine can easily recognize.

The rule engine employed in the behaviour planner is an expert system that uses a lambda architecture framework to process the abstract predicates through a set of four main stages:

1. **Pre-processing:** during this stage, each predicate is decomposed further into a set of atomic propositions and combined with temporal propositions from the previous time steps.
2. **Maneuver generation:** in this stage, the set of propositions from the previous stage is evaluated by an initial set of *maneuver rules*, to produce the set of potential maneuvers, each with its associated constraints.

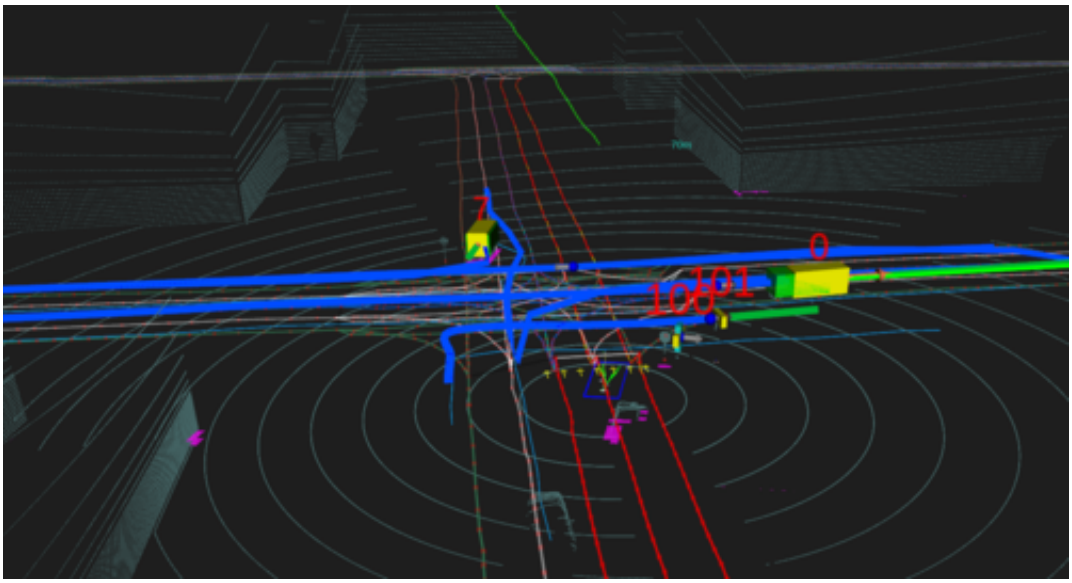
3. **Precedence evaluation:** here, the priority of each of the potential maneuvers is evaluated using a precedence table, with only the highest priority maneuver being retained. There are seven maneuvers that may be executed in increasing order or priority:
   (a) *track speed* which tracks the speed limit of the road
   (b) *lead vehicle follow* which maintains a safe distance to the lead vehicle while obeying the speed limit
   (c) *overtake* which overtakes and passes other vehicles in the path of the ego
   (d) *decelerate to stop* which gradually stops the ego when approaching a stop sign
   (e) *yield* which yields the right of way to traffic
   (f) *stop* which keeps the ego stationary, and
   (g) *emergency stop* which urgently stops the ego in unsafe situations.
4. **Constraints evaluation:** in this final stage, a second set of rules is used to analyze the environment and output a set of constraints that must be obeyed when executing the generated maneuver.

Thus, the final output of the rule engine consists of the maneuver to be executed along with a set of constraints that must be enforced by the vehicle in order to safely navigate its environment. The behaviour planner combines these maneuvers with additional information that it either extracts from mapping or calculates on its own such as stop locations, lane boundaries, and time to collision with surrounding traffic agents. The maneuver combined with the additional information and constraints is then sent to the local planner for execution.

Local Planner

The local planner uses the maneuvers and constraints supplied by the behaviour planner along with static object information (obtained from an occupancy grid) and vehicle state information (from the localizer) to refine and smooth the trajectory from the ego's current location to the next waypoint on the path. The local planner produces a safe and comfortable trajectory by avoiding immediate static and dynamic obstacles, and outputting a velocity profile along the path that satisfies all other system constraints. The full description of the planner can be found in the paper published by Zhang[133].

(a) Unreal visualization of the Simulator.



(b) RViz visualisations of the Simulator.

Figure 4.1: *WiseSim:* the simulator environment used for our motion planning benchmark framework.

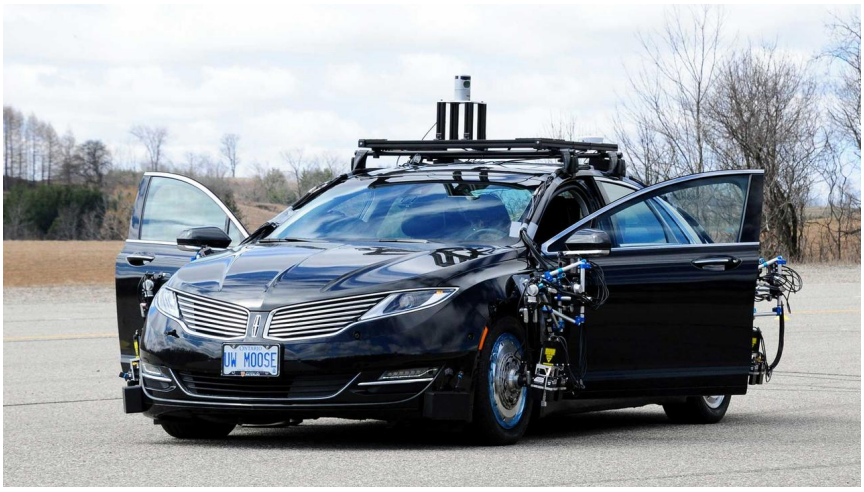


Figure 4.2: The Autonomoose, a 2017 Lincoln MKZ, serving as a platform for autonomous vehicle research at the University of Waterloo.

Figure 4.3: A lanelet representation overlaid on a satellite image of a road. Green dots represent the waypoints constituting the right lanelet boundary, while blue and white dots represent the left boundary and centerline respectively.

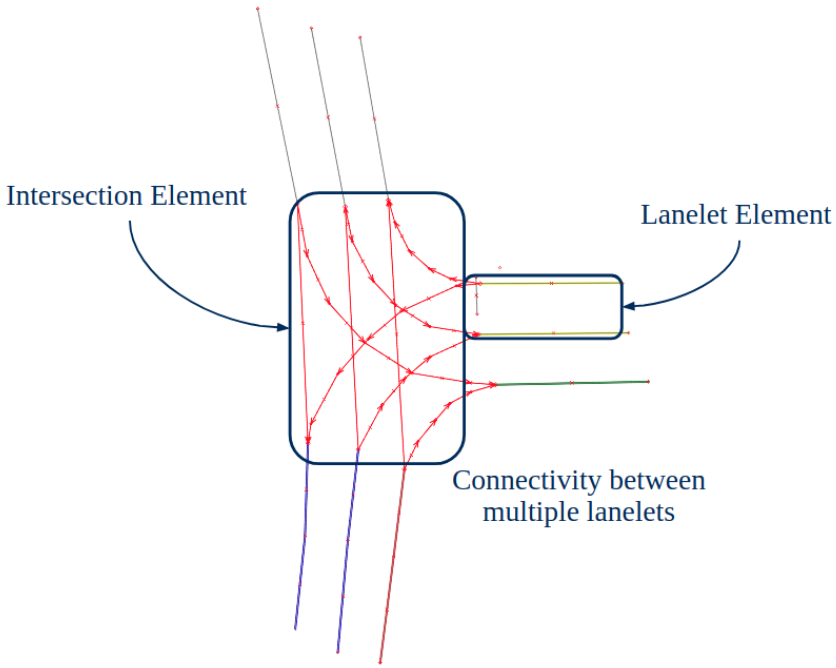


Figure 4.4: A simple representation of a set of connected lanelets and an intersection. Arrows represent the direction of the connection between successive lanelets, and the intersection depicts multiple lanelet connections in different directions.

Chapter 5

Scenario Suite Design

This chapter outlines a methodology that can be used to create a representative scenario suite for benchmarking motion planning algorithms. As highlighted in Chapter 3, the scenario suite used is critical to any benchmark as it serves to uncover behavioral differences across multiple planning algorithms.

This chapter begins by defining a feature model for scenario creation which explicitly defines the possible design choices for scenario creation. It then discuss how the proposed feature model can be used to aid in the construction of a representative scenario suite. Furthermore, it discusses the problem of numerical explosion in the number of scenarios required to cover an ODD, due to the numerous possible variations across each design feature. A solution is presented which groups similar scenarios based on *difficulty levels*, thereby bounding the number of scenarios required to create a *representative* scenario suite.

While most of the scenario creation process is developed exclusively with respect to the ODD of the motion planning algorithm being evaluated, or the shared ODD if multiple algorithms are being compared, this chapter discusses, also, the importance of a *critical* set of scenarios. In particular, the NTHSA pre-crash Scenarios are outlined as a good candidate towards this end.

Finally, this chapter concludes with a discussion on the implementation of the proposed methodology to create a representative scenario suite for the Autonomoose platform defined in Chapter 4. The scenario suite will then be used to benchmark different configurations of the Autonomoose motion planner, presented in the final results of Chapter 7.

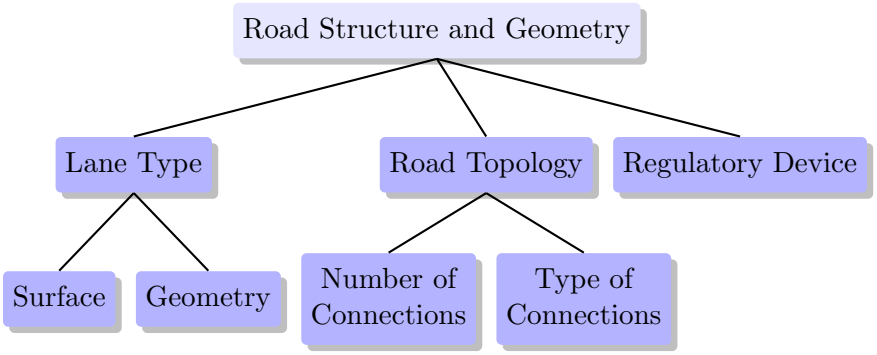


Figure 5.1: Feature diagram of all key design chooses related to the Road Structure and Geometry.

5.1 Scenario Decomposition

The National Highway Traffic Safety Administration (NHTSA), in its 2018 report[118], proposes an ODD Taxonomy to categorize the different considerations within an ODD. In this report, NHTSA demonstrates how the ODD framework should be used to design a scenario suite that has the capacity to completely test an entire ODD. *WISE Drive* [33] provides a much more comprehensive decomposition of an ODD. While very similar to the taxonomy presented below, the NHSTA and *WISE Drive* taxonomies differ in one crucial way in that they fail to classify the interaction difficulty of the AV with the static and dynamic objects of its environment. To elaborate, this taxonomy fails to classify and rank the behavior of other road users and their effects on the AV – a significant limitation towards its use in a representative scenario suite for motion planning benchmarks.

This thesis proposes a feature model for scenario creation that directly addresses the requirements highlighted in Chapter 3, as well as addresses some of the shortcomings of the existing NHTSA taxonomy.

5.1.1 Road Structure and Geometry

Road Structure and Geometry describes all the properties of the roads that an AV may encounter within its ODD. *WISEDrive* [33] presents by far the most comprehensive axes of decomposition. In this thesis, we elect to look at a simplified feature diagram, highlighted in Figure 5.1, which splits the design into three major points of variation: the *lane type*, the *road topology*, and finally the *regulatory device*.

Lane type describes the characteristics of each lane segment in the ODD. There are

two design principles to be considered: the **surface** impact of the lane on the ego vehicle's dynamics, and the interactions between the ego and traffic agents. The lane **geometry** describes the shape of the lane, for example, whether it is uphill or downhill, and whether it is a sharp turn or a straight road. In terms of curvature, Czarnecki [32] describes two aspects: horizontal curvature, and vertical curvature. This curvature has the ability to impact the ego by occluding objects, as well as changing the vehicle's dynamics significantly enough to impact the required time to make decisions.

Road **topology** describes the manner in which lane segments are connected to one another to form the road network. The **number of connections** influences the type of road network; for example, this number determines if the road is a simple two lane residential road, a sectioned 4 lane highway, or a complex interconnected intersection in the heart of a city. The **type of connection** influences how one road segment is connected to another: for instance, is there a solid white line or is it dashed and yellow, or is there a curb in between the two connected lanes?

Finally the **regulatory devices** refer to any elements that regulate the movement of agents (the ego as well as traffic agents) along the road. These include elements that are present at certain points of the road, such as stop lines, stop signs, and traffic lights, as well as regulations that must be obeyed throughout the entirety of the road, such as speed limits. These elements control both the individual movement of agents on the road, as well as the manner in which they interact with each other. For example, while the posted speed limit controls the maximum speed that agents can reach individually, elements such as traffic lights determine which vehicles have the right of way at intersections. [1]

5.1.2 Interaction with Traffic

Interaction with Traffic describes all of the design considerations which need to be made when designing the interactions between the ego vehicle and other traffic agents in the scene. The full feature model of all decomposed design axes is depicted in Figure 5.2.

The **number of interactions** refers to the number of different traffic agents that the ego vehicle is interacting with at any one time. There are only three types of interactions that need to be considered. *No interaction* refers to the case when traffic agents are present in the scene but the ego vehicle is not required to *directly* interact with them. As an example, when a pedestrian is walking on the sidewalk, or when an opposing car is passing the ego there is no direct interaction. A *pairwise interaction* is one in which the ego

[1] The full set of regulatory elements in Ontario Canada can be found at https://www.ontario.ca/laws/regulation/900615

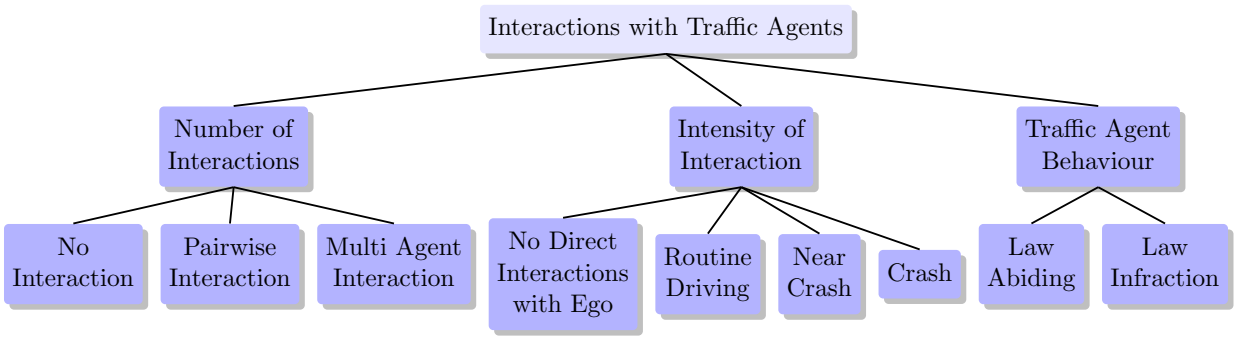


Figure 5.2: Feature Model of the key design consideration with regards to interaction between the ego vehicle and other traffic agents.

vehicle is required to interact with one other traffic agent, such as following a lead vehicle or waiting for a single pedestrian to cross an intersection. Finally, *multi-agent interaction* refers to an interaction with more than one traffic agent at a time, for example, the ego arriving at a busy intersection with multiple traffic agents interacting at once. *Pairwise* and *multi-agent* interactions can be caused either by direct interactions between the ego with one or more traffic agents, or by indirect interactions that occur when the ego reacts to an agent (or agents) that are, in turn, responding to the actions of another agent that has no direct interaction with the ego.

The **intensity of interactions** can be defined as the extent of the actions that the ego needs to take in order to complete its interaction with another vehicle. The extent of these actions can range from: *no direct interaction* at all to *near crash* interactions and even *crash* interactions. *Crash* situations and scenarios are those in which a crash simply cannot be avoided. One such example is when the traffic agent veers into the ego's lane leaving inadequate time for the ego to react. *Near-crash* situations and scenarios are those where a crash is imminent but can still be avoided. For example, near-collision situations and scenarios require an emergency maneuver for one or more of the involved road users in order to avoid the collision. All other situations in which two vehicles are interacting can be classified as *routine driving* interactions [36]. Finally, if the ego requires no interaction with another traffic agent in the scene this situation would be classified as *no direct interaction*. The Autonomoose Scenario Suite defines the intensity of interactions across 6 levels, further elaborated in Section 5.4.

Finally, **traffic agent behavior** describes the complete set of behaviors that a traffic agent may take throughout a scenario. If a traffic agent stops at a traffic light or follows the speed limit, that agent is said to be *law-abiding* throughout the scenario. However, if a traffic agent ever violates a regulatory element, or otherwise does not follow a set of

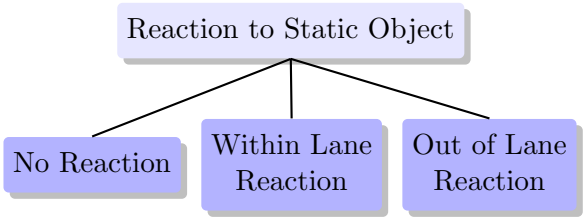


Figure 5.3: Feature diagram of the Reaction to Static Objects design choices.

well-defined social constructs, that vehicle's behavior may be classified as *law infraction*.

5.1.3 Reaction to Static Objects

As depicted by the feature diagram in Figure 5.3, **reaction to static objects** may be classified into 3 possible types: no reaction, within lane reactions, and out of lane reactions.

No reactions cover all instances of static objects in the environment which the ego vehicle does not need to react to, for example, a set of traffic cones outside its lane. **Within lane reactions**, meanwhile, are those that usually require an adaption of the ego vehicle's trajectory. This trajectory adaption is either in the form of hard braking to stop behind the static object, or path modification to avoid the object entirely. Finally, **out of lane reactions** are reactions to static objects that are present outside the ego's lane but still affect its decision making in some manner. A good example here is when a large static object occludes other traffic agents, such as in the case of a large sign preventing the ego from seeing oncoming traffic.

5.1.4 Weather Conditions

Weather conditions can be defined as either being constant throughout the entirety of a scenario, or varying throughout. The feature diagram depicted in Figure 5.4 highlights the main design features that should be considered when creating weather-related events or effects in a scenario.

In particular, the **atmospheric conditions** can be further decomposed into visibility challenges due to current atmospheric conditions, and the current precipitation. **Visibility** can be influenced by independent conditions such as fog, as well as by certain choices in precipitation – such as white out conditions. **Precipitation** itself is, in fact, broken down into two additional design choices, i.e., the **type** of precipitation (snow, rain, or hail) and its **intensity** (light or hard).

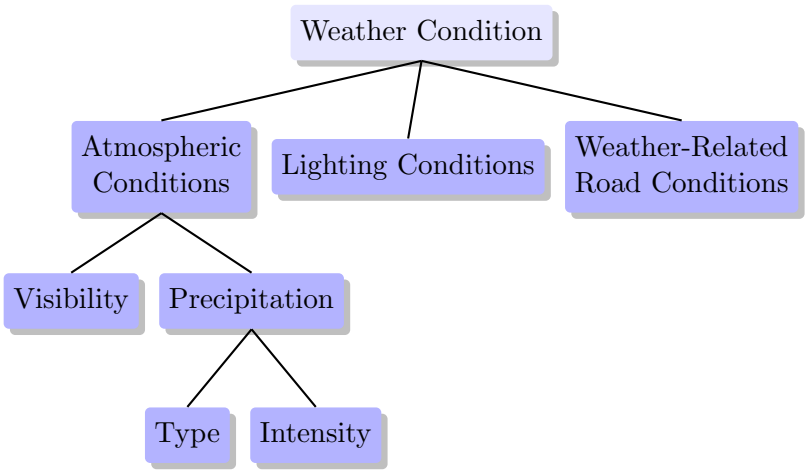


Figure 5.4: The feature model of the key design considerations with relation to the weather conditions in a scenario.

5.1.5 Occlusion

Unlike the previous design features, for the purpose of this thesis, **occlusion** does not have a clearly defined feature design model. This is primarily because it can be *implicitly introduced* by any of the features previously discussed. As an example, when considering road structure and geometry, the shape of a bend on a hill or mountain can occlude approaching objects if the bend is too sharp. Implicit occlusion can be introduced, also, by traffic agents, as in the case when a large vehicle in front of the ego, such as a bus, blocks its forward line of sight. Static objects can also cause implicit occlusion; in fact, most required reactions to static objects are a direct result of implicit occlusion. Lastly, bad weather can implicitly occlude the ego's line of sight, either as a result of precipitation, or even build up of particulate matter on the sensors around the vehicle.

Occlusion can also take on a more **explicit** role and can be introduced into a scenario to increase its relative demand. There are two design elements to consider when applying **explicit occlusion** to a scenario. The first of these would be to identify the set of traffic agents that will be the **target of the occlusion**; this will inform how adding occlusion to a given scenario will increase its demand level. The second design element is to determine the best *method* to occlude the target in a given scenario. This would involve using one or more of the other design features for the sole purpose of occluding an object.

Despite the lack of a clearly defined feature design model in this work, occlusion can still be classified in terms of criticality or difficulty. However, this classification falls beyond the scope of this thesis and can be addressed in future work.

In summary, for the purpose of this thesis, occlusion is a modifier that can be applied to any given scenario using the previously defined design axes. Adding occlusion to a scenario should be done to increase the demand on the ego vehicle and, hence, raise its difficulty level.

5.2 Scenario Suite Composition

Given the above decomposition of scenario design, the question still remains: how can this set of feature models be used to create a scenario suite?

If the goal of the benchmark is to create a scenario suite that will completely cover the entire ODD of the vehicle, a typical methodology would be to first consider all possible variations of each independent design axis individually. For example, for the *Road Structure and Geometry* $\rightarrow$ *Lane Type* $\rightarrow$ *Surface* axis, all possible variations – say, concrete, pavement, and dirt – across the given ODD would have to be identified. The Cartesian product of all these variations would then need to be computed to construct the complete set of scenarios (a simplified version of what this would look like is depicted in Figure 5.5). However, this would result in an infallibly large number of scenarios.

The requirements that we specify in Chapter 3, however, calls for a **representative** scenario suite, rather than a *complete* one. While it is very difficult to accurately measure if a scenario suite is **representative**. Section 5.3 offers an insight into how critical scenarios can be used to identify any possible limitations in scenario coverage.

The creation of this **representative** suite, similar scenarios can first be grouped based on their relative difficulty level, i.e., the scenario *demand* level. There will be set difficultly levels across four design axes: *interaction difficulty*, *reaction difficultly*, *weather difficulty*, and *occlusion difficulty*. *Road structure and geometry* is the only primary axis of decomposition whose demand level is harder to quantify and can thus remain independent and unchanged. Each independent axis (for example *Road Structure and Geometry* $\rightarrow$ *Lane Type* $\rightarrow$ *Surface*) would have to be examined over the entire ODD to identify all possible variations that will be used.

Once the difficulty levels have been determined, a single scenario can be created for each combination of difficulty level and *Road Structure and Geometry* variation. A better depiction of this combination can be seen in Figure 5.5. Each path through this graph is a single scenario in the representative scenario suite. For instance, let us consider one possible scenario highlighted in red *Straight Road, Reaction Level 0, Interaction Level 3, Weather Level 0, without any Occlusion Present* also depicted in Figure 5.6.

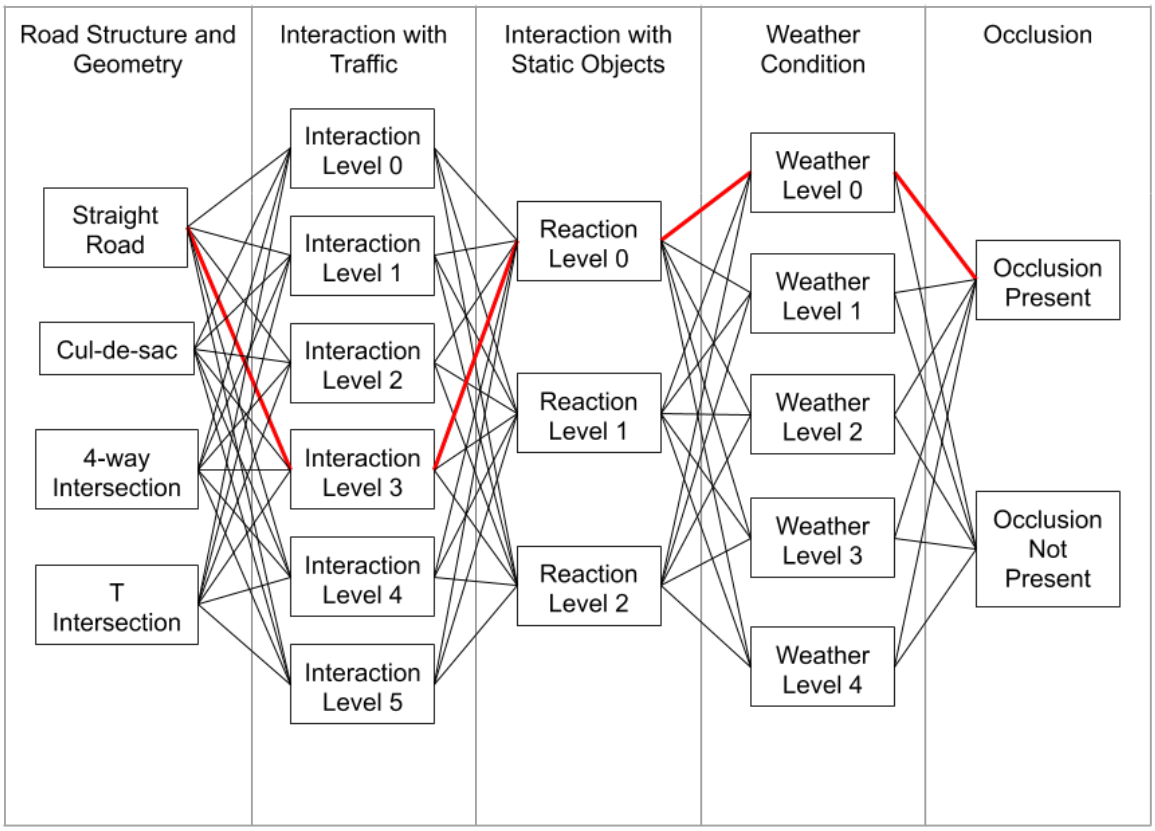


Figure 5.5: A visual representation of the composition of a scenario suite from a set of difficulty levels within each primary design axies.

5.3 Critical Scenarios

In order to ensure a basic level of competence on the part of the motion planners being evaluated, there must exist a fundamental set of scenarios that can be applied across the given ODD or shared ODD. Since the ODDs of most AVs include public road driving, basic competence should at minimum be ensured for this domain. NHTSA, in their 2008 report[92], highlighted a total of 37 pre-crash scenarios which account for "99.4 percent of all light-vehicle crashes" [92] caused by **human drivers**. Since even the human risk in these situations is high, a competent and robust scenario suite should include those critical scenarios that are applicable to the ODD being studied. Any additional NHTSA scenarios can be appended to the ODD specific scenarios, if they do not already exist.

Furthermore, these critical scenarios can also be used as a validation check for the scenario creation process. The exclusion of a valid critical scenario from the scenario suite, or a mapping of two or more critical scenarios to the same set of individual difficulty levels across the 4 axes (discussed in Section 5.2), can indicate a need for a finer level of granularity in the difficulty levels, say from 5 levels to 6 or more. Alternatively, such situations might also be indicative of a major flaw in the scenario design process and, hence, judging the closeness between two critical scenarios should be done based on human expertise.

It should be noted, however, that the NHTSA critical scenarios should not be the only validation check imposed on the scenario suite. Further validation checks should be employed based on critical events that occur in any available raw data. For example, any occurrence of disengagements in the raw data can serve as potential critical scenarios that are tailored to the ODD of the vehicle being evaluated.

Due to the vague descriptions provided in the NHTSA report, it is difficult to create a single depiction of any given NHTSA scenario. Thus, in order to both highlight the importance of these scenarios as well as correctly model the intended behavior, multiple parameterizations of any given scenario is recommended.

5.4 Autonomoose Scenario Suite

Given the scenario creation methodology outlined above, this section describes our specific implementation of the representative scenario suite relative to the Autonomoose and its ODD.

All scenarios were implemented in the GeoScenario Domain Specific Language (DSL)[103], and were hand-made as well as optimized to suit the **ODD of the Autonomoose**. Specifically, the ODD at the time of writing consists of only two-lane roads; lane-changing and

overtaking by ego is not possible. Stop signs and pedestrian crossings, as well as unsignalized intersections, are the only regulatory elements that can be navigated safely. In addition, only perfect weather conditions, i.e., clear and bright skies, can be handled. It should be noted that the ego cannot overtake either a static or a dynamic object in its lane if it would require ego to leave the lane.

Each scenario is comprised of 6 **interaction** difficulty levels, with *level 0* representing the easiest (no traffic interaction) and *level 5* representing the most difficult set of interactions with other traffic agents. As the ODD of the Autonomoose is limited, only 2 **reaction** difficulty levels will be used: *level 0* in which no static objects are present, and *level 1* which involves reactions to static objects outside the ego's lane; all other classes of reaction difficulty are omitted from the scenario suite.

Due to scenario design limitations, all interactions between the ego vehicle and other traffic agents are modeled with location or time based trigger points. To confirm that each trigger is able to achieve its desired results, the Autonomoose motion planner explained in Chapter 4 is made to run each scenario and the results are verified.

Two classes of scenarios are used: those that are specific to the Autonomoose ODD, and a set of critical *Pre-Crash Scenario Typology for Crash Avoidance Research* published by the National Highway Traffic Safety Administration (NHTSA) [92].

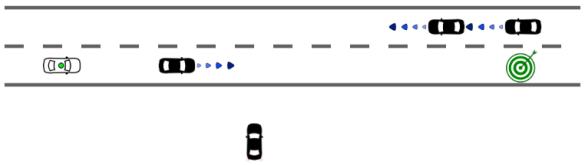
5.4.1 Autonomoose specific scenarios

These scenarios are representative of the ego's capabilities within its ODD, and cover 4 different types of road structure and geometry:

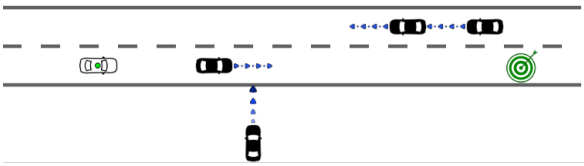
1. **Straight Road:** this scenario involves the ego driving down a two-lane straight road. Level 0 consists of no traffic interference at all, with the ego proceeding straight to its target. Higher levels involve increasing magnitudes of traffic interaction.

   For example, level 3 (Figure 5.6) involves non-interfering traffic agents in the opposite lane, a lead vehicle driving ahead, and an off-road traffic agent attempting to merge into egos lane. In response, the ego first tracks the speed of the lead vehicle, then decelerates and allows the off-road traffic agent to merge into the lane in front of it. It then proceeds towards the target, maintaining the minimum following distance required to the recently merged traffic agent.

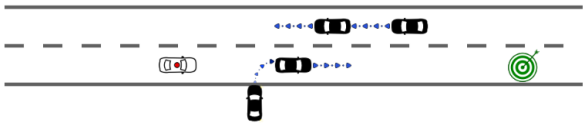
2. **Cul-de-sac:** this scenario involves the ego handling a cul-de-sac. In level 0 with no traffic interference, the ego decelerates to 30 km/h as it enters the cul-de-sac from



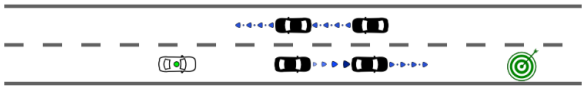
(a) Ego follows lead vehicle while maintaining minimum following distance and never exceeding speed limit.



(b) Off-road traffic agent begins driving towards the road indicating an intent to merge.



(c) Ego decelerates to allow off-road traffic agent to merge.



(d) Ego follows lead vehicle that has just merged, maintains minimum following distance, and reaches the target.

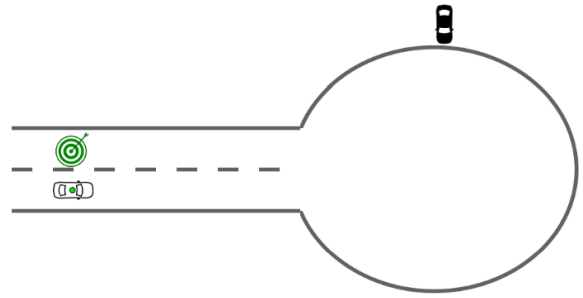
Figure 5.6: Straight Road, Reaction Level 0, Interaction Level 3

the right, proceeds counter-clockwise, and then exits the cul-de-sac and accelerates towards the target, resuming its normal speed of 50 km/h.

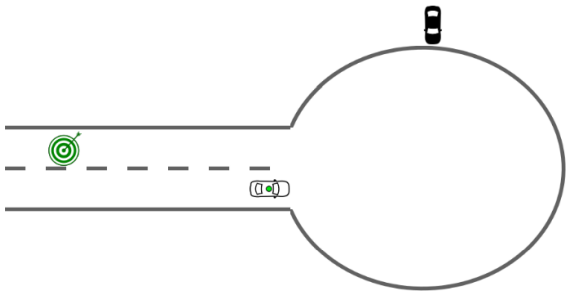
An example of a higher difficulty level: level 2, in which a traffic agent attempts to merge into the cul-de-sac on the opposite side (Figure 5.7). The ego detects the traffic agent, slows down to allow it to merge, and then follows it at the minimum following distance until it reaches the target.

3. **T-intersection:** this scenario involves the ego handling a T-intersection in which it must turn left towards the target. There are three regulatory element variations of this scenario: one with all-way stops, one with a stop only for the ego, and one with a stop for the other two directions of the intersection but not for the ego.

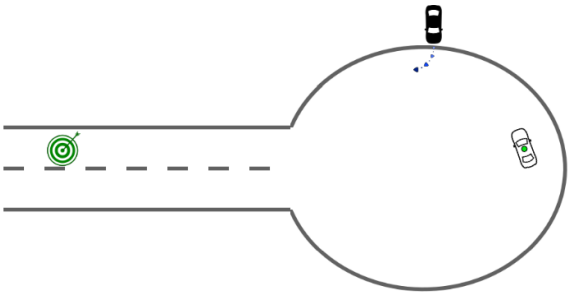
   We highlight level 4 for the case with all way stops (Figure 5.8). Here, a lead vehicle drives ahead and slows down. The ego follows it, slowing down as needed and then halting completely at the stop sign, while the lead vehicle clears the intersection. A non-interfering pedestrian crosses the intersection, while traffic agents stop on the



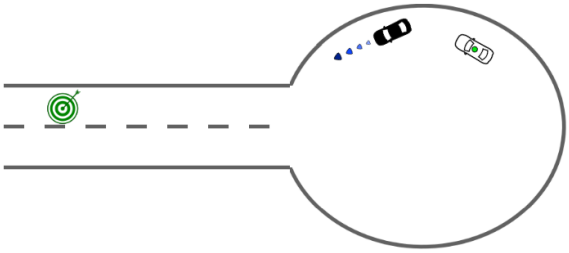
(a) Ego approaches a cul-de-sac there is a traffic agent waiting off road on the far side of the left side of ego.



(b) Ego reduces speed to 30 km/h as it enters the cul-de-sac from the right-hand portion of the road and proceeds counter-clockwise.



(c) Traffic agent begins to merge into the cul-de-sac; the ego notices the merging traffic agent and slows down.



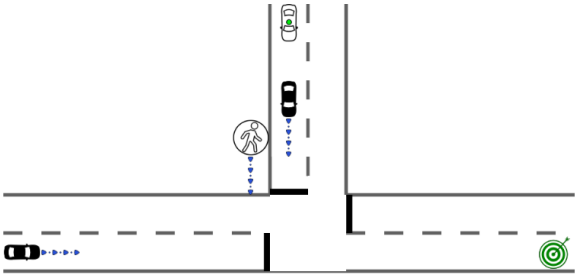
(d) Ego follows the traffic agent and reaches the target.

Figure 5.7: Cul-de-sac, Reaction Level 0, Interaction Level 2

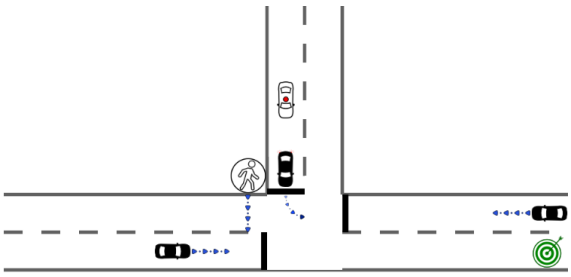
left and right. The ego recognizes that it has right of way since it handling stopped first, and proceeds to comfortably take a left turn towards the target.

4. **Four-way intersection:** this scenario involves the ego handling a four-way intersection in which it must proceed straight ahead towards the target. As before, this scenario also consists of three cases: one with all-way stops, one with a stop only for the ego, and one with no stop for the ego.

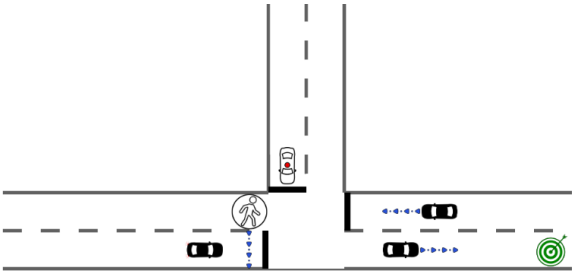
   We discuss the highest difficulty level, i.e., level 5, for the all-way stop case (Figure 5.9). In this level, the ego proceeds straight towards its goal and decelerates to stop at its stop sign. A traffic agent on the left approaches the intersection but, instead



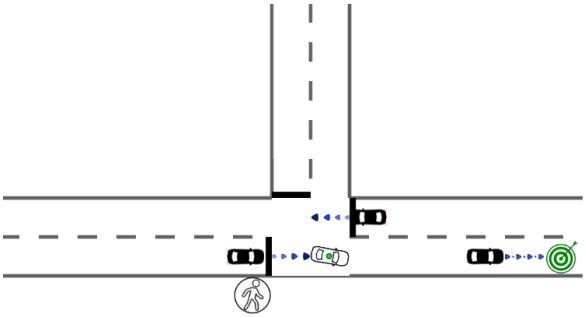
(a) Ego follows lead vehicle while maintaining minimum following distance and never exceeding speed limit.



(b) Ego slows down as lead vehicle slows down



(c) Ego halts at the stop sign



(d) Ego takes left turn as it arrived first at the stop sign and proceeds to the target.

Figure 5.8: T-intersection, Reaction Level 0, Interaction Level 4

of stopping, it runs the stop sign. Instead of crossing, the ego detects a potential crash and remains stopped until the traffic agent clears the intersection, after which it proceeds towards the target.

In summary, a total of 96 scenarios (9 road variations with 6 interaction levels and 2 reaction levels) are implemented on the Autonomoose ODD, in accordance with the methodology outlined above. [2]

5.4.2 NHTSA scenarios

These scenarios consist of a subset of the NHTSA *Pre-Crash Scenario Typology for Crash Avoidance Research*[92]. Some scenarios that were out of the ODD, such as passing or overtaking, are included primarily to show cases where the ego fails to handle the situation correctly and safely (see Figure 5.10 for a representation of a successful overtaking scenario, and Figure 5.11 for a scenario in which the ego takes a right turn and encounters a static object stopped in the middle of its lane).

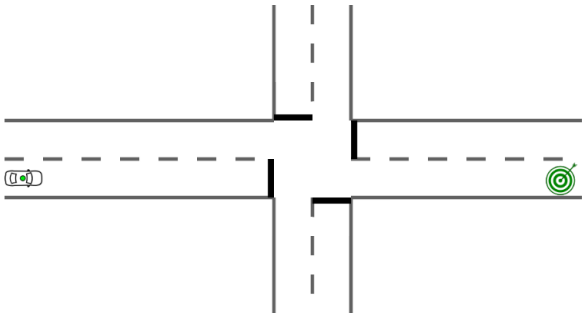
Due to limitations of the hand-crafted scenario creation process, some scenarios are modified. For example, because the ego behaviour cannot be controlled, scenarios in which the ego cuts into the opposite lane containing a traffic agent were modified by causing the traffic agent to veer into the ego instead (see Figure 5.12 for a depiction of this scenario).

We refer the reader to the original NHTSA typology document [118].

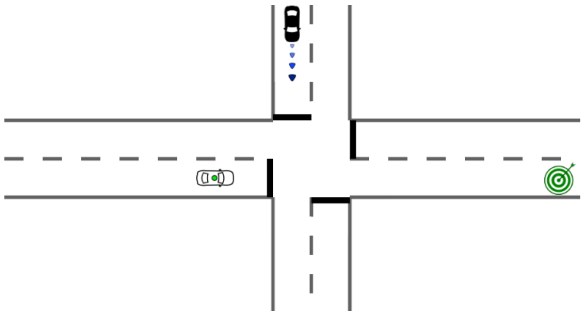
In summary, a total of 13 pre-crash scenarios were identified which meet the ODD constraints, of which 6 variations have been tested, resulting in a total of 78 NHTSA scenarios.

The total number of scenarios across both the Autonomoose-specific and NHTSA classes was 174. The complete list of scenarios along with detailed descriptions can be found on the following website: `http://wiselab.uwaterloo.ca/wisebench/scenario/straight-road`.

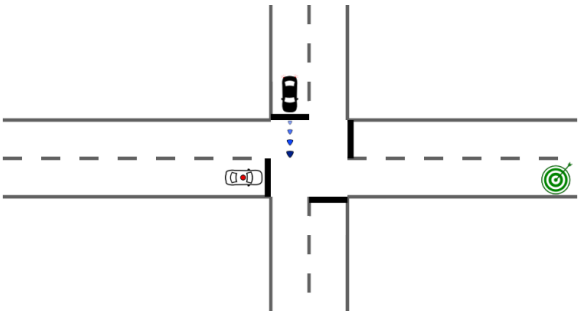
[2] The full list of all scenarios as well as full description can be found on the following website: `http://wiselab.uwaterloo.ca/wisebench/scenario/straight-road`.



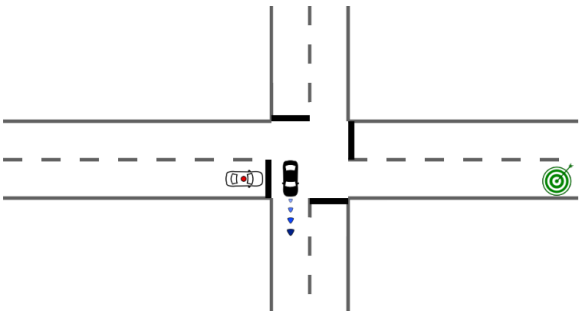
(a) Ego moves straight towards the intersection and target.



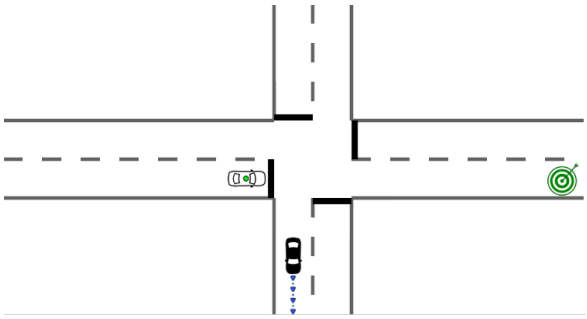
(b) Ego decelerates to stop at the stop sign.



(c) A traffic agent approaches the intersection, ego remains stopped.

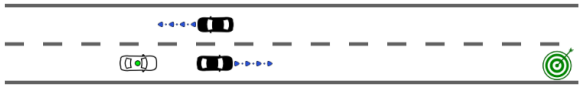


(d) The traffic agent runs the stop sign, ego detects a potential crash and remains stopped.

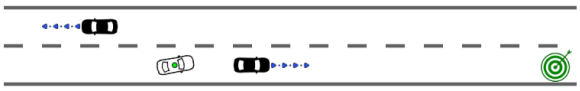


(e) Ego proceeds to the target once the traffic agent clears the intersection.

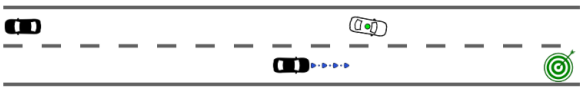
Figure 5.9: Four-way intersection, Reaction Level 0, Interaction Level 5



(a) Ego travels along a straight path following a leading traffic agent, another traffic agent travels in the opposite direction so ego should not overtake.



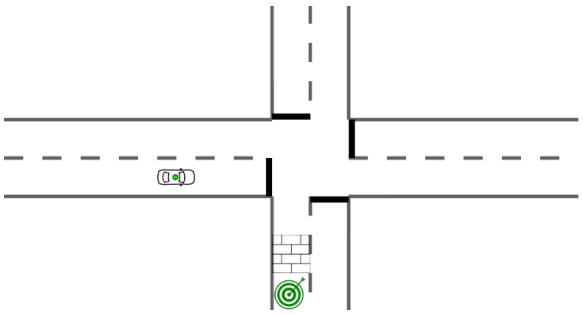
(b) Traffic agent passes completely, ego detects a clear lane for overtaking and starts to merge into the left lane.



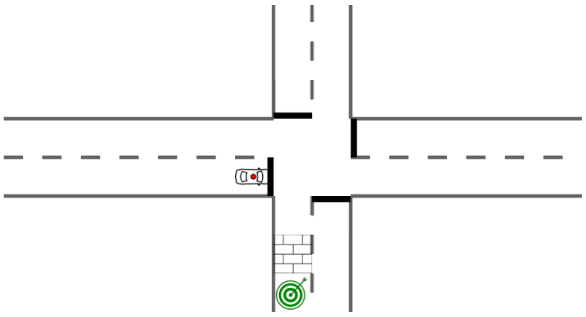
(c) Ego overtakes the lead traffic agent and attempts to merge back into the right lane.

(d) Ego completes the overtake maneuver and proceeds to goal.

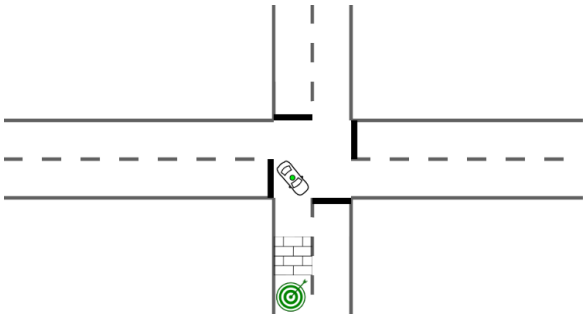
Figure 5.10: NHTSA Scenario, overtake with another traffic agent in the opposite direction.



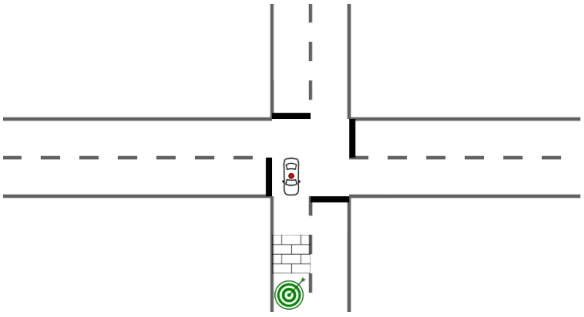
(a) Ego decelerates to stop at a stop sign, a static object is in the middle of the road on the right of the intersection.



(b) Ego comes to a full stop at the stop sign, does not yet detect the static object in its path.



(c) Ego turns right at the intersection.

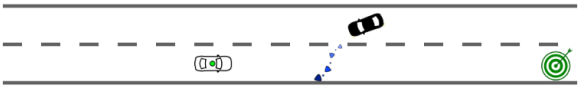


(d) Ego detects the static object and stops completely behind it.

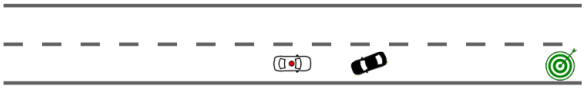
Figure 5.11: NHTSA Scenario, ego encounters a static object during a turn.

(a) Ego travels along straight path, a traffic agent travels ahead in the opposite direction, ego accelerates as it does not yet detect the traffic agent.

(b) Traffic agent begins to swerve towards the left, ego continues to proceed as the object is not in its lane.



(c) Traffic agent drifts into ego's lane, ego decelerates to avoid a crash.



(d) Traffic agent is partially or fully in ego's lane, ego should decelerate to a complete stop behind the object.

Figure 5.12: NHTSA Scenario, traffic agent in opposite lane drifting into ego's lane.

Chapter 6

Comparison Methodology

Motion Planning, at its core, is an optimization problem. Regardless of whether an algorithm is created through hand-tuned features or through a set of learned attributes, the very nature of any optimization problem calls for a set of trade-offs to be made. An example of one such trade-off would be: should the planner attempt to reach the goal quicker, or should it prioritize a a more conservative but safer approach? Alternatively, should the planner always follow the rules of the road, even if in rare occurrences in following them might lead to an accident?

As referenced in Chapter 3, a primary requirement when designing a comparison methodology for different planning algorithms is to both detect as well as quantify these trade-offs. To accomplish this requirement, this chapter presents a comparison methodology for motion planning algorithms that is built on a set of carefully chosen metrics. Each of these metrics seeks to isolate different aspects of the trade-offs that each planning algorithm might make. Since certain metrics are similar and measure the same type of trade-offs, we group them into individual classes, with each class holistically representing a particular aspect of the motion planning problem. Scores are calculated for each class of metrics in order to better understand their impact on the aspect of motion planning that they aim to describe. In order to do this, a penalty system is used to transform raw metrics values to scores on a 100 point scale.

This chapter describes each of these classes, and the individual metrics that constitute them, along with the scoring functions used, and the manner of implementation.

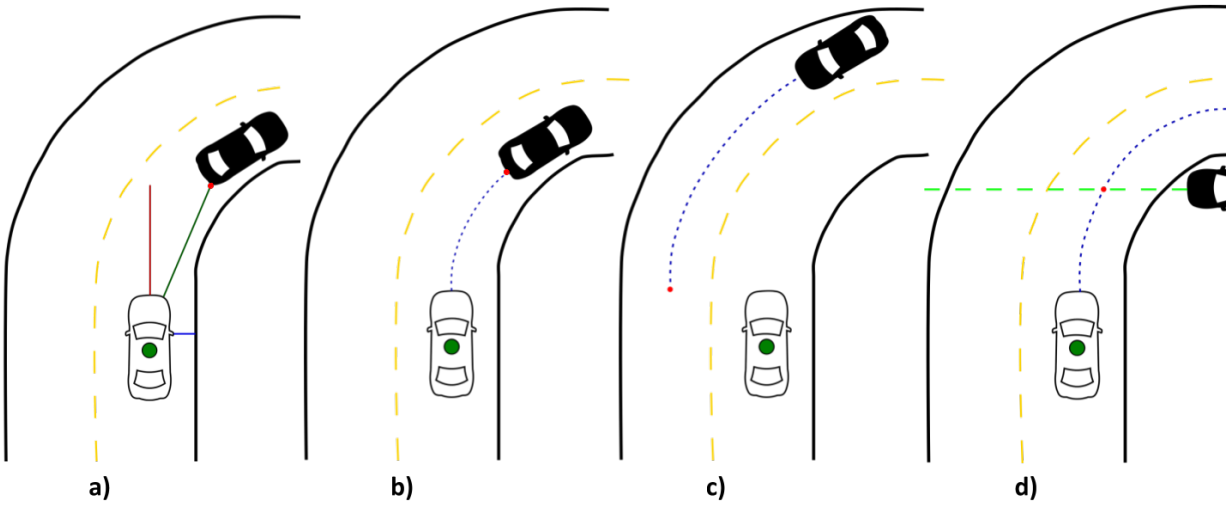


Figure 6.1: Safety metrics: (a) Euclidean metrics (distance, relative velocity, and relative acceleration), (b)Path metrics (following distance and time), (c) Path metrics (oncoming distance and time), (d) Collision metrics (time and distance to collision).

6.1 Metrics

The metrics that we use can be grouped into 5 classes based on the motion planning quality they represent. These include metrics on: the task completion success rate, safety, comfort, progress, and rule-abidance.

6.1.1 Task Completion Success Rate

The task completion success rate measures the number of scenarios that the ego is able to successfully complete. Successful completion of a scenario involves reaching the goal location within a set amount of time. Failure to complete the scenario results from either a complete time-out, where the ego stops making any progress at all, a crash in some part of the software stack that comprises the ego's ability to safely navigate through the environment, or a collision of any kind.

Completion metrics are always calculated for the entire scenario.

6.1.2 Safety Metrics

Due to the inability to calculate safety directly, surrogate or proximal safety metrics are used in conjunction to one another quantify it. Many of these metrics have been proposed in the literature [81], [110]. A good review has been provided by Czarnecki [31]. The National Highway Traffic Safety Administration (NHTSA) also published a report on their studies evaluating surrogate safety measures using both simulations (coming from multiple sources) as well as field data [46].

Our benchmark uses the following set of metrics to quantify safety in the ego vehicle:

Euclidean Metrics

These metrics include:

1. **Euclidean distance to the traffic agent:** this is calculated as the Euclidean distance in meters between the two closest point of the ego vehicle and the traffic agent's bounding box, referred though-out this thesis as the bumper-to-bumper distance. This metric is resolved onto three axes: the lateral and the longitudinal components, as well as the total distance.

2. **Relative velocity and acceleration with respect to traffic agent:** these represent the relative velocity and acceleration (in m/s and m/s^2, respectively) that the ego has with respect to the traffic agent and is also expressed in terms of their lateral and longitudinal components, as well as the total measurement.

Euclidean metrics can be seen in Figure 6.1a.

Path Metrics

These metrics measure path relative distances which are:

1. **Following Distance:** this represents the distance in meters between the front bumper of the ego and rear bumper of the lead traffic agent ahead of it. The distance must obey traffic regulations and be adequately large in order to avoid risking collisions.

2. **Following Time:** this represents the time in seconds to close the gap between the ego and the lead traffic agent at their current velocities, and is calculated by dividing the following distance by the relative velocity between the ego and the lead traffic agent.

3. **Oncoming Distance:** this represents the distance in meters between the front bumper of the ego and that of the oncoming traffic agent.

4. **Oncoming Time:** this represents the time in seconds required for the ego and oncoming vehicle to close the gap between them at their current velocities, and is calculated by dividing the oncoming distance by the relative velocity.

Figure 6.1b and 6.1c represent the following and oncoming distances and times, respectively.

Relative Collision Metrics

These metrics capture the urgency of potential collisions and are represented by the time and distance to collision (Figure 6.1d):

1. **Time to collision:** this represents the amount of time in seconds after which a potential collision will happen between the ego and traffic agent if both continue at the same relative speed along their predicted trajectories. Due to limitations in using a constant velocity model for prediction in the Autonomoose stack, the ego's path is split into multiple line segments, each of a certain maximum length. Intersection points between these line segments and the traffic agents predicted path are then found and the time taken for each of the ego vehicle and traffic agent to reach the closest intersection point is calculated. If the relative difference between these times falls within a specified threshold, it is taken to be the time to collision within that threshold accuracy.

2. **Distance to collision:** this represents the distance in meters from the ego to a potential collision point with a traffic agent, and is calculated based on the closest intersection point between their paths using the same method we just described.

6.1.3 Comfort Metrics

These metrics capture the level of comfort in the driving technique employed by the vehicle (see Figure 6.2) and are described in terms of:

1. **Velocity:** the rate of change of the ego's position with respect to time, expressed in m/s. The ego's velocity is derived from the ego's vehicle state estimate, and is expressed in terms of its lateral and longitudinal components, as well as the total velocity.

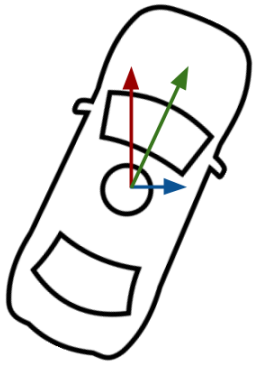


Figure 6.2: Comfort metrics: these represent the ego's total velocity, acceleration, and jerk, as well as their resolution into lateral and longitudinal components.

2. **Acceleration:** the rate of change of the ego's velocity with respect to time, expressed in m/s^2. This is also expressed in terms of the total acceleration, along with its lateral and longitudinal components. Acceleration was measured through a car-mounted IMU unit.

3. **Jerk:** the rate of change of the ego's acceleration with respect to time, expressed in m/s^3. Jerk is measured by calculating the derivative of the acceleration values. A first-order Butterworth filter is used to filter and improve the quality of the jerk signal.

6.1.4 Progress Metrics

We use a single progress metric: **time to goal** (Figure 6.3). This is calculated as the total time in seconds required for the ego vehicle to reach the goal location, and can be measured as the total time taken to complete the scenario in question. The progress metric is only calculated at the end of the scenario once the full completion time is known.

6.1.5 Rule Metrics

Rule-abidance metrics score the ego vehicle based on its ability to adhere to the rules of the road within its ODD. This class of metrics can be categorized into metrics representing **Lane Violations**, and those representing **Regulatory Violations**. Due to the relatively simple ODD of the Autonomous, as described in Section 5, the number of rule metrics is

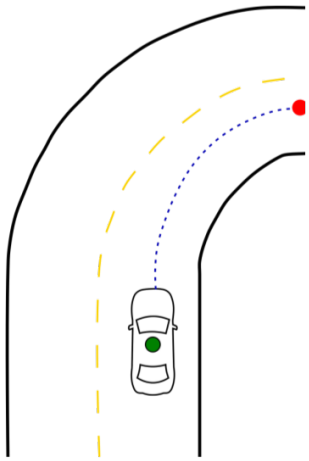


Figure 6.3: Progress metrics: this class consists of the time to goal metric. Here, the red dot represents the goal location.

kept at a relatively low number. Future implementations of this comparison methodology should implement a more complete set. Furthermore, all metrics are calculated in terms of number of occurrences.

Lane Violations

Lane violations metrics deal with situations in which the ego vehicle misbehaves within its lane causing disruptions to the flow of traffic, and are of three types:

1. **Lane Blocking:** when the ego remains stopped in a lane for more than 4 seconds, blocking the path of other traffic agents.
2. **Exiting the Lane:** when the ego vehicle exits its correct lane to enter an incorrect one, or moves outside the mapped regions.
3. **Blocking an Intersection:** when the ego stays completely stationary in the middle of an intersection for more than 2 seconds, blocking all oncoming traffic.

Regulatory Violations

Regulatory violations involve misbehavior on the part of the ego with respect to regulatory elements and attributes present along its path. These include:

Table 6.1: Comfort metrics – the threshold values used to classify the raw metrics into Stress Inducing Zones and Dangerous Value. The threshold values are accompanied with the time spent in each zone to assign a comfort score.

Metric		Stress Inducing Value (Time Spent)	Dangerous Value (Time Spent)
Ego Acceleration	Lateral	$2.5 m/s^2$ (1.2sec)	$5.0 m/s^2$ (0.7sec)
	Longitudinal	$3.0 m/s^2$ (1.2sec)	$5.0 m/s^2$ (0.7sec)
	Total	$2.75 m/s^2$ (1.2sec)	$5.0 m/s^2$ (0.7sec)
Ego Jerk	Lateral	$1.5 m/s^3$ (0.35sec)	$3.5 m/s^3$ (0.50sec)
	Longitudinal	$2.0 m/s^3$ (0.47sec)	$3.0 m/s^3$ (0.60sec)
	Total	$1.75 m/s^3$ (0.50sec)	$3.25 m/s^3$ (0.70sec)

1. **Speed Limit Violations:** represented as the total number of times the ego disobeyed the posted speed limit along its path.

2. **Regulatory Element Violations:** these occur when the ego ignores regulatory elements such as stop signs, and is measured by the number of occurrences in which it is not fully stopped at a stop sign for 3 seconds.

6.2 Metrics classification

While the metrics outlined above appropriately capture details on the manner in which a single scenario plays out, such as how good the ego was at keeping away from other traffic agents, they do little in terms of assessing the high-level trade-offs employed by the vehicle over multiple scenarios, neither do they appropriately represent how passengers themselves might actually feel while riding in the AV. Thus, there is a need to transform the raw metrics into scores that more holistically represent these trade-offs as well as passenger comfort.

While a limited amount of research on the relationship between motion planning metrics and passenger comfort does exist in the AV space [34, 89, 40], a much larger portion of the literature on metrics thresholds and passenger comfort is dedicated towards non-autonomous driving contexts [31, 117]. We discuss these works in further detail in Chapter 8. Furthermore, for the purpose of this thesis, passenger comfort was only discussed in the context of comfort and safety metrics, as no relevant literature has been found for those representing rule-abidance and progress.

Table 6.2: Safety metrics – The thresholds values used to classify the raw metrics into Stress Inducing Zones and Dangerous Value. The threshold values are accompanied with the time spent in each zone to assign a safety score.

Metric	Stress Inducing Value (Time Spent)	Dangerous Value (Time Spent)
Euclidean Distance	5.0m (1.2sec)	2.0m (0.7sec)
Path Relative Distances	5.0m (1.2sec)	2.0m (0.7sec)
Following Distance	10.0m (1.2sec)	2.5m (0.7sec)
Following Time	2.0sec (1.2sec)	1.0sec (0.7sec)
Time to Collision	1.5sec (1.2sec)	1.0sec (0.7sec)

We aim to classify each class of comfort and safety metrics into 3 basic *zones* from the point of view of a passenger in an AV: the zone of *comfort* in which passengers are generally satisfied with the driving, the zone of *discomfort* in which passengers are prone to be dissatisfied and uncomfortable by the type of braking, acceleration profile, or speed that the AV uses, and the *dangerous* zone in which the vehicle drives in an obviously dangerous manner, such as evading stop signs, violating speed limits, and other reckless behavior. For a more detailed description of this classification, see Table 6.1 and Table 6.2 where each metric is assigned a set of thresholds for each zone based on those found in the existing literature, simulation and limited in-vehicle experimentation.

Besides thresholds for the different zones, it is also important to consider the amount of time that a vehicle might spend in a given zone. For example, driving that is only occasionally jerky in nature is much more comfortable than driving with prolonged periods of high jerk. Thus, for every metric threshold and associated zone, we also discuss the amount of time spent in that zone which could further impact the contribution of the associated metric towards the overall score (see Table 6.1 and 6.2 for more detail).

6.3 Scoring functions

As discussed, holistic scores must be used to represent each class of metrics at a high level for better understanding of their effect on comfort, safety, progress, and rule-abidance, over multiple scenarios. Furthermore, each metric should ideally be weighted based on how much effect they have on actual passenger comfort and safety. However, since existing research does not yet provide much insight towards a possible weighting, we decided to keep all metrics equally weighted. A further discussion on the importance of comfort and

safety metrics is, however, provided in Chapter 8 in the context of a real life AV experiment on passenger comfort.

Scores for each of the comfort, safety, progress, and rule-abidance classes of metrics range between 0 and 100. We use a penalty system to arrive at the final scores for each class of metrics, where penalties are subtracted from the maximum score of 100. All scores are non-negative; thus, if the number of penalties subtracted causes the final score to dip below 0, the scores are will have a floor of 0. The penalties work slightly differently depending on the class of metrics being evaluated.

6.3.1 Comfort and safety scores

For comfort and safety metrics, penalties are subtracted for metrics that fall within the *discomfort* and *dangerous* zone. Metrics within the zone of *comfort* are considered ideal and not penalized. The penalty system used is as follows:

- **Within the *dangerous* zone:** 5 points are deducted for each occurrence of a metric within this zone.
- **Within the *discomfort* zone:** 3 points are deducted for each occurrence of a metric within this zone.

The number of occurrences can be calculated as the ratio of the total time spent in the zone to the time duration after which the ego is recognized as being in the zone. Thus, if the minimum time duration required for the *dangerous* zone is 3 seconds, and a given metric stays within this zone for a total of 9 seconds over the scenario, we estimate the number of occurrences within the zone to be 3, and a total of 15 points (3 times 5) will be deducted. See Tables 6.1 and 6.2 for more background on the point system for comfort and safety scores respectively.

The final score for each zone is then determined by the following formula:

$$Score_{class} = \begin{cases} 100 - occ_{dang} * 5 - occ_{discomf} * 3, & \text{if } 100 - occ_{dang} * 5 - occ_{discomf} * 3 > 0 \\ 0, & \text{otherwise} \end{cases} \tag{6.1}$$

6.3.2 Rule-abidance scores

All rule-abidance metrics are calculated in terms of the number of occurrences. A penalty of 5 points is deducted for each occurrence of rule violations. There are no zones associated with this metric.

$$Score_{Rule} = \begin{cases} 100 - occ_{violations} * 5, & \text{if } 100 - occ_{violations} * 5 > 0 \\ 0, & \text{otherwise} \end{cases} \tag{6.2}$$

6.3.3 Progress scores

Progress scores are calculated slightly differently: the minimum time to goal is determined from the baseline and, subsequently, a single point is then deducted for every 5 seconds spent above this calculated time. In cases where the planning algorithm performs even better than the baseline condition, no penalties are applied and the maximum score is capped at a ceiling of 100.

$$Score_{Progress} = \begin{cases} 100 - \frac{(t_{curr} - t_{ideal})}{5}, & \text{if } t_{curr} >= t_{ideal} and 100 - \frac{(t_{curr} - t_{ideal})}{5} > 0 \\ 0, & \text{if } t_{curr} >= t_{ideal} and 100 - \frac{(t_{curr} - t_{ideal})}{5} < 0 \\ 100, & \text{otherwise} \end{cases} \tag{6.3}$$

6.3.4 Limitations in scoring functions

In an ideal world, each class of metrics would have a different order of priority depending on the task domain of the motion planner. For example, safety might be more important than comfort in certain situations, while rule-abidance might be the most important in others.

Furthermore, within each class itself, each individual metric may have a different weighting in terms of its contribution towards the final score for that class. For example, within the class of comfort metrics, jerk might have the highest weighting towards overall passenger comfort as compared to velocity or acceleration.

In this work, equal weighting and orders of priority have been assigned both across as well as within each class of metrics. This is not, however, the only method of scoring this set of metrics but is more of an initial proposal to direct future work. Ideally, such inter-class and intra-class weightings should be determined by a panel of human experts and further optimized by in-depth investigations on each class of metrics; these investigations include passenger comfort studies, crash reports, emission reports, and traffic congestion reports, to name a few.

6.4 Implementation

The entire comparison methodology is programmed as a Metrics Node in C++11 using the ROS [3] middle layer framework, which adheres to all ROS standards, including standards on coding and messaging formats. The metrics are designed to work alongside the Autonomoose Autonomy Stack. The Metrics code base is designed to be deployed in both an offline fashion (as in the simulator experiment described in Chapter 7) and an online one (as in Chapter 8, where we linked the metrics to passenger comfort in a real life AV experiment).

The Comparison Metrics module consumes the following primary inputs from various subsystems of the Autonomoose (as described in Chapter 4):

1. **Planned Trajectory**: the final planned trajectory for the ego, as computed by the *Local Planner*.
2. **Behavioural Attributes**: the planned maneuver along with its associated set of constraints and attributes, as decided by the *Behaviour Planner*.
3. **Vehicle State**: information from the *Localization* module, which describes the ego's current pose and orientation.
4. **Scenario Event**: information from the *simulator*, which communicates details such as the name of the current scenario, and whether it was successful.
5. **Road Map**: the full environment map consisting of all lanelet connections, and regulatory elements such as stop signs and speed limits.

The metrics are then calculated based on these inputs and can immediately be exported as CSV or JSON objects to facilitate visualization by the front-end server.

The Metrics node is restricted to publish its output at a rate of 10 Hz, the same rate of publishing as the Local Planner, which provides it the planned trajectory. [1]

[1] Code for the *WiseBench* metrics node can be found in the following Github repositories: https://github.com/wavelab/autonomoose/tree/master/rospackages/autonomoose_core/path_metrics, https://github.com/wavelab/bp-benchmark-backend, https://github.com/wavelab/bp-benchmark-frontend

Chapter 7

Experiments and Results

This chapter evaluates the ability of our proposed WiseBench framework to distinguish between two or more distinct motion planning algorithms. This is achieved through two experiments. The first tests the ability of the framework to distinguish between subtle parameter differences across two configurations of the same planner. The second, meanwhile, tests the ability of the framework to distinguish behavioral differences between two or more planners.

Both experiments are run using the *WiseSim* simulation environment, the 174 scenarios in the representative scenario suite, and comparison methodology outlined in Chapters 4, 5, and 6, respectively. We now proceed to describe each experiment in detail.

7.1 Experiment 1: Distinguishing Parameterization

This experiment aims to answer the following research question:

Is the proposed *WiseBench* framework able to distinguish between subtle trade-offs made across two given motion planners?

To answer this question, the Autonomoose motion planning algorithm explained in Chapter 4 is modified in two different ways, each modification serving as a separate planner to be benchmarked.

Table 7.1: Motion planning parameters that are modified for Experiment 1: only acceleration values are modified with respect to *baseline* conditions for the *increased aggression* manipulation, while look-ahead and lead-vehicle following distances and times, approaching vehicle time, and parked vehicle distance are modified for the *decreased aggression* manipulation.

Parameter	**Baseline**	**Decreased Aggression**	**Increased Aggression**
Longitudinal acceleration	2.5 m/s^2	2.5 m/s^2	**4 m/s^2**
Lateral acceleration	2 m/s^2	2 m/s^2	**4 m/s^2**
Lead vehicle following distance	10 m	**25 m**	10 m
Lead vehicle following time	4 s	**8 s**	4 s
Look-ahead minimum distance	15 m	**30 m**	15 m
Look-ahead minimum time	2 s	**5 s**	2 s
Approaching vehicle time	4 s	**10 s**	4 s
Parked vehicle distance	0.75 m	**5 m**	0.75 m

7.1.1 Procedure

The primary objective of this experiment is to evaluate whether the proposed framework for scenario creation, combined with the set of proposed comparison metrics, is able to identify and distinguish between trade-offs that different motion planning algorithms make. One such trade-off – which is also the one we investigated – is comfort versus progress.

To accurately answer our research question, the motion planning algorithm is manipulated through two distinct parameterizations: by changing the **level of aggression** employed from the point of view of the local planner, and by changing the **level of defensiveness** with respect to other traffic agents employed by the local and behaviour planners. The actual values of the modifications applied for each of these manipulations can be found in Table 7.1.

Manipulation: increased aggression

The first manipulation involves increasing the values of thresholds for the maximum amount of longitudinal acceleration and deceleration reached, as well as the lateral acceleration. These modifications are made from the local planner's point of view. The intended effect of raising these thresholds is to make the ego vehicle accelerate and decelerate faster, as well as take more aggressive, sharper turns.

Table 7.2: Significance values for each set of metrics (with respect to baseline conditions) for each of the 2 manipulations for Experiment 1.

Metric	**Increased Aggression** p (t)	**Decreased Aggression** p (t)
Progress	*0.348 (-0.940)*	*0.000 (4.411)* ***
Comfort	*0.001 (3.192)* ***	*0.089 (-1.703)* *
Safety	*0.350 (-0.935)*	*0.043 (-2.028)* **
Rule-abidance	*0.935 (-0.080)*	*0.696 (0.390)*
Note:		*p<0.1; **p<0.05; ***p<0.01

Expectation: with respect to the original algorithm serving as a baseline, the increased aggression should cause the *comfort* metrics to decrease in score, and the *progress* metrics to increase. *Safety* and *rule abidance* scores should stay about the same. The overall number of completed scenarios should also remain constant.

Manipulation: decreased aggression

The second manipulation involves decreasing the value of several parameters with the aim to make the car more defensive in its behaviour with respect to other road agents. The changes made involve significantly increasing the *approaching vehicle time* (the maximum time required for a traffic agent to reach an intersection and be considered in the decision making process of the ego at that intersection), as well as distances with respect to other road agents (such as the *following* and *look-ahead* distances). Consequently, the ego maintains larger distances to other objects, and exhibits an overall cautious behaviour when navigating through the environment. Like before, these changes are made from the point of view of both the local and behaviour planners.

Expectation: with respect to the original algorithm serving as a baseline, all *safety* metrics scores should improve as a result of the increased defensive behaviour. There should be no significant changes in *comfort*, *progress*, or *rule abidance* metrics.

7.1.2 Results

The number of successful scenarios and distributions of the metrics scores are computed for each manipulation, and single-tailed t-tests are used to compare differences in these distributions and the baselines (due to the large sample size (174 scenarios), these t-tests are valid despite the non-normal nature of these distributions). A significance level of

.05 is used, although we also consider less significant results ($p < 0.1$) as indicative of potential differences. The significance values for each set of metrics and for each of the two manipulations can be found in Table 7.2.

Completed scenarios

Both manipulations successfully passed **128 out of the 174 scenarios**. This is in line with our expectations, as the behavioral decisions taken by the ego are not changed, and, hence, should not have affected its ability to safely complete each scenario.

Distribution of metrics scores

Progress metrics As can be seen in Figure 7.1, the distributions for the *baseline* and *increased aggression* manipulation are comparable with no significant differences ($p > .05$). However, significant differences do exist between the *baseline* and *decreased aggression* distributions ($p < .05$).

Figure 7.2c shows that the progress scores for *decreased aggression* are in general lower for most scenarios as compared to those of either the *baseline* or *increased aggression* cases. In fact, even for the *increased aggression* case, while the results are not significant, we are still able to see an overall higher number of scenarios obtaining higher progress scores as compared to the *baseline* (Figure 7.2b).

Comfort metrics Figure 7.2 depicts the distributions for the comfort metrics. The distribution for *increased aggression* is significantly different from that of the *baseline* condition ($p < .05$), while the *decreased aggression* case is found to be weakly significant ($p < .1$).

Visually, the comfort scores in general can be seen to be on the lower side (compared to *baseline*) for the *increased aggression* case (Figure 7.2b), and tends towards the higher side for the *decreased aggression* manipulation (Figure 7.2c).

Safety metrics The safety metrics do not show any significant differences between the distributions for *increased aggression* and the *baseline* condition, although there is a weakly significant difference ($p < .1$) for the *decreased aggression* manipulation. Distributions for these manipulations can be seen in Figure 7.3.

Rule-abidance metrics No significant differences are observed between either manipulation and *baseline* for the rule-abidance metrics. However, a ceiling effect can be seen in all 3 experimental conditions as seen in Figure 7.4, which may be attributed to the robust nature of the rule engine in appropriately handling all scenarios that are within the ego's ODD.

7.2 Experiment 2: Distinguishing Behavioral Capabilities

This experiment aimed to answer the following research question:

> Is the proposed *WiseBench* framework able to detect differences in the behavioral capabilities between two given motion planners?

As in the previous experiment, to test the framework's ability to detect differences in behavioural capabilities, the Autonomoose motion planning algorithm is modified in two ways.

7.2.1 Procedure

The primary objective of this experiment is to evaluate whether the scenario suite and comparison metrics are able to identify and distinguish between the behavioural capabilities of different motion planning approaches. To realize this objective, the motion planning algorithm used is modified to misbehave by manipulating a subset of rules employed by its rule engine in two ways, each governing the behaviour of the ego with respect to stop signs.

Manipulation: reduced time spent at a stop sign

The first manipulation involves reducing the amount of time spent stopped at a stop sign from the legal requirement of 3 seconds (according to Ontario traffic regulations) to 0.5 seconds. While this does not render the ego completely useless at stop signs, it does significantly reduce its capabilities in scenarios that involve them.

Expectations: the *safety* and *rule abidance* metrics scores should decrease, and the total number of successful completions of a scenario are also expected to go down as the planner loses capability to completely handle the ODD. However, the intensity of these

Table 7.3: Significance values for each set of metrics (with respect to baseline conditions) for each of the 2 manipulations for Experiment 2.

Metric	**Reduced Time** *p (t)*	**Stop Sign Disabled** *p (t)*
Progress	*0.002 (-3.026)* ***	*0.000 (-4.336)* ***
Comfort	*0.156 (1.420)*	*0.837 (-0.205)*
Safety	*0.860 (0.177)*	*0.379 (0.881)*
Rule-abidance	*0.024 (2.272)* ***	*0.000 (4.820)* ***
Note:		*p<0.1; **p<0.05; ***p<0.01

score changes should be less than the previous manipulation as the ego is still somewhat capable of handling stop signs.

Manipulation: stop signs disabled

The second manipulation involves completely purging all rules governing the ego's behaviour at stop signs. This renders the ego completely incapable of safely handling intersections.

Expectations: the *safety* and *rule abidance* metrics scores should decrease to a large extent, and the total number of successful completions of a scenario are also expected to go down as the planner completely loses its capability to handle stop signs in the ODD.

7.2.2 Results

As before, we use the number of successfully completed scenarios as well as metrics distributions to compare each manipulation with the baseline conditions, using a significance level of .05. For convenience, we refer to the two manipulation conditions as *reduced time* and *stop sign disabled*, respectively. The significance values for each set of metrics and for each of the two manipulations (compared to their respective baseline conditions) can be found in Table 7.3.

Completed scenarios

The total number of completed scenarios for the **baseline case is 128**. The *reduced time* manipulation results in a total of **136 completed scenarios**, while the *stop sign disabled* manipulation results in **142 successful completions**. This set of results is observed due

to limitations in the scenario creation process, which we highlight further in the Discussion section.

Distribution of metrics scores

Progress metrics Figure 7.5 shows the distributions for each experimental condition for the progress metrics. Highly significant differences compared to *baseline* ($p < .01$) are found for each manipulation. However, it should be noted that the actual effect itself is actually in the opposite direction than that expected, as the progress scores increase, rather than decrease, for each manipulation.

Comfort metrics No significant differences with respect to *baseline* are observed for the comfort scores. On observing Figure 7.6, the distributions of the scores are confirmed to be largely the same for the *reduced time* manipulation, and slightly higher for the *stop sign disabled* manipulation.

Safety metrics The safety metrics also did not exhibit any significant differences for either manipulation with respect to *baseline* conditions, as can be seen in Figure 7.7.

Rule-abidance metrics Highly significant differences ($p < .05$) for rule-abidance scores with respect to the *baseline* are found for both the *reduced time* and *stop-sign disabled* manipulations. Figure 7.8 shows that the rule-abidance scores decrease for a larger number of scenarios in the *reduced time* condition. Furthermore, in the *stop sign disabled* condition these scores deteriorate even further due to the ego's complete inability to handle stop signs.

7.3 Discussion

As discussed previously in Chapter 5, the scenarios used display a wide range in difficulty levels, which facilitates a thorough and exhaustive comparison between the motion planners that have been benchmarked. Our experiments provided 3 major takeaways:

1. **Our benchmark is able to clearly distinguish between subtle trade-offs made across two given motion planners.** The results of Experiment 1 are, by and large, in line with our expectations: progress scores increase or decrease proportionally with respect to an increase or decrease in aggression, while comfort scores display an inverse relationship. The number of successfully completed scenarios also

remain unhindered, as expected. Furthermore, in Experiment 2, the rule-abidance metrics deteriorate as the planner's ability to handle stop signs is increasingly compromised.

2. **While unexpected limitations in scenario execution prevent certain behavioral differences from being uncovered, our benchmark is still able to reflect the actual situations that occur.** Unlike Experiment 1, Experiment 2 does not confirm all of our expected results – only the rule-abidance scores deteriorate as expected but the successful completion of scenarios is not affected. We attribute this, however, to limitations in the triggering mechanism in the scenario-creation tool and not in our metrics. These limitations prevent the intended interactions with traffic agents at intersections from occurring, causing many of the scores reported above to misrepresent the true differences in the planners created by the experimental manipulations.

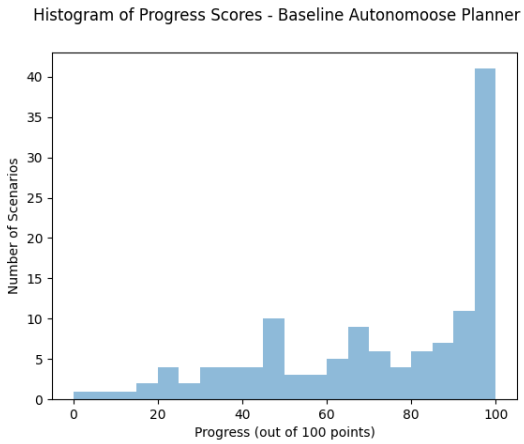
   On the other hand, our progress metrics demonstrate that the ego makes significantly quicker progress towards its target in each of the manipulation conditions compared to the baseline. This is also a direct consequence of the trigger limitations: the combination of reduced waiting times (or not waiting at the stop sign at all) and the lack of intended interactions with traffic agents at the intersection, causes the ego to approach the target quicker compared to the baseline condition. Thus, we are still able to see correct reflections of the actual situation that occurs in our benchmark scores, which allows us to diagnose the problem in the first place.

3. **Violating any of the requirements outlined for a motion planning benchmark will likely prevent a complete and accurate comparison of different motion planners.** To highlight this fact, we draw attention to the scenario creation process described in Chapter 5, in which each scenario contains a set of time and location-based trigger points, which are the only types of triggering mechanisms available. This triggering mechanism means that the scenario is optimized for a specific planning algorithm. Consequently, when we reduce the time spent at a stop sign or even eliminate it completely, the nature of the planning algorithm fundamentally changes and a subset of the intended interactions between the traffic agents and ego vehicle cannot be realized. For instance, a scenario in which the ego is supposed to stop (for a reduced period of time) and let traffic agents pass through the intersection, actually involves the ego driving through the intersection before the traffic agents even approach it. As a result, the objects are never encountered, and the metrics are unable to capture the intended behavioral differences in the experimental manipulations.

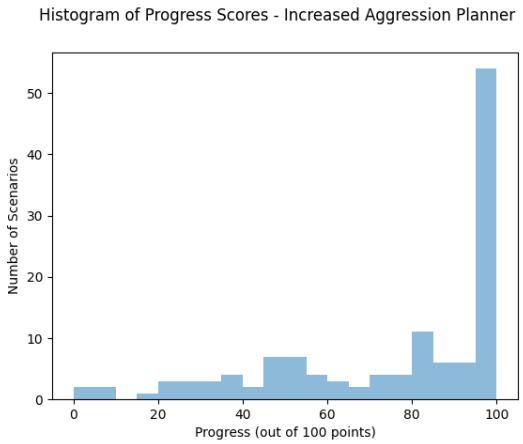
Besides the major takeaways outlined above, one of the more *unexpected* findings from Experiment 1 actually turns out to be the distributions for rule-abidance metrics; for each of the manipulations, a clear ceiling effect is observed, making it difficult to comprehend the true effect that each manipulation may have on these scores. This is, however, most likely a manifestation of the robust nature of the Autonomoose's rule-engine within its ODD, as lower rule-abidance scores only occur in rare circumstances, such as when the ego veers completely off-track or drives around in meaningless circles. These rare events are primarily due to the limitations in the physics of the simulation environment, rather than weaknesses in the planner itself.

Another potential limitation is the choice of the t-test to compare the manipulations and the baseline. Since the score distributions appear to be highly skewed, future works should apply a non-parametric test rather than the t-test.

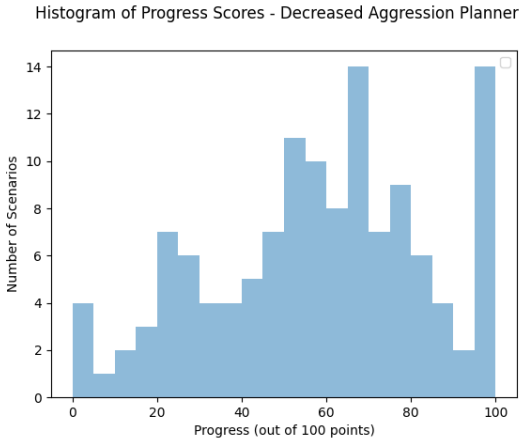
Thus, in light of the two experiments performed, and even the associated limitations, we believe that our complete motion planning benchmark framework is a beneficial tool for any application that requires comparisons between multiple motion planning algorithms, or even evaluations of individual approaches. Our metrics are able to discern differences between different motion planners, and are even able to highlight unexpected behavior caused due to limitations in the scenario creation tool, suggesting their potential as useful debugging tools. To elaborate, as a result of these metrics, we are now aware of some of the disadvantages of using time and location based triggering mechanisms in scenario creation tools. We now know that it would be far more beneficial to our use-case to represent scenarios using metric-based triggers instead of those that are simplistically time or location based. For example, triggers that are based on the ego's distance from the intersection would have been far more useful in such experiments, as they guarantee that the desired types of interactions will occur regardless of the underlying planner being used.



(a) Baseline

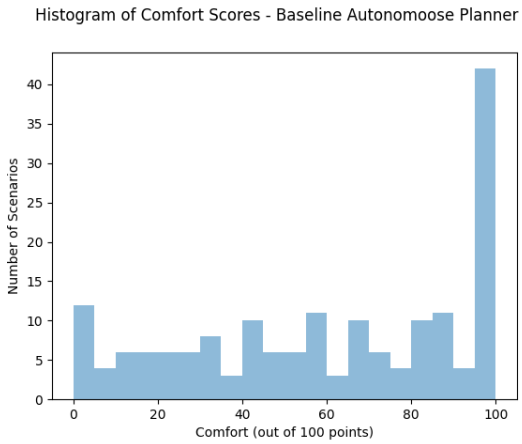


(b) Increased Aggression

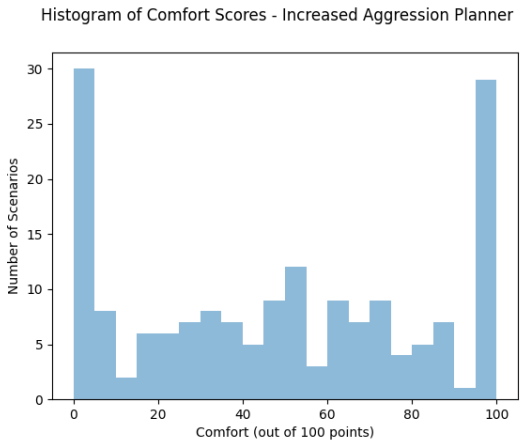


(c) Decreased Aggression

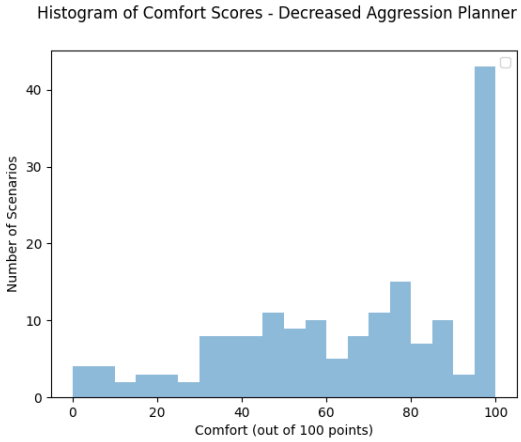
Figure 7.1: Experiment 1: distributions of Progress metrics for (a) baseline, (b) increased aggression, (c) decreased aggression.



(a) Baseline

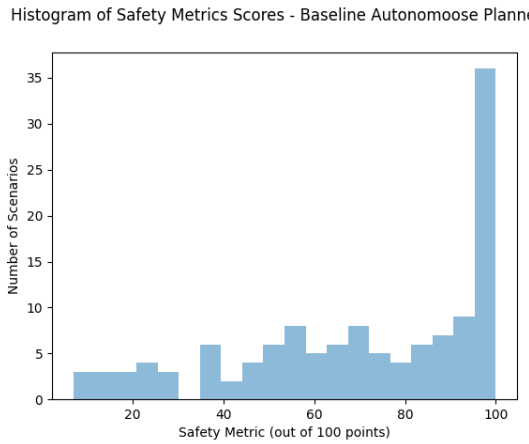


(b) Increased Aggression

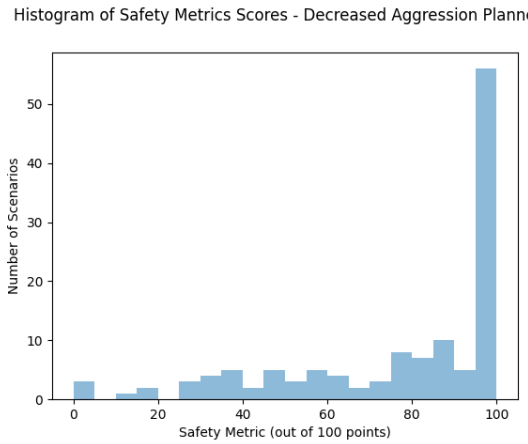


(c) Decreased Aggression

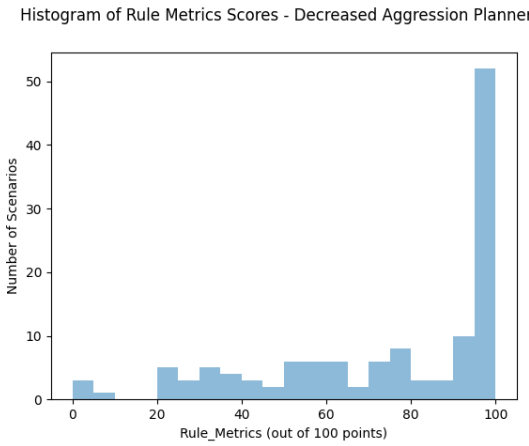
Figure 7.2: Experiment 1: distributions of Comfort metrics for (a) baseline, (b) increased aggression, (c) decreased aggression.



(a) Baseline

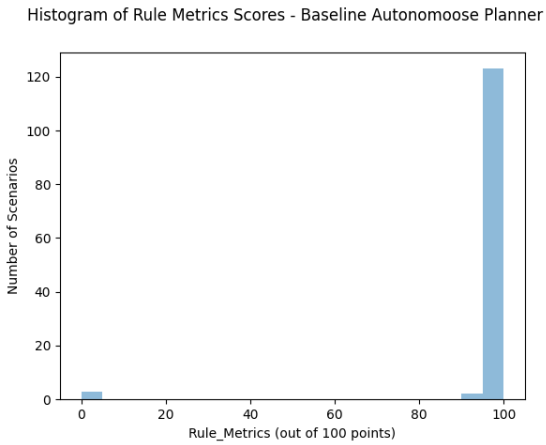


(b) Increased Aggression

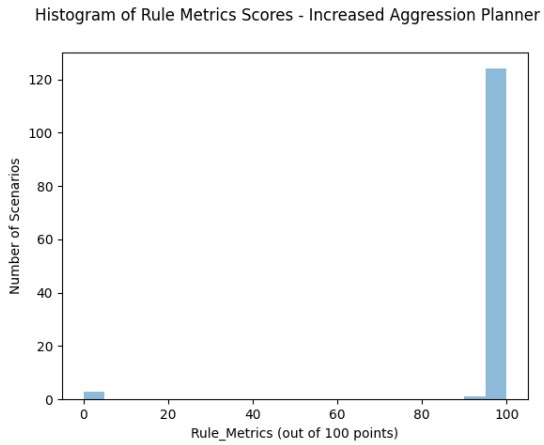


(c) Decreased Aggression

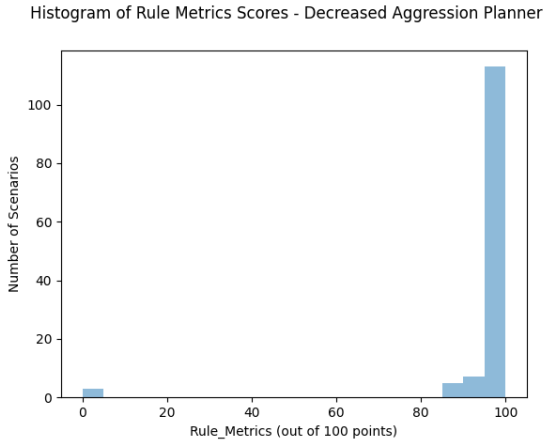
Figure 7.3: Experiment 1: distributions of Safety metrics for (a) baseline, (b) increased aggression, (c) decreased aggression.



(a) Baseline

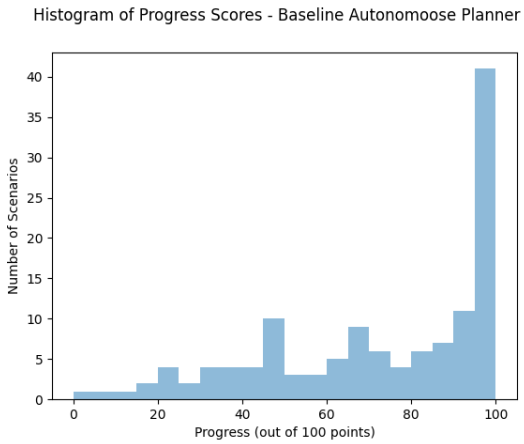


(b) Increased Aggression

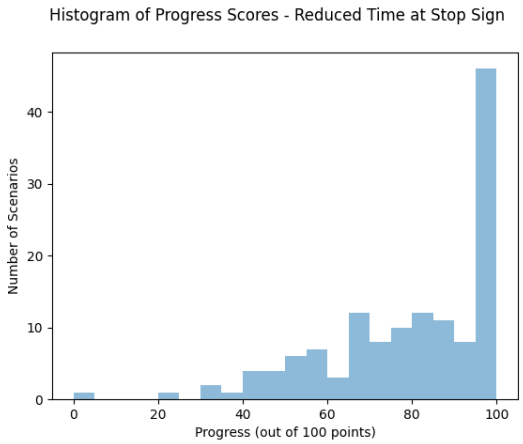


(c) Decreased Aggression

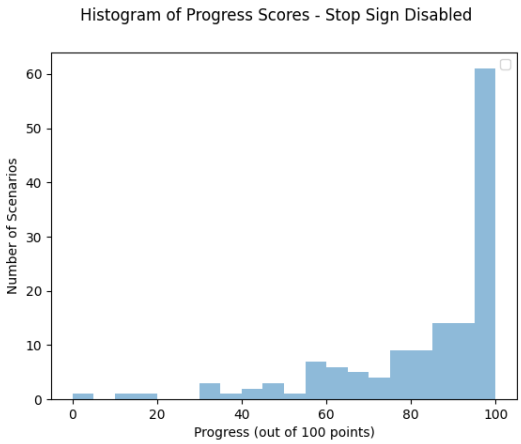
Figure 7.4: Experiment 1: distributions of Rule-abidance metrics for (a) baseline, (b) increased aggression, (c) decreased aggression.



(a) Baseline

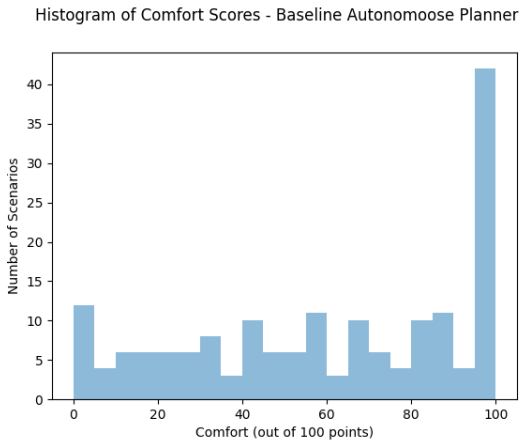


(b) Reduced time spent at stop sign

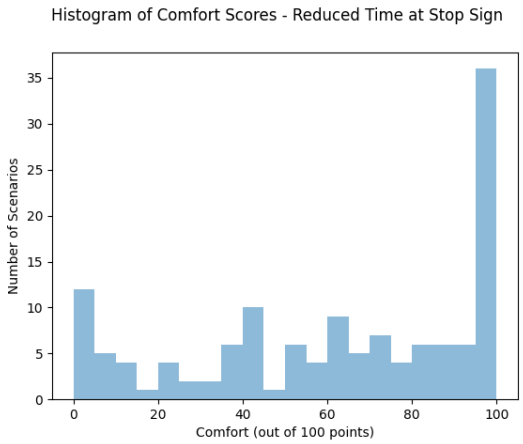


(c) Stop sign disabled

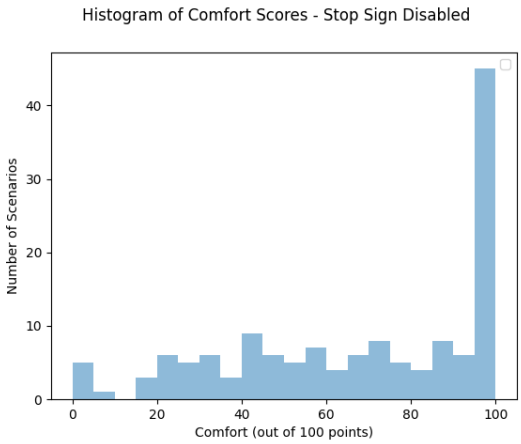
Figure 7.5: Experiment 2: distributions of Progress metrics for (a) baseline, (b) reduced time spent at stop sign, (c) stop sign disabled.



(a) Baseline

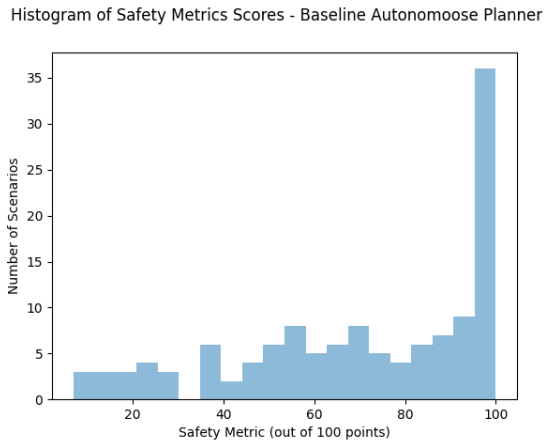


(b) Reduced time spent at stop sign



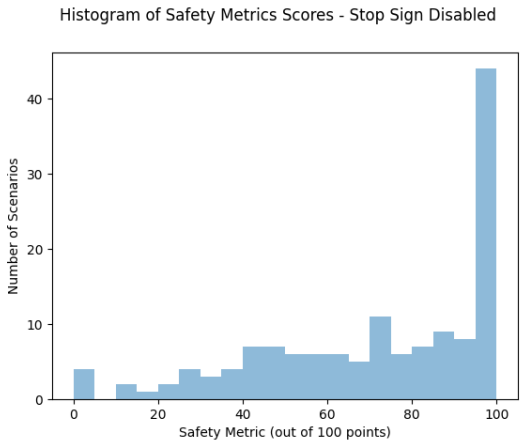
(c) Stop sign disabled

Figure 7.6: Experiment 2: distributions of Comfort metrics for (a) baseline, (b) reduced time spent at stop sign, (c) stop sign disabled.



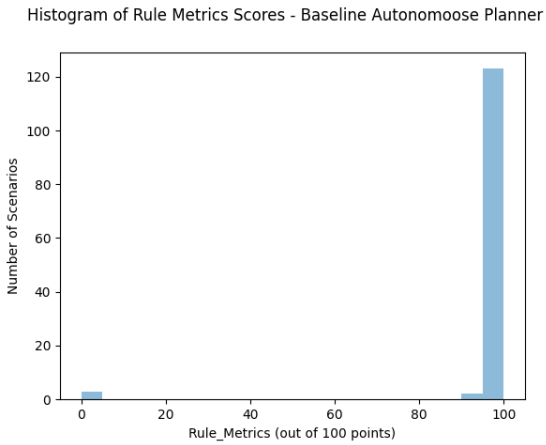
(a) Baseline

(b) Reduced time spent at stop sign

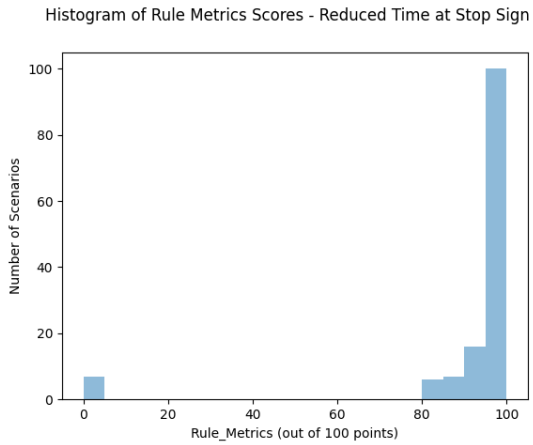


(c) Stop sign disabled

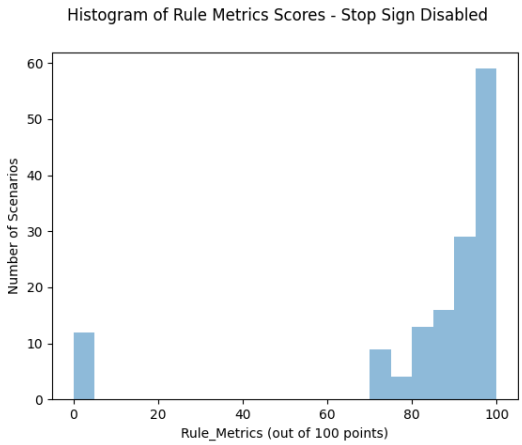
Figure 7.7: Experiment 2: distributions of Safety metrics for (a) baseline, (b) reduced time spent at stop sign, (c) stop sign disabled.



(a) Baseline



(b) Reduced time spent at stop sign



(c) Stop sign disabled

Figure 7.8: Experiment 2: distributions of Rule Abidance metrics for (a) baseline, (b) reduced time spent at stop sign, (c) stop sign disabled.

Chapter 8

Passenger Comfort

As discussed in Chapter 6, thresholds for passenger comfort in AVs has been a sparsely explored topic. In order to investigate this in more depth, we attempted to relate passenger comfort in a real AV with a set of comfort metrics. To do this, we ran an experiment in which we varied the driving profile of an AV to study its effect on passenger comfort: the study involved the use of physiological sensors, alongside self-reported scores and questionnaires, to gauge the comfort levels of the participants while the vehicle drove autonomously.

One of the original goals of this study was to derive a set of comfort metric thresholds, which could then be used in Chapter 6, to more accurately benchmark motion planning algorithms with respect to real passenger comfort. While this specific goal was not met, the study still provided invaluable insights towards and validation of our proposed set of comfort and safety metrics, and is thus discussed at a high level in this chapter, along with possible directions for future work. The complete study was a collaboration between Dillen and Ilievski, et al., and a detailed description of it can be found in our paper [34].

8.1 Background

In this section, we provide some background review on the study that may not have been adequately covered in the related work. We first discuss the state-of-the-art in passenger-vehicle interaction research, and then go on to briefly review the physiological measures of passenger comfort that were employed in the study.

8.1.1 Passenger-vehicle interaction

State-of-the-art research on passenger interaction with autonomous vehicles can be broadly grouped into three categories: physical autonomous vehicle studies, simulator studies, and Wizard of Oz studies.

Physical autonomous vehicle studies have been mostly conducted either in a lab (but with no physiological measurement) [95] or as a field study [89, 40]. Such studies are notoriously difficult to conduct in real life due to the complex nature of daily traffic and the interactions involved. For example, Mühl et al. [89] studied passenger trust in an autonomous vehicle and conducted an exploratory field study in an uncontrolled traffic environment, but investigated driving styles in a controlled simulated environment. Festner et al. [40], meanwhile, only studied lane-changing behaviour, and did not consider physiological responses in their experiment.

Simulator studies, on the other hand, have generally used lower to medium fidelity driving simulators which fail to account for the feeling of realistic physical forces and motion components, and present an inherent safety bias to participants. Scenarios are also limited to avoid causing simulator sickness. Passenger-vehicle interaction studies have primarily been conducted in simulators: some of these have investigated effects on physiological response [19, 20, 63, 101], while others rely solely on self-report scores [18, 21, 48, 107].

Wizard-of-Oz studies have been conducted in manually driven vehicles [38, 131], but these too suffer from shortcomings: they tend to rely solely on self-reported data and fail to provide a fully realistic experience (for example, screens may block passengers from seeing the steering wheel move autonomously).

8.1.2 Physiological sensing of emotional response

Comfort was defined as an abstraction for stress, anxiety, or frustration experienced. In this regard, the physiological responses were mainly used as an indicator of stress or (state) anxiety and, thus, an indirect measure of comfort. Several studies have used heart rate (HR) and heart rate variation (HRV), galvanic skin response (GSR), and eye movement patterns as indicators of anxiety.

HR has been used as a stress indicator in contexts ranging from noisy environments to dental surgery [60, 15], while low frequency (LF) components of HRV have been found to be directly proportional to elevated stress levels across multiple domains [91, 104, 77].

Physiological arousal, one of the effects of stress or anxiety, stimulates the secretion of sweat, which in turn raises the skin conductance level (SCL). Consequently, GSR, commonly measured through SCL and skin conductance response (quick burst of elevated SCL

levels, resembling peaks), is a popular measure of arousal [82] and has found applications in driving research [11] and studies on mental stress [24, 77].

Eye movement patterns have also been linked to stress as well as frustration, as in the case of driver monitoring [42, 72] and anxiety-induced distraction [125, 28]. In particular, eye movement entropy—the randomness in scan behavior—has been found to be elevated in high anxiety situations with high cognitive load [10].

8.2 Experimental Design and Set-up

The experiment was conducted on a closed test track in Waterloo, Ontario, Canada. Two vehicles were used for the study: a Lincoln MKZ installed with the Autonomoose platform - the self-driving technology stack developed by the University of Waterloo which has been referred to in Chapters 4 - and a single traffic agent, a Lexus 450 Rx which was driven manually. No other traffic agents or pedestrians were involved in this experiment. Although the vehicle was driven with autonomy engaged, a safety driver was present to take control in case of an emergency.

Ethics approval for the study was granted by a University of Waterloo Research Ethics Committee (ORE #40512).

8.2.1 Terminology

The main goal of the experiment was to study the effect of driving profile – or, more specifically, driving profile parameters – on passenger comfort, and to subsequently use the results to generate a set of comfort and safety thresholds for our comparison metrics.

The definition of driving profile followed for the study followed the defensive-aggressive paradigm where more defensive driving involves lower speeds, smaller accelerations, and smaller following distances. Although a particular profile can be constituted by several parameters such as speed, acceleration, following distance, and lane-changing parameters, only acceleration and distance were manipulated for this study. This was done to keep the experiment controlled and feasible.

8.2.2 Manipulations

In order to vary the overall driving profile, manipulations were made to individual driving parameters. More specifically, the thresholds for maximum acceleration and minimum distance from the traffic agent were varied in order to achieve two levels of aggressiveness: less

Table 8.1: We varied the thresholds for the lateral and longitudinal components of two parameters: acceleration and distance. Both components of each parameter were linked for a total of four different driving profiles: Low Acceleration Low Distance, Low Acceleration High Distance, High Acceleration Low Distance, and High Acceleration High Distance.

Parameter	**Less Aggressive**		**More Aggressive**	
	Long.	*Lat.*	*Long.*	*Lat.*
Acceleration	2.5 m/s^2	2 m/s^2	4 m/s^2	4 m/s^2
Distance	10 m	4.5 m	7.5 m	2 m

aggressive and more aggressive (Table 8.1). Both the lateral as well as longitudinal components of the two thresholds were manipulated. For both driving profiles, the threshold for speed was kept the same at 9.72 m/s, the maximum limit that could be safely handled by the autonomous driving system. In fact, even the final settings for the more aggressive profile reached the maximum safety limit.

Due to the experimental manipulations on 4 different thresholds (for each component of acceleration and distance), there would have been a total of 16 different overall driving profiles for each combination of thresholds. In order to keep the experiment feasible, we linked the lateral and longitudinal components for either of acceleration and distance. Thus, both components were either set to their more aggressive threshold, or both set to the less aggressive threshold.

Varying the thresholds while linking both components of acceleration and distance resulted in four overall driving profiles: *Low Acceleration Low Distance, Low Acceleration High Distance, High Acceleration Low Distance, and High Acceleration High Distance.* Each of these profiles was tested in a separate experimental trial for each participant in a within-subjects study design. The order in which the thresholds were varied for each trial was randomized for each participant.

It should be noted, however, that while the *thresholds* for acceleration and distance were varied, due to the continuously varying nature of these variables in a realistic driving scenario, the actual experimental analysis was performed on samples from the entire signal obtained for each of these variables. In addition, the derivatives of acceleration, namely, longitudinal and lateral jerk, were also considered.

8.2.3 Scenarios

In a realistic driving scenario, it is difficult to completely separate the effects of different driving profile parameters. For example, straight road driving will always involve small,

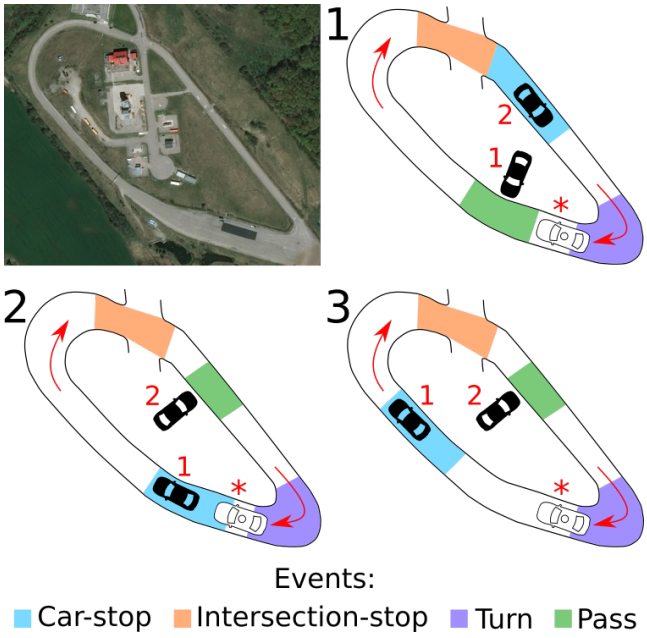


Figure 8.1: The layout of the test track. The order of scenarios for each trial, is indicated by the number beside the vehicle and the star represents the start and end location of the ego vehicles. Colored segments of the track represent the locations for the different scenarios: green for passing, orange for intersection-stop, blue for car-stop, and purple for turning scenarios. Trial 4 was a repetition of trial 1.

unintended lateral translations, resulting in some level of confounding.

In order to isolate the influence of each component of the driving profile parameters, each trial was divided into four different scenarios, as seen in Figure 8.1, with each scenario testing a different component of acceleration and distance:

1. **Passing a parked traffic agent.** This scenario was aimed at isolating the influence of lateral distance to a traffic agent and involved passing the parked traffic agent from the left. The traffic agent was oriented perpendicular to the ego vehicle.

2. **Stopping at an intersection.** This scenario isolated the effect of longitudinal acceleration wherein the ego vehicle was made to stop at a clear intersection. Although there was no other traffic agent present during this scenario, participants could sometimes see a traffic agent in the distance (when this scenario preceded the car-stop).

3. **Stopping behind a traffic agent.** In order to test the effect of longitudinal bumper-to-bumper distance to the traffic agent, i.e., the other car, the ego vehicle was made to stop behind it. Of course, this scenario also included the effect of longitudinal acceleration..

4. **Turning.** The lateral component of acceleration was tested through a turning scenario, in which the ego was made to take a sharp turn at the end of the track.

For the rest of this chapter, these scenarios will be referred to as "passing", "intersection-stop", "car-stop", and "turning" scenarios, respectively.

8.2.4 Study task

Throughout the course of the study, participants were asked to watch a neutral-themed video on a 5-inch display smartphone fixed to the dashboard. This is a common and natural activity among passengers in cars, and its purpose was to serve as an area of interest (as seen in the bottom panel of Figure 8.2) for measuring the eye movement entropy.

8.2.5 Participants

A total of 20 participants (10 Male, 10 Female) were recruited for the main experiment, aged between 19 to 64 years (Mean=33.5, S.D=3.52). Participants were recruited mainly from the University of Waterloo campus, and represented a diverse set of backgrounds, including education and familiarity with AVs.

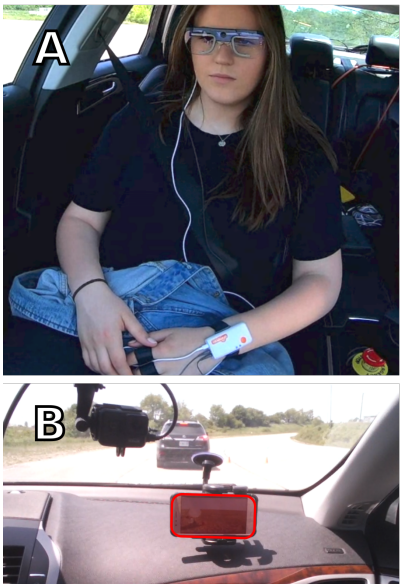


Figure 8.2: Participant (P12) fitted with sensors (top), and their view from the passenger seat (bottom). The area of interest around the phone screen is outlined in red.

8.2.6 Data collected

Collected data consisted of the set of **vehicle state signals**, i.e., the longitudinal and lateral components of acceleration and distance (as well as jerk), and the corresponding participant responses, measured using a combination of self-reported scores as well as physiological response.

Self-reported scores were measured by means of on-the-fly questions at the end of each scenario. Participant was asked to rate their perceived level of comfort on a scale of 1 to 10, 1 meaning not comfortable at all, and 10 meaning very comfortable. The purpose of these on-the-fly questions was to gauge the influence of each scenario on the participant's perceived comfort level irrespective of their actual physiological response.

Physiological responses, on the other hand, consisted of the participant's GSR, HR and HRV, and eye movement entropy.

8.2.7 Study Procedure

Each participant was first briefed about the experiment, and then given an information letter and consent form. After signing the consent form, the participant was fitted with

the sensors and asked to filled out a pre-study questionnaire. The video was started and participants were instructed to watch the video but to feel free to look up if they felt the need to.

Each trial commenced with eye tracker calibration. After each scenario, self-reported scores were collected, and after all four trials were complete, each participant was further interviewed for their general feedback on the experiment. In addition, participants were specifically asked to rank each scenario in order of increasing comfort. They were also given the opportunity to ask any questions about the autonomous vehicle or the experimental procedures.

8.3 Signal Processing

This section describes the method of collection for the various vehicle state and physiological signals and any additional signal processing that was required for the purpose of the experimental analyses. The section on physiological signals also provides a brief overview on each signal for the benefit of the reader. As vehicle state signals have already been discussed in detail in Chapter 6, we provide only a brief overview of these signals, along with subtle differences in their method of collection with respect to our previously conducted simulator experiments.

8.3.1 Vehicle state signals (metrics)

Vehicle state signals were sampled at 20 Hz and were measured using the car-mounted inertial measurment unit (IMU), Global Positioning System (GPS), Light-Detection-and-Ranging (LIDAR) sensors, and vision sensors fixed to the Autonomoose. These signals can be seen in the middle panel of Figure 8.3.

Acceleration

Acceleration was measured using the IMU unit. The raw acceleration signal was filtered using a second-order Butterworth filter to remove inherent signal noise, and then resolved into its lateral and longitudinal components for subsequent analysis. For more information refer to Chapter 6.

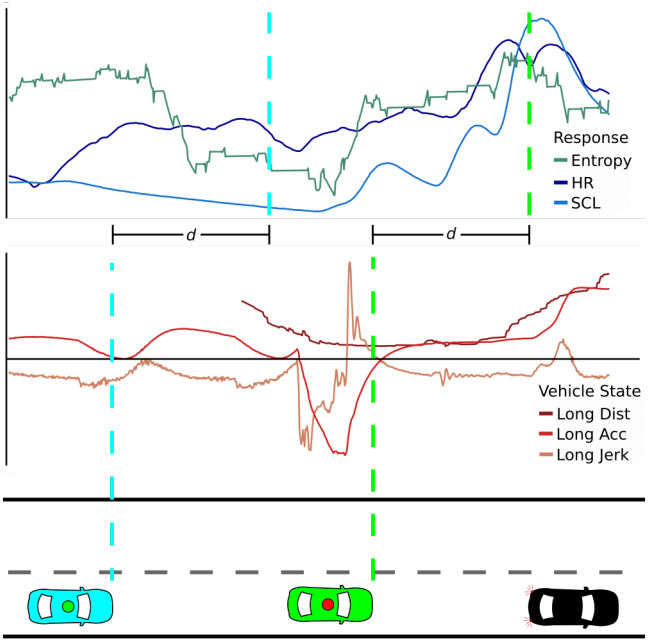


Figure 8.3: A car-stop scenario (bottom) that occurred in one of P19's trials. The topmost panel depicts the physiological response to the scenario, and the middle panel represents the corresponding vehicle state signals. There is an inherent delay associated with the physiological response.

Jerk

The first-order derivative of both acceleration components was used to calculate the jerk signals, which were further filtered to reduce signal noise using a mean filter. The filter used a window size of 2 s. For more information refer to Chapter 6.

Distance

Contrary to the simulator set-up which provided ground truths for traffic agent locations and bounding boxes, all surrounding traffic agents (in this case, only the other vehicle) were detected using the AVOD [67] algorithm. The AVOD algorithm combines data from the vision and LIDAR sensors to produce traffic agent bounding boxes in KITTI format [44]. Kalman filters are then used, as in the simulator experiment, to track each detected bounding box. However, while the algorithm is relatively robust, further manual processing was sometimes necessary for the purpose of the experimental analyses. This manual processing was primarily done in cases of mis-detection and involved forcefully constructing the correct bounding boxes based on recorded image and LIDAR data. As a result, the most accurate positioning of the traffic agent was always guaranteed for the analyses.

The bumper-to-bumper Euclidean distance was calculated between the the closest points for the ego and traffic agent bounding boxes, and was resolved into the lateral and longitudinal components. For more information refer to Chapter 6.

8.3.2 Participant physiological response

A Shimmer3+ device was used to measure the participant's GSR by means of two electrodes wrapped around the index and middle fingers of their non-dominant hand. The device was also used to obtain a photoplethysmogram (PPG) signal (used for measuring HR and HRV) by means of an optical ear clip. Eye movement patterns were measured using a Tobii Glasses 2 Pro eye tracking device. Participant set-up can be seen in the top panel of Figure 8.2. The GSR and PPG signals were sampled at 512 Hz, while the eye tracker operated at 50 Hz. Physiological signals are depicted in the topmost panel of Figure 8.3.

GSR

The GSR variables used in the analysis were SCL, the number of peaks, and the maximum peak amplitude. The raw GSR signal was de-trended to remove the baseline and obtain

the SCL signal. Peaks were then detected from the SCL signal using a median filtering technique [53].

HR and HRV

HR was derived from local maxima in the raw PPG signal over sliding windows of 10 s. For HRV, the inter-beat interval (IBI) series was calculated from the local maxima, and transformed to the frequency domain using the Fourier transform. HRV was represented through the normalized low frequency (0.04–0.15 Hz) and high frequency (0.15–0.4 Hz) components, along with the ratio between them.

Eye movement entropy

Fixations from the detected gaze patterns were detected using the iMotions software tool [100]. Fixation sequences were used to calculate the AOI transition matrix, which was then combined with prior probabilities for the AOIs to calculate the eye movement entropy. The priors for each AOI were calculated based on dwell times.

8.3.3 Signal synchronization

While vehicle signals were sampled at 20 Hz, GSR and eye-tracking signals were sampled at 512 Hz and 50 Hz respectively. The extracted SCL, HR, and entropy signals were down-sampled collectively at 20 Hz using linear interpolation. However, physiological and vehicle state signals were measured on two different systems with different system clocks. To solve this problem, video data on each system was used to manually label and align common events in time, followed by nearest neighbour interpolation which was used to synchronize and combine all signals at 20 Hz.

8.4 Analyses

Through this experiment, we tried to determine how a subset of the metrics defined in Chapter 6, relates to passenger comfort inside an AV. In order to do this we ran two separate analyses: one across driving profile parameters, and the second across self-reported scores. (The original study had a third analysis across scenarios, but the details are not relevant to this thesis). All analyses were carried out using linear mixed-effects (LME) models.

8.4.1 Analysis I: Driving profile parameters

We regressed each physiological response variable against all vehicle state predictor variables. The analysis was carried out at multiple levels of aggregation, ranging from the entire trial itself, to the individual samples in a time-series analysis.

The physiological and vehicle state signals were aggregated into their maximum and mean values over the interval considered in each level of aggregation. Of course, in the time-series level there was no aggregation involved. Aggregation using mean and maximum values was done to capture the average and extreme effects, respectively. Where applicable, maximum valued response variables were regressed against the corresponding maximum valued vehicle state variables. The same procedure was applied for the mean valued variables.

The levels of aggregation considered included:

1. **The trial level** where aggregations were performed over the entire trial
2. **The scenario level** where aggregations were performed over each scenario (note that non-scenario samples were not considered)
3. **The window level** where aggregations were performed over a non-overlapping sliding window. Each window was spatially aligned across all participants and was 50 m in length, representing an approximate duration of 5 s for an average ego speed of about 10 m/s. This was purposely designed to accommodate the inherent lag in the physiological responses which is maximum for GSR (in the order of 1–5 s).
4. **The time-series level** where the individual samples themselves were considered.

Response and predictor variables

For the trial and window levels of analysis, the response variables considered were the mean SCL, maximum peak amplitude, number of GSR peaks, mean and maximum HR, and mean and maximum entropy. The trial level also included HRV response variables, while the time-series level included only the original SCL, HR, and entropy signals.

The predictor variables from the vehicle state signals included the lateral and longitudinal components of acceleration and jerk (a_{long}, a_{lat} and j_{long}, j_{lat}, respectively) and the presence of a lead and parked vehicle ($pres_{lead}$ and $pres_{pass}$, respectively). As before, maximum and means were used when aggregation was necessary, while original values were used when analyzing the time-series.

Separate analyses were conducted on the minimum (or absolute, in case of time-series) lateral and longitudinal distances (d_{lat} and d_{long}, respectively). These were performed only on the subset of the samples in which the lead or parked vehicle was present.

8.4.2 Analysis II: Self-reported scores

On-the-fly scores of comfort were regressed against physiological responses in order to determine if there was indeed a relationship between the two. We performed two regressions: one against all mean valued response variables and the number of peaks, and one against all maximum valued response variables and the number of peaks. Variables were aggregated over scenario level intervals.

8.5 Results

All analyses used a significance level of .05. While multiple driving profile parameters were tested in the analysis, for brevity, we report and comment on only the terms that were significant.

8.5.1 Analysis I: Driving Profile Parameters

The results of analysis I for the window level can be found in Table 8.2. The trial and time-series level of analyses proved to be inconclusive due to contradictory results within each level of analysis, possibly due to too large and too small interval sizes. There were three major findings relevant to this thesis:

1. **Longitudinal acceleration and jerk significantly affected physiological response.** This is reflected in both the quantitative effects on the maximum peak amplitude, number of peaks, mean SCL, and maximum HR variables, as well in the qualitative interview feedback from participants. In particular, participants talked about the "jerkiness" in breaking and the "jittery" driving, and even compared the driving to "when my kids were learning how to drive".

   Interestingly, the positive direction of acceleration and jerk seemed to have a greater effect than the negative direction. For example, on average, the maximum HR increased by more than 3.5 BPM for every m/s^2 increase in longitudinal acceleration.

2. **The presence and proximity of a lead vehicle affected physiological response.** The presence of a lead vehicle positively affected GSR, HR, as well as

Table 8.2: Regression coefficients and confidence intervals for all significant predictors at the window level. Due to space constraints only the maximum valued responses are shown; *, **, and *** indicate p-values less than 0.05, 0.01, and 0.001 respectively.

Predictor (max)	Max. Pk. Amp. ($b \pm 95\%CI$)	Num. Pks. ($b \pm 95\%CI$)	Max. HR ($b \pm 95\%CI$)	Max. Entropy ($b \pm 95\%CI$)
$a_{long}-$	–0.025 ±0.037 ∗∗	–0.081±0.085 ∗∗∗	–1.949±2.964 ∗∗	
$a_{long}+$	0.087±0.042 ∗∗	0.246±0.094 ∗∗∗	3.596±2.645 ∗∗	–
$j_{long}-$	–0.026±0.043 ∗∗	–0.068±0.095 ∗∗∗		
$j_{long}+$	0.088±0.039 ∗∗	0.27±0.09 ∗∗∗	–	–
$pres_{lead}$	0.12±0.072 ∗∗	0.432±0.161 ∗∗∗	6.546±4.223 ∗∗	0.014±0.012 ∗
d_{long}	–	-0.014±0.013∗	–	–

entropy alike. The maximum HR went up, on average, by more 6.5 BPM, while the maximum peak amplitude went up by 0.12 μS. Proximity (i.e., the longitudinal bumper-to-bumper distance), on the other hand, affected only the number of peaks response variable, where every 100 m increase in proximity generated an additional 1.4 GSR peaks.

3. **The presence (and to a lesser extent, the proximity) of a lead vehicle moderated the effect of longitudinal acceleration and jerk.** Lead vehicle presence resulted in an exaggeration in the effect of acceleration and jerk on passenger response. The effect of longitudinal acceleration was exaggerated for the mean SCL ($b = 0.207 \pm 0.154, p < .01$), maximum peak amplitude ($b = 0.247 \pm 0.188, p < .05$) and number of peaks ($b = 1.130 \pm 0.417, p < .001$). With longitudinal jerk, the effect was significant for the number of peaks ($b = 0.504 \pm 0.309, p < .01$) and maximum HR ($b = 9.115 \pm 8.029, p < .05$).

   Weak ($p < .1$) to nearly significant ($p \approx .05$) effects between *proximity* and longitudinal acceleration or jerk were found for the number of peaks, mean or maximum HR, and mean entropy.

8.5.2 Analysis II: Self-reported scores

The results on self-reported scores can be found in Table 8.3. We found that GSR—specifically, the mean SCL and number of peaks response variables—was a predictor of

comfort. For instance, for every additional GSR peak, the self-reported comfort score decreased by about one-fourth of a point. Similarly, a 1 μS increase in the mean SCL corresponded to nearly a 1-point decrease in the comfort score.

These results tell us that the self-reported scores and physiological responses were indeed correlated, and thus help us further relate the results from analysis I not just to the participant's physiological response, but also to their self-reported comfort.

8.6 Discussion

In this chapter, we described a real-world AV study in which we investigated passenger comfort as a physiological response to various driving profile parameters. These parameters, namely, acceleration and jerk, and distance to a traffic agent corresponded directly to our *comfort* and *safety* comparison metrics respectively.

This study awarded us with two key takeaways:

1. **Insight into how a motion planning algorithm's driving profile parameters might influence passenger comfort:** the presence of external interacting agents has a direct effect on passenger physiological response, and hence, their comfort inside an AV. The effect of longitudinal acceleration and jerk on passenger response, was magnified by both the presence of a lead vehicle, and its bumper-to-bumper distance from the ego. Thus, we see that while a passenger might have individual preferences for the acceleration and jerk profile employed by the vehicle, the magnitude of these preferences can vary depending on the presence or absence of another interacting agent.

2. **Validation of our choice of comfort and safety metrics:** passengers generally complained about the "jerkiness" in driving and breaking at times, citing it as a direct source of discomfort. This serves as a real-life affirmation in our choice of acceleration and jerk as comfort metrics, a result that we would not have otherwise confirmed from our simulator study alone. Furthermore, the exaggerated effect of acceleration and jerk in the presence of a lead vehicle, tells us that our safety metrics (in this case, longitudinal "bumper-to-bumper" distance) also have a valid role to play in the benchmarking process.

These results bring us closer to understanding human thresholds for each set of comparison metrics. Whether it be in regard to safety or comfort, this experiment highlights how the combination of these two sets of metrics provides insight towards the overall level of passenger comfort inside an AV.

Table 8.3: Regression coefficients and confidence intervals for significant physiological response predictors for self-reported comfort; *, **, and *** indicate p-values less than 0.05, 0.01, and 0.001 respectively.

Predictors	Num. Pks.	Mean SCL
Comfort (b $\pm 95\% CI$)	–0.245***±0.113	–0.934*±0.837

There were, however, some limitations associated with this work. In the context of this thesis, while this experiment went a long way in demonstrating how the comparison metrics proposed in Chapter 6 may be indicative of comfort, significant work is yet to be done to establish the actual thresholds for these metrics. Towards this end, a more involved study should be conducted with metrics at different levels of granularity to provide further insight into the thresholds that should be established.

Other limitations also exist, mostly with respect to the experimental implementation: because multiple variables must always work together to execute the task of driving, some amount of confounding between driving profile parameters was unavoidable, despite our best efforts to isolate them as much as possible. Furthermore, external factors, such as manual takeovers or unexpected events, also sometimes occurred and could have influenced physiological responses to some extent. Finally, state-of-the-art limitations prevented us from running the experiment on a real road reducing its validity. The presence of a safety driver too could have actually established a larger sense of security in passengers and driven the responses down to some extent.

Future work should, therefore, serve to address these implementation level limitations, and should explore driving profile parameters in greater detail so as to establish a more concrete set of thresholds that allow for better comparisons between different motion planning approaches.

Chapter 9

Conclusion

In this thesis, we propose a novel framework for benchmarking motion planning algorithms for Autonomous Vehicles. We first review and outline an extensive set of requirements that should be able to guide the development of any future motion planning benchmark tools in any ODD. These requirements encapsulate several dimensions of the benchmarking problem, from the simulation environment and representative scenario suite used to generate reproducible results, to the comparison methodology that should be employed to thoroughly compare and contrast multiple algorithms.

In accordance with our proposed set of requirements, we present our own comprehensive scenario creation methodology along with a set of robust comparison metrics that evaluates motion planning algorithms across 4 dimensions: comfort, progress, safety, and rule abidance. We then implement *WiseBench*, a proof of concept motion planning benchmark that incorporates the scenario creation methodology and comparison metrics, and subsequently use it to run benchmarking experiments on the University of Waterloo Autonomoose platform.

We later show that our framework is able to distinguish between the subtle tradeoffs employed by different motion planning configurations. Furthermore, while certain limitations in the scenario creation tool causes unexpected distortions to the manner in which some scenarios play out and prevents certain behavioral capabilities from being distinguished, our benchmark is still representative of the actual situations that take place. In addition to this, we also run a real world experiment that sheds further light on the validity of our proposed metrics to effect human comfort. While this experiment does not highlight exact thresholds it brings us closer to understanding the human-centred thresholds that should be employed for each set of comparison metrics.

We thus believe that the proposed set of metrics will be beneficial to many real world

applications. For instance, the metrics could be used in a real-time tracking module for autonomous motion planning algorithms to track progression over successive versions of the algorithm. They may also be employed for logging and diagnostic purposes throughout the entirety of trip; combined with a correct set of thresholds, these metrics can illuminate the quality of an autonomous motion planning algorithm in real time. Finally, the comparison metrics would be able to serve as a core component in an adaptive driving system where, once again combined with human thresholds and unobtrusive physiological sensing, they will be able to shed light on when the driving profile must be altered in order to promote a sense of comfort among passengers riding in the autonomous vehicle.

Among the limitations associated with our framework and experimental approach is the level of simulator fidelity or, more specifically, the accuracy of the physics model used, agent behavior, and the initial conditions prevalent for each scenario. There are also hindrances caused due to the nature of triggers used in the scenario creation tool; specifically, location and time based trigger points that are tailored to the baseline planner prevent the intended interactions with traffic agents from occurring when certain planner configurations are changed. As a result, the true behavioral capabilities between the manipulated planners cannot be uncovered by the benchmark. Limitations also exist in terms of the available literature and research conducted on autonomous vehicle thresholds; while our real-life experiment takes us a few steps closer to closing the gap in this research, we are still unable to achieve a definitive set of thresholds as this is out of scope of the experiment. Furthermore, the scoring mechanism still lacks an agreed upon set of weightings to aggregate the set of metrics.

Future work should focus on addressing the limitations outlined: more accurate models should be used to achieve a higher level of simulator fidelity and, hence, more realistic results, while more robust triggering mechanisms should be used in the scenario creation tool. Extensive research and experimentation should also be carried out towards developing a set of concrete human-centered thresholds for each of each of the metrics used in this framework and beyond. In addition, an extended set of metrics should be developed to track higher fidelity rules – this may go hand in hand with addressing the limitations associated with simulator fidelity. Finally, an adequate set of weightings for each metric should be developed to accurately create a combined score of overall performance for the motion planning algorithms being benchmarked. These can initially be established through a consensus of expert opinions, and can be further developed through data driven approaches.

References

[1] Applied intuition. https://www.appliedintuition.com/.

[2] The most powerful real-time 3d creation platform. https://www.unrealengine.com/en-US/.

[3] Ros - powering the world's robots. http://www.ros.org/.

[4] Wise sim. https://uwaterloo.ca/waterloo-intelligent-systems-engineering-lab/projects/wise-sim, journal=Waterloo Intelligent Systems Engineering Lab, year=2020, month=May.

[5] Nvidia drive constellation virtual reality autonomous vehicle validation platform. https://www.nvidia.com/content/dam/en-zz/Solutions/self-driving-cars/drive-constellation/nvidia-drive-constellation-datasheet-2019-oct.pdf, October 2019.

[6] Voyage deepdrive. https://deepdrive.voyage.auto/, 2019.

[7] The carla autonomous driving challenge. https://carlachallenge.org/, 2020.

[8] Olzhas Adiyatov and Huseyin Atakan Varol. A novel rrt*-based algorithm for motion planning in dynamic environments. In *2017 IEEE International Conference on Mechatronics and Automation (ICMA)*, pages 1416–1421, 2017.

[9] Yasuhiro Akagi, Ryosuke Kato, Sou Kitajima, Jacobo Antona-Makoshi, and Nobuyuki Uchida. A risk-index based sampling method to generate scenarios for the evaluation of automated driving vehicle safety. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 667–672. IEEE, 2019.

[10] Jonathan Allsop, Rob Gray, Heinrich H Bülthoff, and Lewis Chuang. Eye movement planning on single-sensor-single-indicator displays is vulnerable to user anxiety and cognitive load. *Journal of Eye Movement Research*, 10(5):8–1, 2017.

[11] Georg W Alpers, Frank H Wilhelm, and Walton T Roth. Psychophysiological assessment during exposure in driving phobic patients. *Journal of Abnormal Psychology*, 114(1):126, 2005.

[12] Matthias Althoff, Markus Koschi, and Stefanie Manzinger. Commonroad: Composable benchmarks for motion planning on roads. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 719–726. IEEE, 2017.

[13] Michael Ardelt, Constantin Coester, and Nico Kaempchen. Highly automated driving on freeways in real traffic using a probabilistic framework. *IEEE Transactions on Intelligent Transportation Systems*, 13(4):1576–1585, December 2012.

[14] Andrew Bacha, Cheryl Bauman, Ruel Faruque, Michael Fleming, Chris Terwelp, Charles Reinholtz, Dennis Hong, Al Wicks, Thomas Alberi, David Anderson, et al. Odin: Team VictorTango's entry in the DARPA urban challenge. *Journal of field Robotics*, 25(8):467–492, 2008.

[15] Carol F Baker. Discomfort to environmental noise: Heart rate responses of sicu patients. *Critical Care Nursing Quarterly*, 15(2):75, 1992.

[16] Mayank Bansal, Alex Krizhevsky, and Abhijit S. Ogale. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. http://arxiv.org/abs/1812.03079, 2018.

[17] Hannah Bast, Daniel Delling, Andrew Goldberg, Matthias Müller-Hannemann, Thomas Pajor, Peter Sanders, Dorothea Wagner, and Renato F Werneck. Route planning in transportation networks. In *Algorithm engineering*, pages 19–80. Springer, 2016.

[18] Chandrayee Basu, Qian Yang, David Hungerman, Mukesh Singhal, and Anca D. Dragan. Do you want your autonomous car to drive like you? In *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, HRI '17, pages 417–425, New York, NY, USA, 2017. ACM.

[19] Matthias Beggiato, Franziska Hartwich, and Josef Krems. Using smartbands, pupillometry and body motion to detect discomfort in automated driving. *Frontiers in human neuroscience*, 12, 2018.

[20] Matthias Beggiato, Franziska Hartwich, and Josef Krems. Physiological correlates of discomfort in automated driving. *Transportation research part F: traffic psychology and behaviour*, 66:445–458, 2019.

[21] Hanna Bellem, Barbara Thiel, Michael Schrauf, and Josef F Krems. Comfort in automated driving: An analysis of preferences for different automated driving styles and their dependence on personality traits. *Transportation research part F: traffic psychology and behaviour*, 55:90–100, 2018.

[22] Asher Bender, James R. Ward, Stewart Worrall, and Eduardo M. Nebot. Predicting driver intent from models of naturalistic driving. In *Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on*, pages 1609–1615, Sept 2015.

[23] Philipp Bender, Julius Ziegler, and Christoph Stiller. Lanelets: Efficient map representation for autonomous driving. In *2014 IEEE Intelligent Vehicles Symp. (IV)*, pages 420–425, June 2014.

[24] Jens Blechert, Marta Lajtman, Tanja Michael, Jürgen Margraf, and Frank H Wilhelm. Identifying anxiety states using broad sampling and advanced processing of peripheral physiological information. *Biomedical sciences instrumentation*, 42:136–141, 2006.

[25] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D. Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Jake Zhao, and Karol Zieba. End to end learning for self-driving cars. http://arxiv.org/abs/1604.07316, 2016.

[26] Sebastian Brechtel, Tobias Gindele, and Rudiger Dillmann. Probabilistic decision-making under uncertainty for autonomous driving using continuous POMDPs. *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 392–399, Oct 2014.

[27] Martin Buehler, Karl Iagnemma, and Sanjiv Singh. *The DARPA urban challenge: autonomous vehicles in city traffic*, volume 56. springer, 2009.

[28] Joe Causer, Paul S Holmes, Nickolas C Smith, and A Mark Williams. Anxiety, movement kinematics, and visual attention in elite-level performers. *Emotion*, 11(3):595, 2011.

[29] Sachin Chitta, Ioan Sucan, and Steve Cousins. Moveit![ros topics]. *IEEE Robotics & Automation Magazine*, 19(1):18–19, 2012.

[30] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The

cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[31] Krzysztof Czarnecki. Automated driving system (ads) high–level quality requirements analysis– driving behavior safety. *Waterloo Intelligent Systems Engineering Lab (WISE) Report, University of Waterloo*, 2018.

[32] Krzysztof Czarnecki. Operational world model ontology for automated driving systems–part 1: Road structure. *Waterloo Intelligent Systems Engineering Lab (WISE) Report, University of Waterloo*, 2018.

[33] Krzysztof Czarnecki. Wise drive: Requirements analysis framework for automated driving systems, Jan 2020.

[34] Nicole Dillen, Marko Ilievski, Edith Law, Lennart E. Nacke, Krzysztof Czarnecki, and Oliver Schneider. Keep calm and ride along: Passenger comfort and anxiety as physiological responses to autonomous driving styles. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI 20, page 113, New York, NY, USA, 2020. Association for Computing Machinery.

[35] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. *arXiv preprint arXiv:1711.03938*, 2017.

[36] Automated Driving. Levels of driving automation are defined in new sae international standard j3016: 2014. *SAE International: Warrendale, PA, USA*, 2014.

[37] Ahmad El Sallab, Mohammed Abdou, Etienne Perot, and Senthil Yogamani. Deep reinforcement learning framework for autonomous driving. *Electronic Imaging*, (19):70–76, 2017.

[38] Mohamed Elbanhawi, Milan Simic, and Reza Jazar. In the passenger seat: investigating ride comfort measures in autonomous cars. *IEEE Intelligent Transportation Systems Magazine*, 7(3):4–17, 2015.

[39] Haoyang Fan, Fan Zhu, Changchun Liu, Liangliang Zhang, Li Zhuang, Dong Li, Weicheng Zhu, Jiangtao Hu, Hongye Li, and Qi Kong. Baidu apollo em motion planner. *arXiv preprint arXiv:1807.08048*, 2018.

[40] Michael Festner, Alexandra Eicher, and D Schramm. Beeinflussung der komfort- und sicherheitswahrnehmung beim hochautomatisierten fahren durch fahrfremde tätigkeiten und spurwechseldynamik. *von*, 11, 2017.

[41] Emilio Frazzoli, Munther Dahleh, and Eric Feron. Real-time motion planning for agile autonomous vehicles. In *Proceedings of the 2001 American Control Conference. (Cat. No.01CH37148)*, volume 1, pages 43–49 vol.1, 2001.

[42] Lex Fridman, Heishiro Toyoda, Sean Seaman, Bobbie Seppelt, Linda Angell, Joonbum Lee, Bruce Mehler, and Bryan Reimer. What can be predicted from six seconds of driver glances? In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 2805–2813. ACM, 2017.

[43] Enric Galceran, Alexander G. Cunningham, Ryan M. Eustice, and Edwin Olson. Multipolicy decision-making for autonomous driving via changepoint-based behavior prediction: Theory and experiment. *Autonomous Robots*, 41(6):1367–1382, Aug 2017.

[44] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, June 2012.

[45] Siegfried Johannes Gerathewohl. *Fidelity of simulation and transfer of training: a review of the problem.* US Department of Transportation, Federal Aviation Administration, Office of , 1969.

[46] Douglas Gettman, Lili Pu, Tarek Sayed, Steven Shelby, and ITS Siemens. Surrogate safety assessment model and validation. Technical report, United States. Federal Highway Administration. Office of Safety Research and , 2008.

[47] David González, Joshué Pérez, Vicente Milanés, and Fawzi Nashashibi. A review of motion planning techniques for automated vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 17(4):1135–1145, 2015.

[48] Franziska Hartwich, Matthias Beggiato, and Josef F Krems. Driving comfort, enjoyment and acceptance of automated driving–effects of drivers age and driving style familiarity. *Ergonomics*, 61(8):1017–1032, 2018.

[49] Hosking, Bryce Antony. Modelling and model predictive control of power-split hybrid powertrains for self-driving vehicles, 2018.

[50] Xinyu Huang, Xinjing Cheng, Qichuan Geng, Binbin Cao, Dingfu Zhou, Peng Wang, Yuanqing Lin, and Ruigang Yang. The apolloscape dataset for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 954–960, 2018.

[51] Constantin Hubmann, Jens Schulz, Marvin Becker, Daniel Althoff, and Christoph Stiller. Automated driving in uncertain environments: Planning with interaction and uncertain maneuver prediction. *IEEE Transactions on Intelligent Vehicles*, 3(1):5–17, 2018.

[52] Marko Ilievski, Sean Sedwards, Ashish Gaurav, Aravind Balakrishnan, Atrisha Sarkar, Jaeyoung Lee, Frédéric Bouchard, Ryan De Iaco, and Krzysztof Czarnecki. Design space of behaviour planning for autonomous driving. *arXiv preprint arXiv:1908.07931*, 2019.

[53] iMotions. Galvanic skin response (gsr): The complete pocket guide. https://imotions.com/blog/galvanic-skin-response/, 2016. Accessed: 2019-09-11.

[54] Mark Moll Ioan A. ucan. Create a benchmark configuration file.

[55] Thomas Fridolin Iversen and Lars-Peter Ellekilde. Benchmarking motion planning algorithms for bin-picking applications. *Industrial Robot: An International Journal*, 2017.

[56] Gregory Kahn, Adam Villaflor, Vitchyr Pong, Pieter Abbeel, and Sergey Levine. Uncertainty-aware reinforcement learning for collision avoidance. http://arxiv.org/abs/1702.01182, 2017.

[57] Rahul Kala and Kevin Warwick. Multi-level planning for semi-autonomous vehicles in traffic scenarios based on separation maximization. *Journal of Intelligent & Robotic Systems*, 72(3-4):559–590, 2013.

[58] Sertac Karaman and Emilio Frazzoli. Incremental sampling-based algorithms for optimal motion planning. *Robotics: Science and Systems VI*, 2010.

[59] Sertac Karaman and Emilio Frazzoli. Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research*, 30(7):846–894, 2011.

[60] Aaron Katcher, Herman Segal, and Alan Beck. Comparison of contemplation and hypnosis for the reduction of anxiety and discomfort during dental surgery. *American Journal of Clinical Hypnosis*, 27(1):14–21, 1984.

[61] Christoph G Keller, Thao Dang, Hans Fritz, Armin Joos, Clemens Rabe, and Dariu M Gavrila. Active pedestrian safety by automatic braking and evasive steering. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):1292–1304, 2011.

[62] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *National Academy of Sciences*, pages 3521–3526, 2017.

[63] Toru Kobayashi, Tetsushi Ikeda, Yumiko O Kato, Akira Utsumi, Isamu Nagasawa, and Satoshi Iwaki. Evaluation of mental stress in automated following driving. In *2018 3rd International Conference on Robotics and Automation Engineering (ICRAE)*, pages 131–135. IEEE, 2018.

[64] LLC Kodiak. Kodiak safety report 2020. *Kodiak Safety Report*, pages 1–49, 2020.

[65] Jan Koutník, Giuseppe Cuccu, Jürgen Schmidhuber, and Faustino Gomez. Evolving large-scale neural networks for vision-based reinforcement learning. In *15th Annual Conf. on Genetic and Evolutionary Computation (GECCO)*, pages 1061–1068. ACM, 2013.

[66] Daniel Krajzewicz, Jakob Erdmann, Michael Behrisch, and Laura Bieker. Recent development and applications of sumo-simulation of urban mobility. *International journal on advances in systems and measurements*, 5(3&4), 2012.

[67] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven L Waslander. Joint 3d proposal generation and object detection from view aggregation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–8. IEEE, 2018.

[68] Markus Kuderer, Shilpa Gulati, and Wolfram Burgard. Learning driving styles for autonomous vehicles from demonstration. In *2015 IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 2641–2646, May 2015.

[69] Fabien Lagriffoul, Neil T Dantam, Caelan Garrett, Aliakbar Akbari, Siddharth Srivastava, and Lydia E Kavraki. Platform-independent benchmarks for task and motion planning. *IEEE Robotics and Automation Letters*, 3(4):3765–3772, 2018.

[70] Morteza Lahijanian, Marius Kloetzer, Sara Itani, Calin Belta, and Sean B Andersson. Automatic deployment of autonomous cars in a robotic urban-like environment (RULE). In *2009 IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 2055–2060, May 2009.

[71] Steven M. Lavalle. Rapidly-exploring random trees: A new tool for path planning. Technical report, 1998.

[72] Michael Glazer William Angell Spencer Dodd Benedikt Jenik Jack Terwilliger Aleksandr Patsekin Julia Kindelsberger Li Ding Sean Seaman Alea Mehler Andrew Sipperley Anthony Pettinato Bobbie Seppelt Linda Angell Bruce Mehler Bryan Reimer Lex Fridman, Daniel E. Brown. Mit advanced vehicle technology study: Large-scale naturalistic driving study of driver behavior and interaction with automation. *IEEE Access*, 7:102021–102038, 2019.

[73] Richard Liaw, Sanjay Krishnan, Animesh Garg, Daniel Crankshaw, Joseph E. Gonzalez, and Ken Goldberg. Composing meta-policies for autonomous driving using hierarchical deep reinforcement learning. http://arxiv.org/abs/1711.01503, 2017.

[74] Maxim Likhachev and Dave Ferguson. Planning long dynamically feasible maneuvers for autonomous vehicles. *The International Journal of Robotics Research*, 28(8):933–945, 2009.

[75] Zachary C Lipton, Kamyar Azizzadenesheli, Abhishek Kumar, Lihong Li, Jianfeng Gao, and Li Deng. Combating reinforcement learning's Sisyphean curse with intrinsic fear. http://arxiv.org/abs/1611.01211, 2016.

[76] Michael L Littman, Ufuk Topcu, Jie Fu, Charles Isbell, Min Wen, and James MacGlashan. Environment-independent task specifications via GLTL. http://arxiv.org/abs/1704.04341, 2017.

[77] Changchun Liu, Pramila Agrawal, Nilanjan Sarkar, and Shuo Chen. Dynamic difficulty adjustment in computer games through real-time anxiety-based affective feedback. *International Journal of Human-Computer Interaction*, 25(6):506–529, 2009.

[78] Dahai Liu, Nikolas D Macchiarella, and Dennis A Vincenzi. Simulation fidelity. *Human factors in simulation and training*, pages 61–73, 2008.

[79] Wei Liu, Seong-Woo Kim, Scott Pendleton, and Marcelo H Ang. Situation-aware decision making for autonomous driving on urban road using online POMDP. In *Intelligent Vehicles Symp. (IV)*, pages 1126–1133, 2015.

[80] Yiting Liu and Umit Ozguner. Human driver model and driver decision making for intersection driving. In *2007 IEEE Intelligent Vehicles Symp. (IV)*, pages 642–647, June 2007.

[81] SM Sohel Mahmud, Luis Ferreira, Md Shamsul Hoque, and Ahmad Tavassoli. Application of proximal surrogate indicators for safety evaluation: A review of recent developments and research needs. *IATSS research*, 41(4):153–163, 2017.

[82] Regan L Mandryk and Lennart E Nacke. Biometrics in gaming and entertainment technologies. In *Biometrics in a Data Driven World*, pages 215–248. Chapman and Hall/CRC, 2016.

[83] Reza Mashayekhi, Mohd Yamani Idna Idris, Mohammad Hossein Anisi, and Ismail Ahmedy. Hybrid rrt: A semi-dual-tree rrt-based motion planner. *IEEE Access*, 8:18658–18668, 2020.

[84] Markus Maurer and ED Dickmanns. A system architecture for autonomous visual road vehicle guidance. In *Proceedings of Conference on Intelligent Transportation Systems*, pages 578–583. IEEE, 1997.

[85] Matthew McNaughton, Chris Urmson, John M Dolan, and Jin-Woo Lee. Motion planning for autonomous driving with a conformal spatiotemporal lattice. In *2011 IEEE International Conference on Robotics and Automation*, pages 4889–4895. IEEE, 2011.

[86] Jonathan Meijer, Qujiang Lei, and Martijn Wisse. Performance study of single-query motion planning for grasp execution using various manipulators. In *2017 18th International Conference on Advanced Robotics (ICAR)*, pages 450–457. IEEE, 2017.

[87] Jeff Michels, Ashutosh Saxena, and Andrew Y Ng. High speed obstacle avoidance using monocular vision and reinforcement learning. In *22nd Int. Conf. on Machine Learning*, pages 593–600. ACM, 2005.

[88] Mark Moll, Ioan A Sucan, and Lydia E Kavraki. Benchmarking motion planning algorithms: An extensible infrastructure for analysis and visualization. *IEEE Robotics & Automation Magazine*, 22(3):96–102, 2015.

[89] Kristin Mühl, Christoph Strauch, Christoph Grabmaier, Susanne Reithinger, Anke Huckauf, and Martin Baumann. Get ready for being chauffeured: Passenger's preferences and trust while being driven by human and automation. *Human factors*, page 0018720819872893, 2019.

[90] Urs Muller, Jan Ben, Eric Cosatto, Beat Flepp, and Yann L Cun. Off-road obstacle avoidance through end-to-end learning. In *Advances in neural information processing systems*, pages 739–746, 2006.

[91] Hiroki Murakami and Hideki Ohira. Influence of attention manipulation on emotion and autonomic responses. *Perceptual and Motor Skills*, 105(1):299–308, 2007.

[92] Wassim G Najm, John D Smith, Mikio Yanagisawa, et al. Pre-crash scenario typology for crash avoidance research. Technical report, United States. National Highway Traffic Safety Administration, 2007.

[93] Kristoffer Öfjäll and Michael Felsberg. Biologically inspired online learning of visual autonomous driving. In *British Machine Vision Conf. 2014, Nottingham, UK September 1-5 2014*, pages 137–156. BMVA Press, 2014.

[94] Kristoffer Öfjäll, Michael Felsberg, and Andreas Robinson. Visual autonomous road following by symbiotic online learning. In *2016 IEEE Intelligent Vehicles Symp. (IV)*, pages 136–143, June 2016.

[95] Luis Oliveira, Karl Proctor, Christopher G Burns, and Stewart Birrell. Driving style: How should an automated vehicle behave? *Information*, 10(6):219, 2019.

[96] Brian Paden, Michal Cap, Sze Zheng Yong, Dmitry Yershov, and Emilio Frazzoli. A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Transactions on Intelligent Vehicles*, 1(1):33–55, 2016.

[97] Jia Pan and Dinesh Manocha. Fast probabilistic collision checking for sampling-based motion planning using locality-sensitive hashing. *The International Journal of Robotics Research*, 35(12):1477–1496, 2016.

[98] Chris Paxton, Vasumathi Raman, Gregory D. Hager, and Marin Kobilarov. Combining neural networks and tree search for task and motion planning in challenging environments. In *2017 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 6059–6066, September 2017.

[99] Christian Pek, Vitaliy Rusinov, Stefanie Manzinger, Murat Can Uste, and Matthias Althoff. Commonroad drivability checker: Simplifying the development and validation of motion planning algorithms. 2020.

[100] Imotions: Biometric Research Platform. Eye tracking: The complete pocket guide. https://imotions.com/blog/eye-tracking, 2018. Accessed: 2019-07-17.

[101] Kathrin Pollmann, Oilver Stefani, Amelie Bengsch, Matthias Peissner, and Mathias Vukelić. How to work in the car of the future?: A neuroergonomical study assessing concentration, performance and workload based on subjective, behavioral and neurophysiological insights. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, pages 54:1–54:14, New York, NY, USA, 2019. ACM.

[102] Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. In *Advances in neural information processing systems*, pages 305–313, 1989.

[103] Rodrigo Queiroz, Thorsten Berger, and Krzysztof Czarnecki. Geoscenario: An open dsl for autonomous driving scenario representation. In *IEEE Intelligent Vehicles Symposium (IV)*, Paris, 2019. IEEE, IEEE.

[104] Pramila Rani, Nilanjan Sarkar, Craig A Smith, and Leslie D Kirby. Anxiety detecting robotic system–towards implicit human-robot collaboration. *Robotica*, 22(1):85–95, 2004.

[105] James Reeds and Lawrence Shepp. Optimal paths for a car that goes both forwards and backwards. *Pacific journal of mathematics*, 145(2):367–393, 1990.

[106] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *14th Int. Conf. on artificial intelligence and statistics*, pages 627–635, 2011.

[107] Patrick Rossner and Angelika C Bullinger. How do you want to be driven? investigation of different highly-automated driving styles on a highway scenario. In *International Conference on Applied Human Factors and Ergonomics*, pages 36–43. Springer, 2019.

[108] Dorsa Sadigh, Eric S Kim, Samuel Coogan, S Shankar Sastry, and Sanjit A Seshia. A learning based approach to control synthesis of Markov decision processes for linear temporal logic specifications. In *Decision and Control (CDC), 2014 IEEE 53rd Annual Conf. on*, pages 1091–1096. IEEE, 2014.

[109] Atrisha Sarkar and Krzysztof Czarnecki. A behavior driven approach for sampling rare event situations for autonomous vehicles. *arXiv preprint arXiv:1903.01539*, 2019.

[110] Chris Schwarz. On computing time-to-collision for automation scenarios. *Transportation research part F: traffic psychology and behaviour*, 27:283–294, 2014.

[111] Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. Safe, multi-agent, reinforcement learning for autonomous driving. http://arxiv.org/abs/1610.03295, 2016.

[112] Anthony Stentz. Optimal and efficient path planning for partially known environments. In *Intelligent unmanned ground vehicles*, pages 203–220. Springer, 1997.

[113] Ioan A Sucan, Mark Moll, and Lydia E Kavraki. The open motion planning library. *IEEE Robotics & Automation Magazine*, 19(4):72–82, 2012.

[114] Liting Sun, Cheng Peng, Wei Zhan, and Masayoshi Tomizuka. A fast integrated planning and control framework for autonomous driving via imitation learning. In *ASME 2018 Dynamic Systems and Control Conf.* American Society of Mechanical Engineers, 2018.

[115] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2446–2454, 2020.

[116] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[117] Meirav Taieb-Maimon and David Shinar. Minimum and comfortable driving headways: Reality versus perception. *Human factors*, 43(1):159–172, 2001.

[118] Eric Thorn, Shawn C Kimmel, Michelle Chaka, Booz Allen Hamilton, et al. A framework for automated driving system testable cases and scenarios. Technical report, United States. Department of Transportation. National Highway Traffic Safety , 2018.

[119] Sebastian Thrun, Mike Montemerlo, Hendrik Dahlkamp, David Stavens, Andrei Aron, James Diebel, Philip Fong, John Gale, Morgan Halpenny, Gabriel Hoffmann, et al. Stanley: The robot that won the darpa grand challenge. *Journal of field Robotics*, 23(9):661–692, 2006.

[120] Christopher Urmson, Joshua Anhalt, Hong Bae, J. Andrew (Drew) Bagnell, Christopher R. Baker, Robert E. Bittner, Thomas Brown, M. N. Clark, Michael Darms, Daniel Demitrish, John M. Dolan, David Duggins, David Ferguson, Tugrul Galatali, Christopher M. Geyer, Michele Gittleman, Sam Harbaugh, Martial Hebert, Thomas Howard, Sascha Kolski, Maxim Likhachev, Bakhtiar Litkouhi, Alonzo Kelly, Matthew McNaughton, Nick Miller, Jim Nickolaou, Kevin Peterson, Brian Pilnick, Raj Rajkumar, Paul Rybski, Varsha Sadekar, Bryan Salesky, Young-Woo Seo, Sanjiv Singh, Jarrod M. Snider, Joshua C. Struble, Anthony (Tony) Stentz, Michael Taylor, William (Red) L. Whittaker, Ziv Wolkowicki, Wende Zhang, and Jason Ziglar. Autonomous driving in urban environments: Boss and the Urban Challenge. *Journal of Field Robotics Special Issue on the 2007 DARPA Urban Challenge, Part I*, 25(8):425–466, June 2008.

[121] Van Gennip, Matthew. Vehicle dynamic modelling and parameter identification for an autonomous vehicle, 2018.

[122] Jiankun Wang, Wenzheng Chi, Chenming Li, Chaoqun Wang, and Max Q-H Meng. Neural rrt*: Learning-based optimal path planning. *IEEE Transactions on Automation Science and Engineering*, pages 1–11, 2020.

[123] LLC Waymo. On the road to fully self-driving. *Waymo Safety Report*, pages 1–43, 2017.

[124] Junqing Wei, Jarrod M Snider, Tianyu Gu, John M Dolan, and Bakhtiar Litkouhi. A behavioral planning framework for autonomous driving. In *IEEE Intelligent Vehicles Symp. Proc.*, pages 458–464. IEEE, 2014.

[125] Mark R Wilson, Greg Wood, and Samuel J Vine. Anxiety, attentional control, and performance impairment in penalty kicks. *Journal of Sport and Exercise Psychology*, 31(6):761–775, 2009.

[126] Peter Wolf, Christian Hubschneider, Michael Weber, André Bauer, Jonathan Härtl, Fabian Dürr, and J. Marius Zöllner. Learning how to drive in a real world simulation with deep Q-networks. In *2017 IEEE Intelligent Vehicles Symp. (IV)*, pages 244–250, June 2017.

[127] Matthew Wood, Philipp Robbel, D Wittmann, et al. Safety first for automated driving, 2019. *URL: https://www. daimler. com/documents/innovation/other/safety-first-for-automat e d-driving. pdf*, 2020.

[128] Kyle Hollins Wray, Stefan J Witwicki, and Shlomo Zilberstein. Online decision-making for scalable autonomous systems. In *26th International Joint Conference of Artificial Intelligence (IJCAI)*, pages 4768–4774, 2017.

[129] Fisher Yu, Wenqi Xian, Yingying Chen, Fangchen Liu, Mike Liao, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving video database with scalable annotation tooling. *arXiv preprint arXiv:1805.04687*, 2(5):6, 2018.

[130] Lingli Yu, Xuanya Shao, and Xiaoxin Yan. Autonomous overtaking decision making of driverless bus based on deep Q-learning method. In *2017 IEEE Int. Conf. on Robotics and Biomimetics (ROBIO)*, pages 2267–2272. IEEE, 2017.

[131] Nidzamuddin Md Yusof, Juffrizal Karjanto, Jacques Terken, Frank Delbressine, Muhammad Zahir Hassan, and Matthias Rauterberg. The exploration of autonomous

vehicle driving styles: preferred longitudinal, lateral, and vertical accelerations. In *Proceedings of the 8th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, pages 245–252. ACM, 2016.

[132] Chaoyong Zhang, Duanfeng Chu, Shidong Liu, Zejian Deng, Chaozhong Wu, and Xiaocong Su. Trajectory planning and tracking for autonomous vehicle based on state lattice and model predictive control. *IEEE Intelligent Transportation systems magazine*, 11(2):29–40, 2019.

[133] Yu Zhang, Huiyan Chen, Steven L Waslander, Jianwei Gong, Guangming Xiong, Tian Yang, and Kai Liu. Hybrid trajectory planning for autonomous driving in highly constrained environments. *IEEE Access*, 6:32800–32819, 2018.

[134] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. Maximum entropy inverse reinforcement learning. In *AAAI*, volume 8, pages 1433–1438. Chicago, IL, USA, 2008.

[135] Julius Ziegler and Christoph Stiller. Spatiotemporal state lattices for fast trajectory planning in dynamic on-road driving scenarios. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1879–1884. IEEE, 2009.