Empirical Game Theoretic Models for Autonomous Driving: Methods and Applications

by

Atrisha Sarkar

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Computer Science

Waterloo, Ontario, Canada, 2022

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner: James R. Wright
Assistant Professor, Dept. of Computing Science
University of Alberta

Supervisor(s): Krzysztof Czarnecki
Professor, Dept. of Electrical and Computer Engineering
University of Waterloo

Internal Members: Kate Larson
Professor
David R Cheriton School of Computer Science
University of Waterloo

Joanne Atlee
Professor
David R Cheriton School of Computer Science
University of Waterloo

Internal-External Member: Derek Rayside
Associate Professor
Dept. of Electrical and Computer Engineering
University of Waterloo

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the 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.

Statement of Contributions

Atrisha Sarkar was the sole author for Chapters 1, 2, 3, 4, 5, 6, and 8, which were written under the supervision of Prof. Krzysztof Czarnecki. Research contributing to Chapters 5 and 8 was performed in collaboration with Prof. Kate Larson. This thesis consists in part of six manuscripts written for publication. The research contained in Chapter 4 has not yet been published.

- Sarkar, Atrisha, Kate Larson, and Krzysztof Czarnecki. "Generalized dynamic cognitive hierarchy models for strategic driving behavior." *2022 AAAI Conference on Artificial Intelligence.* (AAAI) 2022.
- Kahn, Maximilian, Sarkar, Atrisha, and Krzysztof Czarnecki. "I Know You Can't See Me: Dynamic Occlusion-Aware Safety Validation of Strategic Planners for Autonomous Vehicles Using Hypergames." *2022 IEEE International Conference on Robotics and Automation* (ICRA). IEEE, 2022.
- Sarkar, Atrisha, and Krzysztof Czarnecki. "Solution Concepts in Hierarchical Games Under Bounded Rationality With Applications to Autonomous Driving." *Proceedings of the AAAI Conference on Artificial Intelligence.* Vol. 35. No. 6. (AAAI) 2021.
- Sarkar, Atrisha, Kate Larson, and Krzysztof Czarnecki. "A taxonomy of strategic human interactions in traffic conflicts." *NeurIPS 2021 Cooperative AI Workshop*.
- Sarkar, Atrisha, and Krzysztof Czarnecki. "A behavior driven approach for sampling rare event situations for autonomous vehicles." *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems* (IROS). IEEE, 2019.
- Sarkar, Atrisha, et al., "Trajectory prediction of traffic agents at urban intersections through learned interactions." *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2017.

The exception to sole authorship of material is Chapter 7, which was co-authored with Maximilian Kahn under the supervision of Prof. Krzysztof Czarnecki. Atrisha Sarkar was responsible for the development of the research idea, authorship of the introduction, background, dynamic occlusions, and hypergame construction sections. Maximilian Kahn was responsible for dynamic occlusion identification and vehicle injection section, experiments and evaluation, and related work sections of the chapter. The Waterloo Multi Agent Traffic Dataset, which is one of the contributions of this thesis was a joint work with other members of WISE Lab. Atrisha Sarkar was responsible for the data collection of the intersection dataset, relevant agent assignment, schema creation and processing the trajectory data in the structured format of the schema.

Abstract

In recent years, there has been enormous public interest in autonomous vehicles (AV), with more than 80 billion dollars invested in self-driving car technology. However, for the foreseeable future, self-driving cars will interact with human driven vehicles and other human road users, such as pedestrians and cyclists. Therefore, in order to ensure safe operation of AVs, there is need for computational models of humans traffic behaviour that can be used for testing and verification of autonomous vehicles. Game theoretic models of human driving behaviour is a promising computational tool that can be used in many phases of AV development. However, traditional game theoretic models are typically built around the idea of rationality, i.e., selection of the most optimal action based on individual preferences. In reality, not only is it hard to infer diverse human preferences from observational data, but real-world traffic shows that humans rarely choose the most optimal action that a computational model suggests. Through the lens of behavioural game theory, my thesis bridges the gap between observational naturalistic behaviour and game theory to create models of traffic behaviour that can have versatile applications in AV development, including testing, verification, and motion planning.

The first part of the thesis makes a set of methodological contributions towards creating models of traffic behaviour from naturalistic datasets using behavioural game theory. In the second part, the thesis demonstrates practical uses of models for safety validation of autonomous vehicle planners. At each step, the models are built around the behaviour of *boundedly rational* agents and demonstrate multiple ways of modelling suboptimal behaviour, with suboptimality being considered from the perspective of a game designer.

Although there has been an increasing interest in the use of game theoretic models for AV, it is not clear which solution concepts align well with naturalistic driving behaviour. Based on the structure of a hierarchical game, the thesis first presents various design choices that can be used in the construction of a game, along with the solution concepts from behavioural game theory that can be applied to solve such games. These choices result in thirty behaviour models, which are evaluated based on their model fit and predictive accuracy on naturalistic data. The results provide practical guidance for practitioners for the construction of traffic behaviour models.

Driving is a multi-objective task, and humans aggregate objectives of safety and progress in a context and individual specific manner. It is challenging to infer the parameters of multiobjective utility aggregation solely from observations because of a number of unobserved variables. Based on the concept of *rationalisability*, the thesis develops algorithms for estimating multiobjective aggregation parameters for two aggregation methods, weighted and satisficing aggregation, and also when the underlying model of reasoning consists of both strategic and nonstrategic reasoners. Experiments conducted in three different datasets provide interesting insights into how road users aggregate these objectives, as well as the situational dependence of the aggregation process.

In the final methodological contribution, the thesis addresses two key challenges of building traffic behaviour models using dynamic games; model instability and model uncertainty. Model instability arises when a class of boundedly rational agents who follow elementary nonstrategic models of behaviour have no reason to adhere to elementary models over time in the game. The thesis addresses this problem by developing a nonstrategic yet sophisticated finite-state transducer-based model of level-0 behaviour within the level-k framework. Model uncertainty arises when agents are free to follow any model of reasoning as is often the case in naturalistic data. This problem is addressed by developing a generalised cognitive hierarchy model consisting of three layers, nonstrategic, strategic, and robust. Each layer can hold multiple behaviour models, and the chapter develops solutions for heterogeneous models based on the consistency of beliefs over observations. Simulation experiments demonstrate that a robust layer model is an appropriate choice for an AV behaviour planner.

Building on the game theoretic models, the second part of the thesis demonstrates the application of the models by developing novel safety validation methodologies for testing AV planners. The first application is an automated generation of interpretable variations of lane change behaviour based on Quantal Best Response model. The proposed model is shown to be effective for generating both rare-event situations and to replicate the typical behaviour distribution observed in naturalistic data. The second application is safety validation of strategic planners in situations of dynamic occlusion. Using the concept of hypergames, in which different agents have different views of the game, the thesis develops a new safety surrogate metric, dynamic occlusion risk (DOR), that can be used to evaluate the risk associated with each action in situations of dynamic occlusion. The thesis concludes with a taxonomy of strategic interactions that maps complex design specific strategies in a game to a simpler taxonomy of traffic interactions. Regulations around what strategies an AV should execute in traffic can be developed over the simpler taxonomy, and a process of automated mapping can protect the proprietary design decisions of an AV manufacturer.

Acknowledgements

Working on this dissertation has been one of the most challenging and rewarding experiences of my life, and it would not have been possible without the love and support of several people, including my supervisor, mentors, family and friends.

First and foremost, I would like to thank my supervisor Prof. Krzysztof Czarnecki. He is one of the most insightful people I have met, and at every step of the seven years of my grad school, I am deeply grateful for his supervision and support. I am thankful for the encouraging conversations we had after my masters that made me believe in myself and helped me realise that doing a Ph.D. is a right choice for me. This thesis would not have been possible without him showing confidence and allowing me to work on a topic that was relatively new to me. There were many personal crises during the period I was working on my Ph.D., and having an incredibly supportive supervisor in Krzysztof was the foremost reason I did not consider quitting as an option.

I would like to thank the members of my examining committee, Prof. James R Wright, Prof. Derek Rayside, Prof. Jo Atlee, and Prof. Kate Larson for providing very valuable feedback. I am also thankful to Prof. Kate Larson for the generous guidance and help she provided while I was working on some of the tough chapter of the thesis. Her deep expertise in multi-agent systems and game theory helped me frame research questions in the right way, and her feedback significantly improved the quality of the papers. I am also grateful for other professors in the computer science department who supported me in many ways and created a welcoming environment. In particular, I would like to thank Prof. Robin Cohen, Prof. Maura Grossman, and Prof. Jo Atlee in that regard.

My colleagues in the WISE lab (Nicole, Rodrigo, Prarthana, Samin, Matt, Changjian, Max, Frederic, Jinwei and several others) and the erstwhile GSD lab (Pavel, Edward, Leo, Jimmy, Thorsten and several others) made grad school life a wholesome and enjoyable experience. I am thankful for the mentorship of Prof. Sean Sedwards and Dr. Michal Antkiewicz, on whom I could rely for help and advice whenever I needed one. I also would like to mention Prof. Jianmei Guo, who taught me early on the ropes of how to conduct research and write a paper, and I have been following those advice ever since.

A lot of things had to go right for me growing up in rural West Bengal to be able to go to grad school in computer science at the University of Waterloo. I was lucky to have my dad work for a company that provided their employees housing, healthcare, and, in my case, my dad's successful bargain for a three-hour transportation to a different city for me to be able to go to a good school. I don't think I would have ever considered the life I have today possible without the opportunities that came as a result of that.

Closer to me in time and space here in Waterloo, I am thankful to the community of LGBTQ+ people in the region that SPECTRUM was able to help connect. I met so many wonderful people there, more than I can ever list here, who made me feel like I was not alone and many of whom became my close friends. I am thankful to Wash for helping me take care of my mental health and the support they provided in critical moments of my university life. I would also mention my friends; Mayukh, with whom I discovered the allure of academia as a career choice in high school and has been a friend through so many ups and downs for many decades; Piyusha, with whom I started on the grad school journey together; and Oliver, who was my first friend in a new country and culture and whom I still consider as a close friend.

The person who deserves the most credit for me being able to complete this thesis is my best friend and partner, Valerie. I am so lucky to have such an amazing person by my side with whom I could share not only a life, but also graduate school and research interests together. Beyond all the love and support she gave me, I am deeply grateful for her expertise and for entertaining my annoying dinner conversations that started with *can I ask you a question about game theory?*

Above all, none of this would have been possible without the love and sacrifices of my mom, Purabi, and my dad, Ashim. So much of their own lives revolved around making sure I got a good life and I will never forget how much they always put my priorities above their own. While I was finishing my thesis, even through her illness, my mom was sometimes more worried about the impact of my additional caregiving responsibilities than her own health. I am also grateful to have a loving family (*dida, mashi, mesho, mama, mami, Riju, Shraeya, manu kutu*, and the memories of *dadu*), and my happiest moments were those when we all got together.

This work was supported in part by the Ontario Graduate Scholarships and Queen Elizabeth II Graduate Scholarship in Science and Technology (QEII-GSST) Program.

Dedication

This is dedicated to my mom who, true to her nature, made sure that I do not forget to dedicate this to my dad and grandma too.

Table of Contents

List of Figures

List of Tables

Me thinking about what others' are thinking about me? That's anxiety, Trish. Not "game theory"!

Owen.

Chapter 1

Introduction

On average, Canadians spend at least one hour every day as occupants or driving a passenger vehicle[1]. This may not seem much, but driving is relatively a new human activity that has established itself as a major drain of our daily time only in the past century [167]. A technological utopia that will help us recover that hour of our time through safe and fast automated transportation has been part of our imagination for at least six decades [214]. Today's futuristic landscape of an urban city is incomplete without the image of a self-driving car transporting individuals for whom the stress of driving is a thing of the past. However, this spectacular dream stays grounded in the modality of the transportation it envisions, that is, the transportation of individual vehicles over roadways. Over the past decade we have been inching closer to this promised utopia, with more than 80 billion US dollars invested towards the development of autonomous vehicles[2]. The technology behind autonomous vehicles has been synonymous with the advent of the AI revolution, and computer scientists have the responsibility to oversee the safe development of that technology.

Autonomous vehicles have also captured the imagination of a wide range of research disciplines, thus heralding a new avenue for multidisciplinary research. AV technologies present cross-cutting concerns that span several disciplines, including engineering, computer science, law [17], economics [244, 125], social sciences [206], and philosophy [168, 157]. At the core of this technology, however, are the human road users; people who (will) interact with the vehicles on the roads. Unless all modes of transportation on the road are automated, for the foreseeable future, humans will continue to interact with these automated vehicles as pedestrians, bicyclists, and occupants or drivers of other vehicles. For the most part, vehicles that interact with AV on

[1] https : / / www150.statcan.gc.ca / t1 / tbl1 / en / tv.action?pid = 4510001401

[2] https://www.brookings.edu/research/gauging-investment-in-self-driving-cars/

roads today are driven by humans, and driving as we understand today is still a human activity based on rules and patterns of behaviour, both explicit and implicit. If an automated system enters this ecosystem, it is clear that there has to be some level of *understanding* of that behaviour for AVs to interact in a safe and sustainable manner. This thesis is an effort in that direction, in which I focus on modelling human driving behaviour for testing, verification, and behaviour planning of autonomous vehicles.

Automated driving system (ADS) architecture. Driving is a complex human activity that involves perception (e.g., assessing the distance and velocity of other road users), motor control (braking, acceleration, steering), attention (staying focused on the road and situational awareness), reasoning (assessing the intent and behaviour of other road users) among other mentally and physically demanding tasks [91]. The goal of an automated driving system is replacement of these complex functions, and with advances of AI and automation, self driving cars on the road today have started that journey. Advances in computer vision can detect relevant road users in the vicinity of a vehicle, LIDAR systems can asses distance and speed [10], and onboard computer systems can execute commands that control the physical movements of the vehicle [145]. Although commercial self-driving cars may use their own proprietary design and architecture to engineer an automated driving system, an example architecture of a typical ADS is shown in Fig. 1.1. The main components of the architecture include i) the vehicle platform, which consists of sensors and actuators, ii) the perception stack (object detection, tracking, and localisation), iii) the environment model, which is responsible for integrating information about lanes, maps, and other static environmental aspects together for the perception and planning modules, and iv) the planning module, which is responsible for planning the physical motion of the vehicle by synchronising three hierarchical tasks: mission, behaviour, and trajectory planning [98]. The mission planner's responsibility is to plan the route between the origin and destination of the vehicle. This action plan is broken down into sequence of manoeuvres by the behaviour planner, which further sets up the problem of trajectory generation for the trajectory planner. The trajectory planner module uses various search and optimisation based methods to generate a set of control actions to be executed by the vehicle controller [173]. Within the context of behaviour planning, traffic behaviour models of other road users play an important role in both the operation of an AV as well as its development[3]. Traffic behaviour models are the focus of this thesis, and I address challenges associated with computational modelling of human interactive behaviour in traffic for autonomous vehicles.

[3] *Development* here refers to the software development life cycle of the ADS

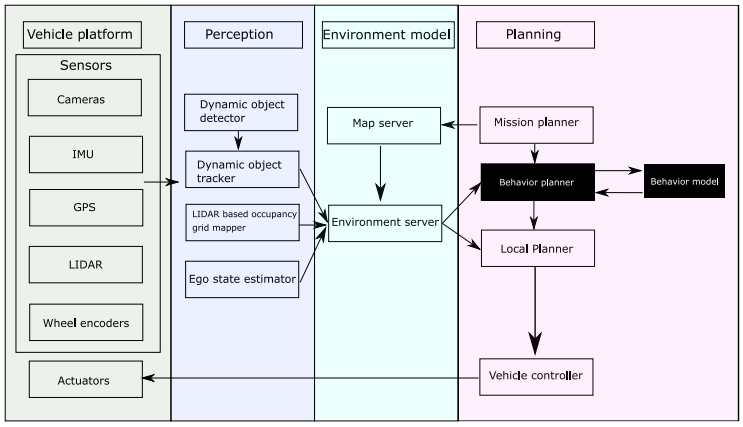


Figure 1.1: A typical high-level architecture for an automated driving system in an AV. Arrows depict flow of information between modules.

1.1 Motivation

In this section, I discuss the motivation behind traffic behaviour models for AV. First, I present the different scopes of use of traffic behaviour models. Next, I highlight the versatile applications of the models in AV development.

1.1.1 Traffic behaviour modelling for autonomous driving.

A traffic behaviour model is a general term that refers to a model of the behaviour of road users, such as vehicles, pedestrians, and cyclists, in the vicinity of the autonomous vehicle. Traffic behaviour models can have different functional scopes in which they can be used in an application.

- *Predictive scope.* When used in a predictive scope, a model predicts future actions (such as high-level manoeuvres or trajectories) of other vehicles and road users in the vicinity of a subject AV. Predictive scope is relevant when an AV has to plan its own actions taking into account the predicted actions of other road users. Additionally, performance in a predictive scope measured through predictive accuracy is also a key metric of evaluation to assess the quality of behaviour models regardless of whether those models are specifically used in the motion planning algorithm in an ADS.

- *Descriptive scope.* Descriptive scope deals with the aggregate behaviour of agents in a population. A typical example of the use of a behaviour model in a descriptive scope is the probability distribution of gap acceptance rates in a population of drivers in a city during

lane changes. To facilitate quantitative analysis from a descriptive scope, models are often made parametric and then fitted to observational data. Subsequently, the parameters of the model can answer specific questions of interest, e.g., what percentage of drivers engage in risky lane change behaviours.

- *Generative scope.* While descriptive model can explain certain characteristics of a population based on observed data, in a generative scope, a model has the ability to generate new situations and behaviours that were not observed in the data. This is done in simulation by changing the parameters in the descriptive scope of the model in a principled way. For example, changing the distribution of gap acceptance rates to sample from a more aggressive population of drivers to evaluate whether an AV can behave safely under those conditions. In the context of AV development, being able to use a model in a generative scope helps to sample novel and edge-case situations that are essential for ensuring safety.

1.1.2 Applications of traffic behaviour models for ADS.

Based on the above functional scopes, traffic behaviour models can have versatile applications in AV development. When used in a predictive scope, traffic behaviour models can be used to plan AV behaviour, whereas when used in a descriptive and generative scope, the models can be used to test, verify, and train behaviour planners in the development lifecycle. Next, I briefly discuss these applications.

- *Planning.* Behaviour planning module in an ADS is responsible for planning vehicle manoeuvres, often interpreted as the *behaviour* of the vehicle as understood in a colloquial sense. This involves decisions such as slowing down, executing a turn, changing lanes, yielding, etc. In order to choose the right behaviour, the AV needs to predict the actions of other road users and respond accordingly. When used in a predictive scope, the role of a traffic behaviour model is the generation of predicted actions of other road users for a given traffic situation.

Descriptive and generative scopes are often applied in conjunction with each other. For the following applications, traffic behaviour models can be used in both descriptive and generative scopes.

- *Training.* In recent years, reinforcement learning based methods have gained prominence as a scalable method of training an ADS to learn optimal behaviour [112]. The learning of the optimal policy during the training process occurs through interaction with other

road users in the simulated training environment. Traffic behaviour models can provide a model of the behaviour of other road users in the training environment that is reflective of real-world traffic behaviour.

- *Testing.* In order to ensure that an AV is safe for operation, simulation plays a critical role, as it is not possible to rely primarily on extensive field tests as a primary testing method [107]. Test scenarios for simulation need to balance the generation of scenarios that are representative of typical behaviour and edge case scenarios. While the former can be derived from naturalistic datasets, the latter are not often observed in datasets because of their rarity. Since the behaviour of road users can be parameterized in traffic behaviour models, the models can augment naturalistic data with rare and edge case scenarios for testing AV motion planners.

- *Verification.* The goal of formal verification techniques applied to autonomous driving systems is to ensure the functional safety of an automated vehicle [180]. Safety properties, often derived from traffic rules, are expressed in a formal specification language such as Linear Temporal Logic (LTL) [183] and Signal Temporal Logic (STL) [219]. Similarly to the environment model in reinforcement learning, model checking techniques that verify safety properties also make use of a model of behaviour against which the motion planner's safety properties are verified. Along with the problem of scale [123], having a realistic model of interactive human behaviour is a bottleneck in formal verification [28]. A computational traffic behaviour model can address the latter problem by acting as a model of the environment, i.e., the behaviour of other road users, thereby advancing the feasibility of the use of formal methods in the development of AV.

1.2 Properties and current landscape

With respect to the specific modelling paradigms, traffic behaviour models in the context of AV can take different forms [32]. In this thesis, I focus on game theoretic models of behaviour. This choice is based on a set of properties of interactive reasoning that enables the modelling framework to be used in a variety of applications, as discussed in the previous section. I present those properties next, followed by a brief overview of the current literature on traffic behaviour models.

- *Conditional reasoning*. This includes the ability of a model to infer the probability of possible outcomes as a consequence of a specific action taken by a road user. This is based on the conditional probability of the form $P(Y|X)$ where Y is a random variable encoding

the outcome (e.g., a collision) and X is a random variable in the domain of possible actions. This property helps answer questions of the form — *if the self driving car executes a merge manoeuvre, what is the probability of a collision with a vehicle in the target lane.*

- *Normative reasoning*. Behaviour models should ideally support a theory of what agents *ought* to do. The importance of normative reasoning for behaviour models is in relation to traffic rules; which provide a prescriptive notion of how drivers and other road users should act in different traffic circumstances. Normative reasoning may also include support for other theories, such as rational choice theory, which says that a rational driver will choose an optimal action with respect to a set of objectives.

- *Counterfactual reasoning*. This refers to the ability of the models to engage in *what-if* type reasoning, i.e., reasoning based on possible events that are contrary to the observed ones. A practical use of counterfactual reasoning is to update beliefs about the behaviour of other road users, which can be later incorporated in planning the action of the subject vehicle. For example, if a self-driving car observes another vehicle *not* slowing down near a stop sign as expected, it can form possible hypotheses about the other vehicle, such as its aggressive disposition, being in a hurry, or being distracted.

- *Strategic reasoning.* Strategic reasoning involves reasoning that takes into account that the actions of others may be influenced by one's own action. It brings together conditional, counterfactual, and normative reasoning in a unified reasoning framework. A typical example in the context of driving involves reasoning of the following nature: a vehicle needs to turn right at an unprotected right turn. The vehicle observes a vehicle approaching the intersection, and reasons that *if I execute a turn instead of yielding, it is in the interest of the other driver, who seems to be a cautious driver, to slow down for me.*. When multiple road users apply strategic reasoning jointly to an interactive problem, along with the belief that others are also strategic reasoners, the solution results in equilibrium.

Current landscape of traffic behaviour models. Over the years, traffic behaviour models have been an area of extensive research in the field of traffic safety engineering and traffic psychology [69]. The focus of that research precedes treating the problem of behaviour modelling as one for AVs, with various objectives for building the models. Traffic accidents and collisions parallel the use of motor vehicles, and a primary objective of building computational models of driver behaviour in the field of safety engineering is to design appropriate interventions to reduce traffic accidents. At the same time, since driving is fundamentally a human activity, understanding driving from a psychological perspective has also played a key role in this endeavour [208]. Treating the problem of traffic safety as an interdisciplinary science, models such as Task Difficulty Homeostasis (TDH) [78], Risk Allostasis Theory (RAT) [76], and Risk Monitoring Model

(RMM) [220], bridge the psychological aspects of driving with behavioural observations such as steering control and speed choices [132]. Driver behaviour models have also been developed in the field of highway and transportation engineering to design effective transportation strategies that reduce congestion. Simulation plays an important role in this regard, and aggregate outcomes of traffic behaviour, such as traffic flow, are modelled with *macroscopic* traffic simulation models [126, 158]. However, since aggregate behaviour results from individual interaction among vehicles, traffic simulation frameworks also support *microscopic* simulation models that capture individual driver behaviour and their interactions. Microscopic models are well suited for applications in AV development, i.e, a model for use in planning, testing, and verification. Popular models such as the Gipps model [88], the Newell model [162], and the Intelligent Driver model [218], have been a cornerstone of microscopic models and have been widely used in simulation platforms such as SUMO [118]. Most of these models, however, were designed for the main purpose of simulating aggregate behaviour of vehicles on a highway, therefore, the support for vehicle interactions were limited to car following scenarios. Autonomous vehicles, on the other hand, have to navigate a wide range of scenarios beyond just car following, and therefore behaviour models developed for AV needs to be general enough to be applied to any traffic scenario.

In autonomous vehicle research, the problem of modelling traffic behaviour has been addressed primarily through machine learning methods [128, 32], and the main functional scope is a predictive scope for the application of motion planning. The use of machine learning models is promising, especially since in theory the models have the potential to generalise to any type of traffic situations. However, to realize that potential, there is need for availability of significant amount of training data to the researchers. In recent years, open datasets like Argoverse [44] and Waymo Open Data [68] have filled that gap. The machine learning based approaches are nevertheless limited in its ability to reason interactively with respect to some of the properties discussed earlier. Still, there has been effort to address those shortcomings within machine learning based models. The first two properties, vis-à-vis conditional reasoning and counterfactual reasoning, have been addressed through multimodal prediction models [154] and simulation of synthetic scenarios [224], respectively. Addressing the latter two, i.e., normative reasoning and strategic reasoning, purely through machine learning based methods is still an open question. In that regard, game theoretic techniques can provide a sophisticated model of interactive reasoning. Models that combine game theoretic approaches with machine learning methods bring together the best of both worlds, and have been an interesting recent development of the field [189, 85]. This thesis restricts itself mainly to game theoretic models, with support of machine learning based techniques in cases where parameters of the models are to be estimated from observational data.

Game theory provides a model of interactive decision making by rational agents modelling

everyone else as rational agents [75]. Game theoretic models are built on rational choice theory of individual decision making, in which a rational agent facing multiple choices selects an action that maximises their individual utility; and the utility is taken to be broadly representative of their ordered preference over those actions [131]. Most planning methods developed for autonomous vehicles, for example, trajectory optimisation [173], model predictive control [145], also work under that principle, i.e., a rational agent (in this case an AV) selecting an action from a set of valid actions (the trajectories or control actions) by optimising a combination of objectives such as safety, progress, comfort. Extending this idea to a multi-agent setting of decision making by a set of rational agents interacting with each other leads to a game theoretic construction. With the idea taking hold that AVs will eventually interact seamlessly with other human drivers in a variety of traffic situations, in recent years, there has been a rapid interest in using game theoretic models for planning the behaviour of an AV [215, 211, 190, 70, 179]; although the first use of game theory models as a computational model of driving goes back several decades [113].

With just a few constituents of a model that includes a set of players, their preferences over actions, and informational assumptions that encapsulate what agents know about each other, game theoretic models provide a general and parsimonious model of interaction. The models also support the set of desirable properties that was listed earlier, namely conditional, normative, counterfactual, and strategic reasoning. However, applying the general framework to the problem as complex as human driving is non-trivial. In reality, not only is it hard to infer human preferences from observational data in the face of diverse behaviour, but real-world traffic shows that humans rarely choose the most optimal action that a computational model suggests. Additional questions such as who are the players of the game and the construction of actions in the game need to be studied methodically in order to build usable models that reflect actual traffic behaviour. This thesis focusses on addressing those essential problems, methodologically and empirically.

1.3 Problem statement

The general form of a game theoretic traffic behaviour model can be expressed as the following function, $\mathcal{O} \backsim f(S, U, \mathcal{B}, \epsilon)$, where $\mathcal{O}$ is the output of the model, which are either high-level manoeuvres, trajectory control signals, or both. If the traffic behaviour model supports probabilistic output, $\mathcal{O}$ can be a random variable in the domain of manoeuvre and trajectories. S, U, $\mathcal{B}$, ϵ are the state, utilities, a model of reasoning, and a model of decision errors, respectively. Each of these aspects is discussed in detail next.

State (S) — S is a vector that represents the state of the world, including instantaneous or historical states of vehicles, for example, position, velocity, acceleration, and traffic information,

such as lanes and the state of traffic lights. Models that use additional information, for example historical information about inferred vehicle manoeuvres, can also be included in S. Any external factor that is relevant to the behaviour can be included in the state vector. Typically, models use S as an independent variable, and other aspects of the models ($U, \mathcal{B}, \epsilon$) are modelled as a dependent variable.

Preferences (U) — The second aspect, U, of the model encapsulates the preferences of road users, which are relevant in their traffic decision making. Game theoretic models work under the premise that every player (a road user) in a game chooses their action broadly based on preferences they have over the possible outcomes. To make the process of human choices amenable to computational modelling, preferences over choices are assigned a real-valued utility that is consistent with that ordering [223, 26]. Standard optimisation based techniques can then be employed to model the selection of an action. In many applications, the utilities have correspondence with a concrete human desire, for example, money in the case of economic models. In the context of driving, utilities often correspond to the desire to avoid collisions with other road users, to make progress towards the destination, to experience a smooth ride, etc. There are two distinct problems when it comes to constructing the utilities in a game theoretic setting of driving behaviour. First, unlike in a laboratory setting in which experimental game theoretic studies are typically based upon, when models are constructed from observational naturalistic data, the underlying utilities are opaque to the model designer. Therefore, the designer must first estimate the utility under which the road user chose the observed action. Estimating the underlying utility from observations is further complicated by other variables in the model, i.e., the underlying model of reasoning of the player, the errors in judgement one can make in their decision making, which all act as confounding factors. Therefore, the first research question the thesis addresses in this context is

> *How to estimate multiobjective utilities of players in a game from observational data that takes into account the underlying model of reasoning?*

The second problem arises from the multi objective nature of utilities. During traffic interactions, humans balance multiple objectives in the process of selecting their action. Game theoretic models on the other hand relies on a scalar utility, based on which a notionally optimal action is calculated. Therefore, the second research question is

> *How are performance of game theoretic models affected when different aggregation methods, such as linear weighting and satisficing, are used in the construction of the player utilities?*

Model of reasoning ($\mathcal{B}$) — In order to select an action, players in a game need to "solve" the game based on a model of reasoning $\mathcal{B}$. The model of reasoning often uses a specific solution concept, which is a formal process that generates one or more actions for each player in the game. The most popular solution concept used is the Nash equilibrium, which has been proposed as a solution to model the actions of drivers in traffic [113, 201]. Nash equilibrium has the appeal of stability, that is, if all players follow the prescribed action of an equilibrium, then no player can gain advantage by unilaterally deviating from the prescribed action. Other solution concepts that are similar in nature but with different informational assumptions (e.g., who takes the first action in the game) include the Stackelberg equilibrium, which has also been proposed as a solution for traffic decisions [70]. However, in real traffic situations, even if the utilities constructed by the designer align well with the actual player in question, humans may possibly deviate from those equilibrium solutions that a computational model suggests. In part, that deviation can be explained by suboptimal reasoning, i.e., in situations of cognitive pressure, humans might not engage in the complex deliberative reasoning required to arrive at an equilibrium solution. In order to accommodate such deviations, level-k based methods have been proposed as a computational model where the population of players in the game consists of different levels, $\{0, 1, .., k\}$, of reasoners with increasing capacity of strategic reasoning [40, 234]. In recent years, level-k models have also gained prominence as a model of traffic behaviour [215, 210]. Another way to model suboptimal reasoning involves the use of non-strategic solution concepts, such as maxmax (choosing an utility maximising action) or maxmin (best of worst case) based action selection. In short, various solution concepts have been proposed as the model of traffic behaviour; however, there has been comparatively less focus on evaluation of different models of reasoning with respect to real-world data. Since there are several possibilities of deviation from the idealised solutions, evaluating a model's efficacy against naturalistic observational data is essential. Therefore, this thesis addresses the following research questions related to the model of reasoning.

> *Which model of reasoning fares best with respect to model fit and predictive accuracy when evaluated against naturalistic data?*

> *Are there models that work best in all traffic situations? If not, then how to build game theoretic models of traffic behaviour that support multiple models of reasoning?*

Errors (ϵ) — Another explanation for deviations from optimal solutions is due to players not always selecting the optimal response. In other words, even if they have a correct understanding of other players' behaviour, the player might make errors in selection of a response action. The

deviation from optimality of this nature can be attributed to one of suboptimal *response*. In a behaviour model, these deviations can be captured by the error term ϵ. Specific theories have been proposed that model these errors in a formal way. Quantal Best Response is a commonly used theory that suggests that players make cost-proportional errors. These errors are modelled with a negative exponential distribution such that higher utility responses have an exponentially higher likelihood of selection [234]. Although it is necessary to have an error term that allows for deviations from optimality, an associated problem is that it can act as a catch all for any deviation from equilibrium; ones that the term is designed to model (i.e., suboptimal response), along with others that can arise from suboptimal reasoning or even any mismatch of other model aspects like preferences and reasoning model. This presents a problem for the good model design because any observed behaviour can be explained away through the errors. In order to address this problem, the thesis addresses the following research question.

> *How to distinguish through observation different forms of bounded rational behaviour in game theoretic models of traffic behaviour?*

Another problem related to errors is their quantification. To be able to use the models in predictive capacity, it is necessary to quantify the errors and examine whether there is an association between the state vector S and the error ϵ. The latter not only helps to quantify the error, but also gives insight into whether there are certain traffic situations (encoded in S) where the model solutions and the observed decisions do not align well. This is addressed as part of the following research question in the thesis.

> *Is there an association between the traffic state S and other model aspects, such as preferences (U), model of reasoning ($\mathcal{B}$), and errors (ϵ)?*

Beyond just developing methodologies for modelling traffic behaviour, one of the goals of the dissertation is also to demonstrate practical uses of the models in AV development. To that end, the thesis also addresses the following research question.

> *How can game theoretic models of traffic behaviour help in the safety validation of autonomous vehicle motion planners?*

1.4 Research contributions

The contributions of this thesis can be categorised into three parts: *methodological*, which includes methods to model human driving behaviour for use in AV development; *theoretical*, which

includes formal techniques that advance methodological contributions; and *empirical*, which includes understanding of human driving and road user behaviour from naturalistic observational data, along with associated datasets.

1.4.1 Methodological and theoretical contributions

- Formalisation of solution concepts of strategic and non-strategic reasoning in hierarchical games as well as estimation of precision parameter based on a generalized linear model . (Chapter 3)
- Axiomatic construction of rationalisability of observed behaviour for multi-objective utility aggregation in strategic and non-strategic behaviour models. (Chapter 4)
- An algorithm for estimating parameters of satisficing based aggregation of multi-objective utilities. (Chapter 4)
- A cognitive hierarchy model that allows for relaxed assumptions about common knowledge by allowing heterogeneous reasoning models. (Chapter 5)
- Development of a model (dlk($\mathcal{A}$)) of level-k based behaviour for dynamic games, where level-0 behaviour is modelled as automata (Chapter 5)
- Two boundedly rational models of equilibrium behaviour are based on the idea of satisficing, namely, safety-satisfied perfect equilibrium (SSPE) and manoeuvre-satisfied perfect equilibrium (MSPE). (Chapter 5)
- A behaviour driven rare event sampling methodology for generating edge case scenarios for AV testing. (Chapter 6)
- A safety validation framework for testing the performance of an AV planner in dynamic occlusion scenarios using hypergames. (Chapter 7)
- A taxonomy of strategic interactions that helps in understanding of the strategies of game theoretic behaviour planners in common language. (Chapter 8)

1.4.2 Empirical contributions

- Evaluation of thirty behaviour models for one-shot moving horizon games of hierarchical structure based on model fit and predictive accuracy on naturalistic datasets. (Chapter 3)

- Study of two modalities of multi-objective aggregation, weighted and satisficing, in human driving behaviour. (Chapter 4)
- Study of the risk and payoff dominance behaviour of road users in different traffic scenarios. (Chapter 4)
- Evaluation of behaviour models for dynamic games with respect to predictive accuracy and as a model of behaviour planning in AV. (Chapter 5)
- A novel multiagent traffic dataset[4] covering a variety of traffic scenarios, such as intersection, crosswalk, and roundabout. (Appendix B)

1.4.3 Thesis statement

Based on the above contributions, the dissertation makes the following thesis statement.

> *Game-theoretic models built from observational naturalistic driving data show deviations between the chosen human actions to the theoretical optimal solutions. Models that take into account boundedly rational behaviour can capture the deviations from optimality as established from the game designers' point of view. Such computational models can also have various applications in the domain of autonomous vehicles, such as behaviour planning, safety verification, and validation.*

1.5 Thesis overview

This section provides a summary of the chapters in the thesis. Chapters 3, 4, and 5 address the methodological challenges of game theoretic models of traffic behaviour, Chapters 6 and 7 focus on models' applications, and Chapter 8 develops a taxonomy for better understanding the decisions made by an AV planner.

Chapter 3 models driving behaviour in the context of one-shot moving horizon games. The solution concepts developed in the chapter are built upon the framework of hierarchical games, which is a game structure that addresses computational complexity through hierarchical abstraction of actions. This construction enables integration of the models with the rest of the AV motion planning architecture. The chapter highlights how different models of strategic and non-strategic

[4] Partial contribution. See statement of contributions.

reasoning as well as other modalities of bounded rationality can be used to solve a hierarchical game. Based on two metamodels, namely the level-k framework and another based on Nash equilibrium, the chapter evaluates 30 models of behaviour based on a contributed observational dataset of unprotected right turns and left turn across path.

Chapter 4 deals with the problem of estimating the utilities of players in a game when players need to aggregate multi-objective utilities. The chapter presents the case that the underlying model of reasoning acts as a confounding factor, and therefore the process of estimation needs to take into account the reasoning model. The chapter provides an axiomatic characterisation based on which observations can be *rationalized* with respect to a reasoning model. With the help of the characterisation, new algorithms to estimate aggregation parameters of multi-objective utilities are developed. Evaluation based on a diverse set of scenarios, including traffic behaviour in crosswalks, intersections, and roundabouts, provides information on how human drivers aggregate safety and progress objectives when viewed through the lens of different reasoning models.

Chapter 5 focusses on behaviour models for dynamic games. Dynamic games provides a more sophisticated space of strategies for players in the game; however, it also presents additional challenges when behavioural game theoretic concepts are applied to dynamic games, namely the problems of model instability and model uncertainty. The chapter addresses the two problem through a generalised cognitive hierarchy model consisting of non-strategic, strategic, and robust layers, where each layer can hold multiple models of behaviour. Furthermore, the chapter also demonstrates how satisficing can be used as a model of boundedly rational behaviour within the context of behaviour models. The models are evaluated based on naturalistic data as well as simulation of critical scenarios.

Chapter 6 develops a methodology for sampling rare event scenarios for the safety validation of AV motion planners. Verification of AV motion planners in rare event situations is essential for guaranteeing safety. However, generating such scenarios presents a challenge since naturalistic datasets usually do not contain enough of such observations. Methods to address this problem usually rely on sampling situations from a surrogate metrics distribution, e.g. distance or time gap. This chapter presents an alternate method based on the idea that crashes are more likely to be caused by certain combination of road user behaviours. The proposed approach is compared with state-of-the-art methods of accelerated evaluation and are shown to expedite the process of rare event generation.

Chapter 7 presents an application of game theoretic models for safety validation of AV planners under conditions of dynamic occlusions. Dynamic occlusions are transient moments in traffic where a vehicle occludes a relevant road user from the view of a subject vehicle. Dynamic occlusions are one of the leading factors in traffic accidents. This presents a challenge for game theoretic planners since the games rely on players being aware of the other players in the game.

The chapter develops a method of safety validation of such scenarios based on the framework of hypergames, which is a modelling framework that allows each player to have different views of the same interactive scenario. With the help of metric that quantifies the safety risk due to conflicting views (Dynamic Occlusion Risk metric), the developed approach can identify situations where a game theoretic planner can fail. Evaluation of the methodology shows that it can recreate real-world dynamic occlusion crash scenarios 4000% faster compared to direct sampling from naturalistic dataset.

Chapter 8 focusses on developing a common taxonomy of interactions in traffic. To make the behaviour of AVs amenable for regulation, there needs to be a common language of understanding of AV based traffic interactions. The simultaneous development of several behaviour models, game structure, solution concepts, and other ways of designing the game, makes this task challenging. Therefore, based on common patterns of interaction in traffic conflicts, the chapter develops a taxonomy for strategic interactions in everyday traffic situations. The taxonomy is developed along the dimensions of agents' initial response to right-of-way rules and subsequent response to other agents' behaviour. Furthermore, the chapter also presents a process of automatic mapping of strategies generated by a game theoretic planner to the categories in the taxonomy. Case studies include vehicle-vehicle and vehicle-pedestrian interaction simulation, and strategies generated for those interactions by QLk and Subgame perfect ϵ-Nash Equilibrium based planners are mapped to the categories in the taxonomy.

The scenarios in the dataset were selected based on traffic locations where there is a higher likelihood of strategic interactions. These scenarios, namely, intersection, roundabout, and crosswalk, are typical locations where static traffic conflict (in which the location of the conflict depends only on the lane structure) occurs between road users. Compared to dynamic conflicts (e.g., lane changes along a highway), static conflicts are well suited for capturing interactions through drone video capture, and are also more common in an urban traffic network. However, whether the behaviour of road users in scenarios of static conflict easily translated to dynamic ones is not covered in the thesis. Due to the timeline of the work undertaken in the thesis, there are differences in the selection of evaluation scenarios in Chapters 3 and 4; however, in Chapter 4, which presents a more sophistical model, the findings were re-evaluated on a subset of scenarios from Chapter 3.

Chapter 2

Background

The models developed in this dissertation make use of various concepts from game theory in order to model the interaction of road users in different traffic situations. In this chapter, I present the preliminary background of the concepts used throughout the thesis.

2.1 Normal form game

A normal form game is characterised by the tuple (N, A, U) where

- N is the set of players, indexed as $i \in N$, playing a game. These represent the set of interacting road users in a traffic situation.
- A_i is the set of actions that the player i can execute, and the set $A = A_1 \times ...A_N$ is the set of all such actions.
- $u_i : A \rightarrow \mathbf{R}$ is a mapping from the set of actions to a real number that represents the utility that a player i receives as a result of a set of actions chosen by each player. The set $U(a) = [u_1(a), ..., u_N(a)]$ where $a \in A$, is a vector of utilities of all players. For the models in this work, the utilities lie in the bounded interval [-1,1].

A pure strategy of an agent i in the game is an action $a_i \in A_i$ that a player i selects to play in the game. A mixed strategy π_i is a probability distribution over the domain of pure strategies A_i. A strategy profile is a combination of strategies of all players in the game.

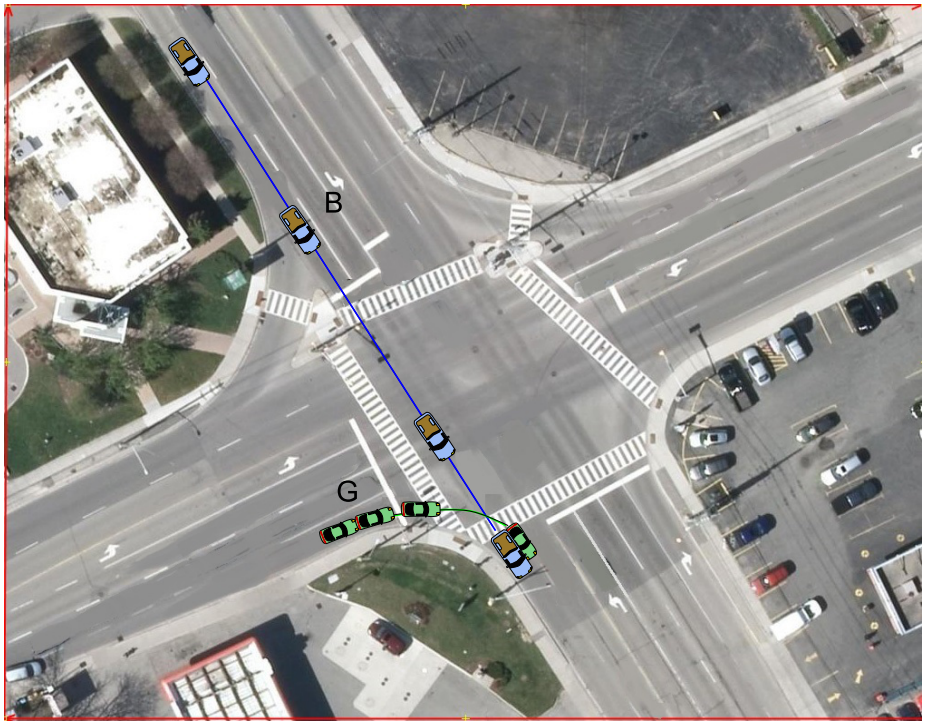


Figure 2.1: Running example of right turn interactions between two vehicles modelled as a game.

I use a running example of a right turn scenario where the interaction between two moving vehicles is modelled as a game (Fig. 2.1). In this game, the two players are the right turning vehicle (marked G) and straight through vehicle (marked B). The actions of the right turning vehicle are i) *stop* — in which the vehicle stops before the stopline, ii) *rolling stop* — in which the vehicle slows down but never stops completely, and iii) *proceed* — in which the vehicle executes a right-turn manoeuvre. The straight through vehicle's action space consist of three self explanatory actions, *speed up*, *slow down*, and *maintain same speed*. The normal form game for this scenario is illustrated in Table 2.1 with example utilities for each strategy.

	speed up	slow down	maintain
stop	0.6,0.75	0,0.1	0.2,0.5
rolling stop	0.4,0.5	0.75,0.2	0.5,0.3
proceed	-1,-1	1,0.2	-1,-0.1

Table 2.1: Utility matrix for the game of Fig. 2.1

.

2.2 Solution concepts

A *solution concept* is a process that maps a game to a set of action profiles $A^* \subseteq A$, commonly referred to as solving the game. Solving the game is often based upon the idea of a rational agent who wishes to maximise their own utility as a response to other players' actions. Formally, this is modelled through a *best response* function $BR_i(a_{-i}) = \arg\max_{a_i} u_i(a_i, a_{-i})$ where the index $-i$ refers to the set of all other players other than i. When the set of other players choose the set of action $a_{-i} \in A_1 \times ..A_{i-1} \times A_{i+1} \times ..A_N$, a rational player i as a response, chooses an action given by the function $BR_i(a_{-i})$.

2.2.1 Nash equilibrium

A *pure strategy Nash equilibrium* is a set of action profiles that is a result of each player playing their best response action to each other. Therefore, $a^* \in A$ is a pure strategy Nash equilibrium iff $\forall i$, $a_i^* = BR_i(a_{-i}^*)$. Another way to look at Nash equilibrium is to think of Nash equilibrium as a strategy profile in which no player can gain by unilaterally deviation to another strategy. In the example of Fig. 2.1, the strategy profiles (*stop,speed up*) and (*proceed,slow down*) are pure strategy Nash equilibria since neither of the two players can gain by deviating from jointly following one of these two strategy profile.

A *mixed strategy Nash equilibrium* is the generalisation of pure strategy Nash equilibrium when players select mixed strategy actions instead of pure strategies. Formally, a strategy profile $\pi \in |\triangle A_i|^N$ is in mixed strategy Nash equilibrium iff $\forall i$ and $\forall \pi_i$, $\mathbf{E}U_i(\pi_i^*, \pi_{-i}^*) \geqslant \mathbf{E}U_i(\pi_i, \pi_{-i}^*)$, where $\mathbf{E}U_i(\pi_i, \pi_{-i}) = \sum_{a_i} \pi_i(a_i) \sum_{a_{-i}} \pi_{-i}(a_{-i}) u_i(a_i, a_{-i})$, $\pi_i(a_i)$ is the probability of selecting the action a_i by i when playing the mixed strategy π_i, and $\triangle A_i$ is the space of all probability distributions over the domain of actions A_i.

2.2.2 Stackelberg model

A stackelberg model is a game between two players, a leader, and a follower. The model works under the constraint that both the leader and the follower are aware *ex ante* of their own assignment. The leader selects the first action in the game, and the follower, upon observing the leader's actions, selects their own action. The leader models the follower as a rational agent who best responds following the leader's action. Formally, the leader's strategy is modelled as follows.

$$a_{\mathrm{L}}^* = \arg\max_{a_{\mathrm{L}}} u_{\mathrm{L}}(a_{\mathrm{L}}, BR_{\mathrm{F}}(a_{\mathrm{L}}))$$

where $a_{\mathrm{L}} \in A_{\mathrm{L}}$ is one of the actions available to the leader and BR_{F} is the best response function of the follower. The follower's action is a best response to the leader's Stackelberg optimal action a^*_{L}, i.e., $a_{\mathrm{F}} = BR_{\mathrm{F}}(a^*_{\mathrm{L}})$. In the example of Fig. 2.1, if we consider the straight through vehicle to be the leader and the right turning vehicle to be the follower, the follower's best response to the three leaders action *speed up*, *slow down*, *maintain* are *stop*, *proceed*, *rolling stop* respectively. Of these three outcomes, since the leader gains the most by choosing *speed up*, which corresponds to the utility 0.75, the strategy profile (*stop,speed up*) is also an equilibrium in the Stackelberg model.

2.3 Boundedly rational models

Behavioural game theory deals with empirical models of behaviour of human players playing games in a naturalistic or laboratory setting. In these models, there is an allowance for systematic deviations from completely rational behaviour. Quantal Best Response (QBR) is a suboptimal response model in which players make cost proportional errors in their response.

$$QBR_i(a_{-i};\lambda) = \frac{\exp[\lambda \cdot u_i(a_i, a_{-i})]}{\sum_{a'_i} \exp[\lambda \cdot u_i(a'_i, a_{-i})]}$$

In the above model of response, λ is a *precision parameter* in a way that when $\lambda \to \infty$, QBR response is identical to a best response, and when $\lambda = 0$, the response models the selection of all actions with uniform probability.

A popular class of boundedly rational models considers that the population playing the game consists of a mix of people with different levels of *iterated strategic reasoning*. Depending on the depth of iterated reasoning that is involved, the behaviour is classified as levels 0 to k.

- *Level - 0* agent uniformly randomises from their set of actions A_i. Therefore, the probability of selecting any action a_i is given by $\pi_i^{\text{level-0}}(a_i) = \frac{1}{|A_i|}$

The behaviour of a level-1 player and players of higher levels depends on the specific model being considered. In level-k model of [53], the strategy of *level - 1* agent is given by

$$\pi_i^{\text{level-1}}(a_i) = \begin{cases} \frac{1-\epsilon_1}{|BR_i(\pi_i^{\text{level-0}})|}, & \text{if } a_i \in BR_i(\pi_i^{\text{level-0}}) \\ \frac{\epsilon_1}{|A_i \setminus BR_i(\pi_i^{\text{level-0}})|}, & \text{otherwise} \end{cases}$$

In the above model, a level-1 agent best responds to the level-0 agent's strategy $\pi_i^{\text{level-0}}$. However, they are modelled as imperfect agents and can select alternate actions with a small probability ϵ_1.

Level - 2 agent in this model believes the population to consist of level-1 agents and best responds with their own error ϵ_2. The special case where level-k ($k \geqslant 1$) agents are perfect responders can be modelled by setting $\epsilon_k = 0$, $k \geqslant 1$.

The model called Quantal level-k [234, 205] is similar in nature; however, instead of level-k$\geqslant 1$ agents' selecting a suboptimal action with probability ϵ_k, the error is modelled through the QBR function. The model of behavior for the level-0 ($\pi_i^{\text{QLk=0}}$) and level-1 ($\pi_i^{\text{QLk=1}}$) agent in this model is as follows.

$$\pi_i^{\text{QLk=0}}(a_i) = \frac{1}{|A_i|}$$

$$\pi_i^{\text{QLk=1}}(a_i) = QBR_i(\pi_i^{\text{Qlk=0}}, \lambda_1)$$

Level-2 agent in this model is aware of suboptimal nature of level-1 agent; however, they can have an erroneous perception of level-1 agent's precision parameter. Therefore, a level-2 agent's response is modelled as $QBR_i(\pi^{\hat{\text{Qlk=1}}}, \lambda_2)$, where $\pi^{\hat{\text{Qlk=1}}}$ is the misperceived strategy of level-1 agent.

2.4 Dynamic game

Whereas normal form games model situations of one time interaction among players, extensive form games in comparison models situations where players interact with each other multiple times in sequence. Formally, an *extensive form* game or a *dynamic game* is characterised by the tuple $(N, A, X, Z, \rho, \chi, \sigma, U, H)$

- N is the set of players, indexed as $i \in N$, playing a game. These represent the set of interacting road users in a traffic situation.
- A_i is the set of actions that the player i can execute, and the set $A = A_1 \times ...A_N$ is the set of all such actions.
- X is a set of non-terminal nodes, also called the choice nodes.
- Z is a set of terminal nodes. The union set $X \cup Z$ is the set of nodes that form the game tree.
- $\rho : X \to N$ is a node assignment function that assigns each choice node to a player $i \in N$ whose turn it is to act in the corresponding choice node.

- $\chi : X \rightarrow 2^{A_i}$ assigns each choice node a set of possible action that the assigned player i can take in that node.
- $\sigma : X \times A \rightarrow X \cup Z$ is the transition function that models the movement of a player to a new successor node (another choice node or a terminal node) after taking an action at node X.
- $u_i : Z \rightarrow \mathbf{R}$ is a mapping from the set of terminals to a real number that represents the utility that a player i receives as a result of the game play sequence that ends at a terminal node. The set $U = \{u_1, .., u_N\}$ represent the utility of all players in the game at a specific terminal node in Z.
- $H = \{H_1, .., H_N\}$: an information partition such that for each $x \in X$, $h(x)$ denotes the set of nodes given what the assigned player $\rho(x)$ knows at node x of the game. This captures the uncertainty of player $\rho(x)$ at a specific stage in the game, and if $x' \in h(x)$, then $\rho(x) = \rho(x')$, $\chi(x) = \chi(x')$, and $h(x') = h(x)$. $h(x)$ is called the *information set*, and the set $H_i = \{h(x) : \rho(x) = i\}$ is the set of all *information sets* of player i.

A game where $h(x)$ is a singleton, i.e., $h(x) = \{x\}; \forall x \in X$, represents no uncertainty for any player at any node in the game, and therefore such games are called *perfect information game*. Any game that is not of perfect information, is an *imperfect information* game. The structure of the games used in the thesis are all of perfect information.

Fig. 2.2 shows a nominal extension of the normal form game presented earlier for the scenario in Fig. 2.1 to the extensive form by adding a second sequence of action for player B (straight-through vehicle), who can now after observing the action of player G (right-turning vehicle) can choose to either *maintain speed* or *slow down*. The choice nodes assigned to the straight-through vehicles (marked in blue) are $\{X_1, X_4, X_5, X_6, X_7\}$, and the choice nodes assigned to the right-turning vehicle are $\{X_2, X_3\}$. Only two out of the three straight-through vehicle's actions are shown in the game tree for the sake of simplicity. The terminal nodes Z are shown as shaded nodes in the tree, and the two numbers represent the utilities of the straight-through vehicle and the right-turning vehicle, respectively. In this game of extensive form, the straight-through vehicle chooses an action at node X_1, which is observed by the right-turning vehicle, who in turn can choose their response at one of nodes X_2 or X_3 depending on the straight-through vehicle's action choice. In the third step, the straight-through vehicle can observe and select their next action at one of nodes X_4, X_5, X_6, X_7, after which the game ends and the utility is realized by each player. The purpose of this game is to illustrate the construct of an extensive form game. In a more practical setting, one can imagine a game like this to consist of arbitrary long sequences of action choices of each vehicle in an interactive traffic situation. Therefore, it is important to

make sensible design choices in order to make the games computationally tractable, and in the following chapters, the thesis offers guidance in that regard by evaluating some of those choices.

A pure strategy in an extensive form game is an realization of the function χ, which assigns an action (or a set of actions) for each player at every choice node that is assigned to that player. Similar to the normal form game, a pure strategy *profile* in extensive form is the combination of strategies of all the players in the game.

Solution concepts for extensive form games make use of structures in the game tree called *subgames*. A subgame of a perfect information extensive form game is the game that is constructed from a subtree rooted at every choice node $x \in X$. Therefore, the game of Fig. 2.2 has 7 subgames each starting at the choice nodes X_1 to X_7. The most commonly used refinement of Nash equilibrium in a perfect information extensive form game is *Subgame Perfect Nash equilibrium (SPNE)*. In order to define SPNE, let $s(g)$ be the part of the strategy s that denotes the strategies of the players in the subgame g. A SPNE strategy s^* is a strategy profile such that $s^*(g)$ is a Nash equilibrium for all $g \in G$. SPNE in a pure extensive form game can be calculated through the recursive procedure of *backward induction* as described in Algorithm 1 [203].

Algorithm 1: Backward induction for perfect information extensive form game

```
   input : node x
1  if x ∈ Z then
2  |  return U(x)
3  else
4  |  U* ← [−∞]^N
5  |  for a ∈ χ(x) do
6  |  |  U' ← backward_induction(σ(x, a))
7  |  |  if U'[ρ(x)] > U*[ρ(x)] then
8  |  |  |  U* ← U'
9  |  |  end
10 |  end
11 |  return U*
12 end
```

The algorithm does a depth-first traversal of the game tree by recursively calling the backward_induction procedure for each subgame. At each choice x, the optimal utility U^* is returned that corresponds to the action that maximizes the utility of the player $\rho(x)$ assigned to the corresponding choice node (lines 4-11). The SPNE strategy can be found by keeping track of the

strategy corresponding to u^*. Corresponding to the example in Fig. 2.2, the backward induction procedure would proceed at follows.

- At choice nodes X_4, X_5, X_6, X_7, the comparisons $-0.5 > -1, 0.9 > -1, 0 > -0.5, 0 > -0.5$ for the assigned player B would result in *slow down*, *maintain speed*, *maintain speed*, *maintain speed* being the optimal actions.
- At nodes X_3, X_4, for player G, *proceed* would fetch the highest utility 0.8 in both cases considering the optimal actions that B would choose as a result of the calculations from the previous step.
- Finally, at choice node X_1, the optimal action for player B is to *slow down* corresponding to utility 0, since the alternate action *maintain speed* will only fetch utility -0.5 conditioned on the optimal action of the other player in the rest of the game.

The SPNE for the two players are highlighted in the figure, which are $\{$(*slow down, slow down*),(*slow down, maintain speed*)$\}$ and $\{$(*proceed*)$\}$ for players B and G, respectively. If both players play according to their SPNE strategies, the realised utility will be [0,0.8] corresponding to the sequence of actions *slow down, proceed, maintain speed*.

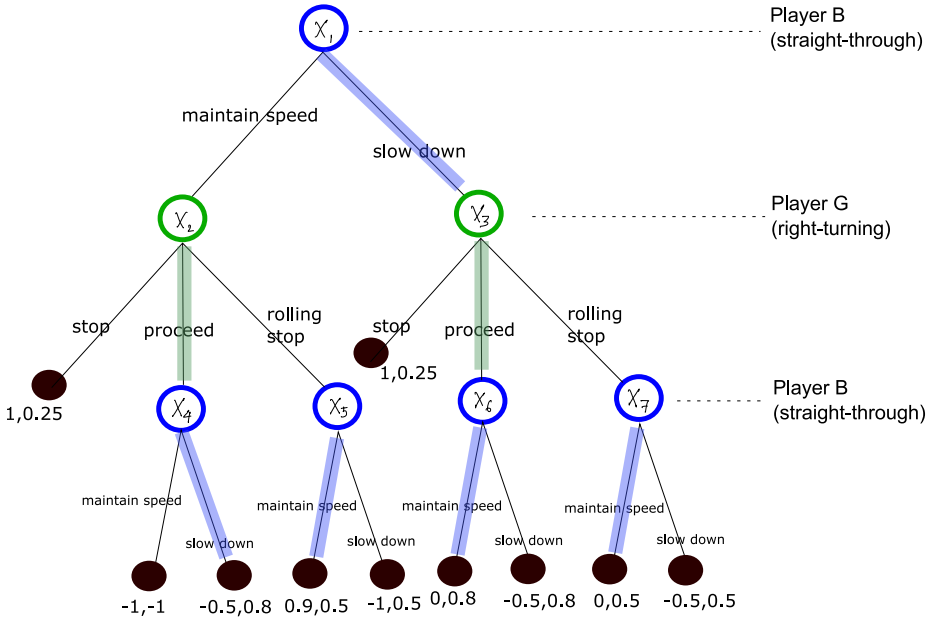


Figure 2.2: Example scenario of Fig. 2.1 modelled as a perfect information extensive form game by addition of a second action for player B (straight through vehicle). The highlighted strategy is the Subgame perfect Nash equilibrium (SPNE) of the game

Chapter 3

Behaviour models for hierarchical games

3.1 Introduction

Models for motion planning in autonomous vehicles and modelling human driving behaviour have typically represented drivers as completely rational, i.e., agents making decisions that are optimal given the constructs of the game [200]. This assumption of optimality takes the form of optimal response, i.e., selecting the best response to possible actions of other drivers, and optimal reasoning, i.e., the ability to reason strategically about how other drivers may choose their actions. In reality, especially since the construction of the game from the modeller's perspective may not reflect the actual game or the reasoning process undertaken by human drivers in real-world scenarios, the assumption of complete rationality may be too strong. The issues around rationality in this context is also exacerbated by the fact that there has been relatively less focus on empirical evaluation of the different game theoretic models, and therefore we do not have a clear understanding on how to quantify human *boundedly* rational behaviour for the case of driving.

In this chapter, I develop solution concepts for hierarchical games that support boundedly rational agents. I formalise the different solution concepts that can be applied in the context of hierarchical games (a framework used in multi-agent motion planning) for the purpose of creating game theoretic models of driving behaviour. Furthermore, based on a contributed dataset of human driving at a busy urban intersection with a total of 4k agents and 13k game instances, I evaluate the behaviour models on the basis of model fit to naturalistic data, as well as their predictive accuracy. I also study the impact of i) game construction, ii) agent response, and iii) agent reasoning, on the performance of the models. The results suggest that among the behaviour models evaluated, modelling driving behaviour as a model where drivers best respond to other

drivers with the belief that everyone else will follow the rules is the superior model of manoeuvre selection. Furthermore, bounds sampling of actions, i.e. by including the prototype and extreme trajectories under each manoeuvre, provides the best fit to naturalistic driving behaviour.

3.2 Motivation

Traditional approaches to motion planning have typically treated the problem as a single agent problem; in this perspective, a vehicle interacts with the environment (in simulation or on-field setting), possibly with the help of recorded human-driven trajectories, and plans its actions by optimizing over its objectives while taking into account the dynamic obstacles in the vicinity [200, 98]. However, in reality human driving is a complex system with a symbiotic relation among agents, where actions of a vehicle influence the future actions of other road users and vice versa. More recently, there has been a focus towards treating motion planning of AVs as a multi-agent problem with game-theoretic solutions to AV decision making [70, 190, 38, 136]. Such approaches can account for heterogeneous objectives in a group of vehicles in a traffic scene and identify equilibrium solutions that guide the actions of the AV. Given that the movement dynamics of a vehicle is in the continuous domain, it is intuitive to model the dynamics as a differential game, an approach adopted by multiple models in the literature [74, 190, 225]. However, the applicability of such games as a general purpose planner is limited by the trade off between the computational burden and expressivity; cases where efficient solutions exist in a multi-agent setting restrict the behaviour of the agents to only linear dynamics [74], and expanding behaviour to realistic nonlinear dynamic make problems computationally challenging. As an alternative, Fisac et al. [70] introduced the concept of a hierarchical game for AV planning where the game is decomposed into two levels: a long-horizon strategic game that can model richer agent behaviour, and a short-term tactical game with a simplified information structure. The motivation for using hierarchical game as a framework is based upon the belief that in order for the behaviour models to be used in in different stages of AV development, the model needs to seamlessly integrate within the AV motion planning architecture. This means that the construction of the games in terms of the agents, available actions, utilities, etc., should be based upon techniques that have already been developed for AV motion planning — and hierarchical games are a natural candidate for that purpose [70]. Although hierarchical games are well suited for AV considering that the idea of hierarchical decomposition of driving actions is well established in the literature [156, 229, 70], for the models to be applicable in real world situations, we need to understand how well the solution concepts in the game match naturalistic human driving behaviour. It is well known that in many realistic settings, the theoretical fixed point of Nash equilibrium is a poor predictor of human behaviour [89]; therefore, it is necessary to investigate

if the same is true for human driving behaviour too. In the absence of that information, we do not know whether the strategies followed by AV are the right ones or not.

Behavioural game theory provides a framework to analyse decision making in a naturalistic setting and models of behaviour that often have higher predictive power than Nash equilibria [39]. A key element in behavioural game theory is *bounded rationality*, where the conventional game-theoretic notion of agents as fully rational is relaxed to allow for sub-optimal behaviour. Such behaviour may arise from limitations in cognitive reasoning, error-prone actions [192], or it may also be that the behaviour is optimal with respect to *some* game structure but not the one the modeller chose to model the situation in question. In any case, driving is a cognitively demanding job that requires situational awareness and sophisticated visuomotor co-ordination, added on to individual habits, biases, and preferences; and it is not hard to imagine that driving at its core is a bounded rational activity. Consequently, it becomes essential for AV game theoretic planners to be able to characterize the bounded rational behaviour in human driving; for example, if humans are prone to making errors in judgement when the signal is about to turn red from amber at a busy intersection, then the AV planner should take that into account since the safety of the AV decision is conditioned on the error made by the human driver. Wright and Leyton-Brown developed a general framework of analysing and estimating parameters of popular behavioural game theory models based on observations of game play. They focus on two models of behaviour, i.e. Quantal Level-k (Qlk) and Poisson-Cognitive Hierarchy (P-CH) [235], which model iterated reasoning where agents have a limited capacity to maintain higher order belief about other agents. Although Qlk and P-CH do not capture all types of bounded rationality that one can think of in the case of human driving, such as the ones that arise from sampling the actions of other agents, the framework developed in [235] nevertheless can be applied to a wider set of behaviour models including the ones we develop in this chapter.

Developing a game-theoretic planner for an AV is a multi-step process, broadly involving a) selection of the right behaviour model and equilibrium concepts for other road agents, b) estimation of the parameters of the model, and c) generation of a safe maneuver and trajectory after accounting for the model and its parameters. In this chapter, I focus on the first two aspects. I also restrict my focus in this chapter to the single-shot moving horizon based setting, which is the planning process where agents play a fixed time horizon game at a constant planning frequency, starts execution of their action and replans again in the next time step. The contributions of this chapter are as follows.

- Formalisation of the concept of a hierarchical game along with the various solution concepts from behavioural game theory that can be applied to the solve such games.
- Development and evaluation of thirty behaviour models demonstrating different methods of game construction and solution concept choices for modelling traffic interactions.

3.3 Related work

One of the first works to include game theory as a methodology to model human driving behaviour is by Kita [113], where a lane change scenario is modelled as a two player game. The solution concept used in that work is a mixed strategy Nash equilibria of merge/give way behaviour. MLE (Maximum Likelihood Estimation) is used to estimate utility parameters based on data recorded by a video camera on a Japanese highway. Since [113], many game theoretic models have focused on lane change behaviour, and a recent review provides good coverage of this literature [103]. Since the interest of this thesis is autonomous vehicles, I review the relevant literature in that domain of application in more detail. During the period the thesis was in progress, there has been several works published in the area of game theoretic models and AV, and I categorise the relevant literature along key dimensions as shown in Table 3.1.

- *Structure*: This refers to the structure of the game as modelled by the available actions and the planning time horizon. Most of the works, including mine in this chapter, are of one shot games with moving horizon [241, 252, 67, 137, 144, 138, 85, 83, 81]. In this type of construction, agents play a normal form game (therefore the name one shot) constructed with respect to a fixed horizon. Based on the solution of the game, the agent starts to proceed with their action, and replan again by constructing a new game at fixed time interval, which is the planning frequency. This method of planning with a moving or receding horizon, in which only a part of the planned action is executed before replanning again, is commonly used in AV motion planning techniques [49]. An alternate way of solving the game is replacing the normal form game with a dynamic game, thereby supporting a richer strategy and action space. In this construction, the agent repeatedly plays a dynamic game, executes the action, and replans again at the terminal node. In a single-agent setting, this method of planning is similar to a Model Predictive Control (MPC) based planning [37]. Since the construction of the dynamic game may involve an exponentially large state space, Monte Carlo-based sampling methods are often used in the construction and evaluation of the game tree [216, 210]. I develop methods for dynamic game structure in Chapter 5.

- *Solution concept* : This categorisation is based on the solution concept used in solving the game. Stackelberg and Nash equilibrium are more commonly used solution concepts in this regard, and more recently, level-k [215] type methods have also been in use in order to support boundedly rational agents. Stackelberg solution lends well to the problem of traffic interaction, since many situations can be modelled as one agent being the leader and the other as the follower. For example, the driver holding the right of way can be modelled as a leader. However, since this assignment is part of common knowledge and has to be agreed

upon by both agents, the use of Stackelberg may be too restrictive due to this assumption. Geary et al. [83] show that the breakdown of this assumption can lead to dangerous situations (such as collisions and stopping on highways) and show that changing the utility structure to model aspects such as altruism can be one way to avoid such situations. The solution concept used in this chapter is based on a hierarchical decomposition of the game and is different for different levels of the game. Such a hierarchical decomposition with respect to the solution concept is seen in the work by Fisac et al. [70], and more recently, the use of local Nash equilibrium in [85] can also be interpreted as a form of hierarchical decomposition.

- *Empirical* : Although there have been several sophisticated models developed within a game theoretic setting, since the problem is one of modelling human behaviour, it is vital to judge the effectiveness of the models in real world setting. Therefore, we cover the literature on the basis of whether the proposed methods include empirical evaluation. Some works include evaluation only in simulation where the efficacy is demonstrated by performance in different scenarios in simulation. Other methods include experiments with human subjects in a driving simulator where either data is collected based on how humans drive in the simulator or models are evaluated against a human taking the role of one of the players. However, there is a gap in the literature on the evaluation of models based on naturalistic driving data in a real world setting, which my work in this chapter aims to address. Contemporary works that address the proposed game-theoretic models based on naturalistic data include [210] and [85], both of which were published in the similar time frame as this chapter.

- *Bounded rationality*: This categorization is based on whether the models have support for boundedly rational agents. Most game theoretic models are built upon the assumption that agents are completely rational, either through their ability to calculate a best response action often in conjunction with their ability to reason over the possible actions of other agents. This can be taken as a reasonable assumption when models are evaluated based on human in simulator studies, since it is possible to give all the relevant information to the participant, which includes the utilities, the actions, game structure, etc. However, in a real-world setting all of these elements are outside of the control of the game modeller, and therefore having support for boundedly rational agents (bounded from the perspective of the game modeller) becomes essential.

- *Scenario* : This column refers to the specific traffic scenario based on which the models are developed or evaluated. Typical examples include the lane change scenario (which has been the most common scenario studied), along with the scenarios of intersection and roundabout in recent years.

- *Utilties* : Driving is often a multi-objective activity that includes balancing multiple objectives such as safety, progress, comfort, and most of the literature shown in Table 3.1 is reflective of that. Typical dimensions along which the proposed methods model driver behaviour include safety, progress, comfort, adherence to lane and speed limits, along with behavioural attributes such as empathy and altruism in some cases. Since solution concepts needed to solve the games involve aggregation of the multivalued utility into a single real number, a linear weighting of the objectives is the most common method of aggregation. In some cases, the weights are estimated *a-priori* through a separate process based on techniques such as Inverse Reinforcement Learning [216], and in other cases fixed weights are also used. Alternatively, utilities can also be modelled purely through demonstrations as in [190], where any action that is more similar to the demonstrated action fetches higher utility and, therefore, the canonical dimensions of safety, progress, etc. are not modelled explicitly.

In relation to the literature presented above, the methods developed in this chapter fall under one-shot moving horizon in terms of the game structure, solution concepts include Nash equilibrium, Level-k, along with elementary non-strategic decision models, and intersection as the scenario of study. The methods also focus on modelling boundedly rational behaviour, and I evaluate the models based on naturalistic driving data.

3.4 Hierarchical Games

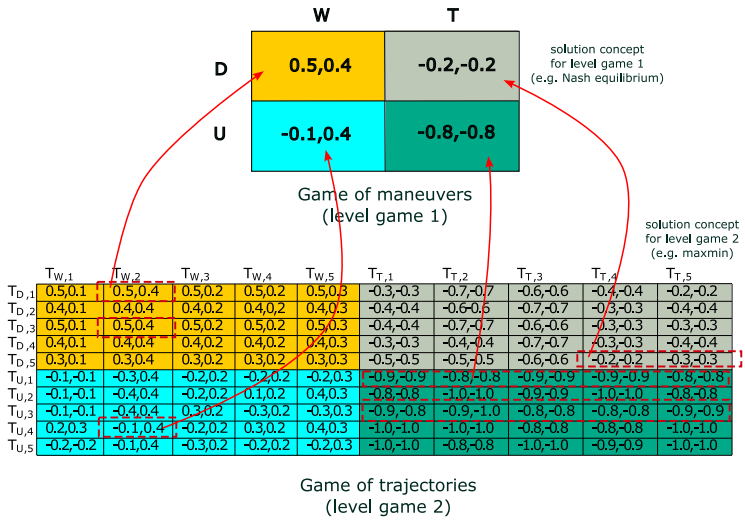
Prior to the recent focus on autonomous driving, there has been a considerable body of research on modelling driving behaviour within the field of traffic psychology with a long history of treating driving behaviour as a hierarchical model [110, 221, 132, 156]. One of the more influential models, the Michon hierarchy of driving tasks [156], decomposes driving into three levels of control; a strategic plan such as a route and general speed choice of going from point A to B is decomposed into several tactical decisions of choosing the right manoeuvres, which is further decomposed into high-fidelity actions that control the steering and acceleration. A primary motivation of a hierarchical decomposition is that drivers have different motivations and risk judgements at each level of the hierarchy, and the functional decomposition into a hierarchical system allows modelling of risk and safety considerations separately at each level. A driver for example may be indifferent about the choices at a granular level of trajectories but care more about choosing the right manoeuvre, e.g., waiting for an oncoming vehicle. Motion planners in autonomous vehicles also follow a similar hierarchical pattern of decomposition; a high level *route planner* plan is given to a *behaviour planner*, which sets up the tactical manoeuvres for

	Year	Structure	Solution concept	Evaluation	Bounded rationality	Scenario	Utilities
Yu et al.. [241]	2018	One shot moving horizon	Stackelberg	In simulator, simulation.	No	Lane change	Distance gap, Time gap
Zimmerman et al.. [252]	2018	One shot moving horizon	Nash	In simulator	No	Lane change	Cooperative utility, time pressure, safety.
Castro [67]	2019	One shot moving horizon	Mixed strategy Nash	Simulation only	No	Trolley problem	Safety.
Li et al.. [137]	2019	One shot moving horizon	Stackelberg	Simulation only	No	Lane change	Safety, progress, effort.
Li et al.. [139]	2019 (2016)	Dynamic game	Level-k	Simulation only	Yes	Lane change	Progress, safety, effort.
Sadigh et al.. [190]	2016	MPC	Level-k	In simulator, simulation	Yes	Lane change	Learned through IRL.
Fisac et al.. [70]	2019	Dynamic	Stackelberg, maxmax	Simulation only	Yes	Lane change	Safety, progress.
Liniger and Lygeros [144]	2019 (2017)	One shot moving horizon	Nash, Stackelberg	Simulation only	No	Racing	Progress, safety, *blocking*.
Sun et al.. [211]	2018	MPC	Nash	Simulation, utility estimation on NGSIM	No	Lane change, intersection	IRL based.
Li et al.. [138]	2020	One shot moving horizon, Dynamic.	Stackelberg, Level-k	Partially naturalistic (selected behaviour replication)	Yes	Uncontrolled intersection	Safety, progress.
Geiger and Straehle [84]	2020	One shot moving horizon	local Nash equilibrium	Naturalistic	No	Lane change	Progress, speed limit adherence.
Geary et al.. [83]	2021	One shot moving horizon	Stackelberg	In simulator, simulation	No	Lane change	Progress, speed limit adherence. Altruism.
Tian et al.. [216]	2021	Dynamic game (Monte Carlo)	Level-k	In simulator, simulation	Yes	Lane change	Safety, progress, comfort.
Sun et al.. [210]	2020	Dynamic game (Monte Carlo)	Nash, Stackelberg, Pareto	Naturalistic	No	Roundabout	Safety, speed limit adherence.
Garzón and Spalanzani [81]	2020	One shot moving horizon	Level-k	Simulation only	No	Merge	Goal completion.
This chapter	2020	One shot moving horizon	Level-k (with variations), Nash equilibrium with Quantal Errors	Naturalistic	Yes	Intersection	Safety and progress

Table 3.1: Relevant literature on the application of game theoretic models for autonomous vehicles.

a lower level *trajectory planner*, which in turn generates the trajectory profile for the vehicle *controller* after respecting its nonholonomic constraints. In addition to the motivation mentioned above, treating the problem of planning as a hierarchical system is also driven by computational efficiency, as previously shown in [70].

3.4.1 Illustrative example



	$T_{W,1}$	$T_{W,2}$	$T_{W,3}$	$T_{W,4}$	$T_{W,5}$	$T_{T,1}$	$T_{T,2}$	$T_{T,3}$	$T_{T,4}$	$T_{T,5}$
$T_{D,1}$	0.5,0.1	0.5,0.4	0.5,0.2	0.5,0.2	0.5,0.3	-0.3,-0.3	-0.7,-0.7	-0.6,-0.6	-0.4,-0.4	-0.2,-0.2
$T_{D,2}$	0.4,0.1	0.4,0.4	0.4,0.2	0.4,0.2	0.4,0.3	-0.4,-0.4	-0.6-0.6	-0.7,-0.7	-0.3,-0.3	-0.4,-0.4
$T_{D,3}$	0.5,0.1	0.5,0.4	0.5,0.2	0.5,0.2	0.5,0.3	-0.4,-0.4	-0.7,-0.7	-0.6,-0.6	-0.3,-0.3	-0.3,-0.3
$T_{D,4}$	0.4,0.1	0.4,0.4	0.4,0.2	0.4,0.2	0.4,0.3	-0.3,-0.3	-0.4,-0.4	-0.7,-0.7	-0.3,-0.3	-0.4,-0.4
$T_{D,5}$	0.3,0.1	0.3,0.4	0.3,0.2	0.3,0.2	0.3,0.3	-0.5,-0.5	-0.5,-0.5	-0.6,-0.6	-0.2,-0.2	-0.3,-0.3
$T_{U,1}$	-0.1,-0.1	-0.3,0.4	-0.2,0.2	-0.2,0.2	-0.2,0.3	-0.9,-0.9	-0.8,-0.8	-0.9,-0.9	-0.9,-0.9	-0.8,-0.8
$T_{U,2}$	-0.1,-0.1	-0.4,0.4	-0.2,0.2	0.1,0.2	0.4,0.3	-0.8,-0.8	-1.0,-1.0	-0.9,-0.9	-1.0,-1.0	-0.8,-0.8
$T_{U,3}$	-0.1,-0.1	-0.4,0.4	0.3,0.2	-0.3,0.2	-0.3,0.3	-0.9,-0.8	-0.9,-1.0	-0.8,-0.8	-0.8,-0.8	-0.9,-0.9
$T_{U,4}$	0.2,0.3	-0.1,0.4	-0.2,0.2	0.3,0.2	0.4,0.3	-1.0,-1.0	-1.0,-1.0	-0.8,-0.8	-0.8,-0.8	-1.0,-1.0
$T_{U,5}$	-0.2,-0.2	-0.1,0.4	-0.3,0.2	-0.2,0.2	-0.2,0.3	-1.0,-1.0	-0.8,-0.8	-1.0,-1.0	-0.9,-0.9	-1.0,-1.0

Figure 3.1: An example of a two level hierarchical game with action level game 1 being the game of manoeuvres and action level game 2 is the game of trajectories. Different solution concepts can be used at different levels to find a game solution.

I first explain the construction of a hierarchical game through a simple illustrative example in this section, followed by a more formal construction in the next section. Fig. 3.1 shows an example hierarchical game played between a vehicle turning right (column player) and a straight-through vehicle (row player) in an unprotected right turn at an intersection. This scenario is similar to the one shown in Fig. 3.3 where the vehicle with id:14 is the right turning vehicle and the vehicle with id:26 is the straight-through vehicle. For the sake of simplicity of this example, I show only a 2 player game between 14 and 26 rather than an N player game involving all the relevant vehicles in the scene. Let us say the right turning vehicle has two high level manoeuvres available to them, *wait* (W) or *turn* (T), and similarly the straight through vehicle has two manoeuvres *slow down* (D) or *speed up* including the special case of maintaining their current speed (U). The high-level manoeuvre determines the velocity profiles, and then sets

up the constraints expressed in terms of the target velocity ranges for a low-level trajectory planner (refer to section A.1 in the appendix for a detailed description of the trajectory generation process). For example, if v_0 is the vehicle speed during the initiation of the game, the slowing down manoeuvre may involve setting a constraint for the trajectory generation process with a target velocity in the range $[v_T^{\min}, v_T]$ where $v_T < v_0$ and $v_T^{\min}$ is the minimum velocity reachable by the vehicle in T seconds after taking into account the kinematic limits of the vehicle. Assuming that the vehicles use a method for sampling the possible trajectories (I elaborate on this later in the chapter), the trajectories are the main actions that the vehicles can execute in the game. Therefore, the set of trajectories for both vehicles forms the game matrix, as shown in the lower matrix of Fig. 3.1. Each coloured section of the matrix represents the trajectories corresponding to a specific combination of high-level manoeuvres. For now, let us assume that the utilities in the table are calculated after taking into account the various objectives such as safety, progress, etc. In the worst case scenario, finding a pure strategy Nash equilibrium of the game involves a quadratic time algorithm in terms of the size of the matrix that grows exponentially with the number of players [188]. As an alternative, one can use a different solution concept that runs in linear time, e.g. maxmax (selection of the utility maximizing action), or does not involve pairwise comparison, e.g. maxmin (action that maximizes the utility of worst case scenario with respect to other agents' actions), for the subgame under each manoeuvre combination. Each shaded matrix in the game of trajectories can then be replaced by a representative solution (in this example I take a sample maxmin solution that maximizes the sum of utilities), thereby forming the game of manoeuvres as shown in the top matrix in the figure. The game of manoeuvres can subsequently be solved using a more computationally involved solution concept such as a pure strategy Nash equilibrium. The key idea behind hierarchical game construction is the use of heterogeneous solution concept at different levels based on the hierarchy of actions (manoeuvres and trajectories in our case). This is akin to the possible mental process involved in driving, where a driver deliberates more at the level of manoeuvres but less so at the level of individual trajectories within each manoeuvre combination.

3.4.2 Formalization

In this section, I formalise the construct of a hierarchical game for the general case of N players and $\mathcal{K}$ levels of action hierarchy. Wherever possible, I use the term *action level* to disambiguate between the levels of action hierarchy, and levels of cognitive hierarchy that appears later in the chapter as part of the behaviour models to solve the games. I also highlight the relation between solving a hierarchical game in the way that was illustrated in the example above and subgame solving through backward induction. A hierarchical game is formulated by

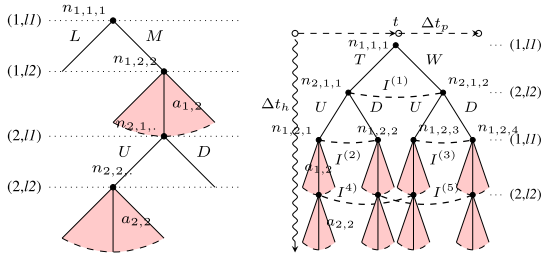


Figure 3.2: Illustration of two instances of hierarchical games. (a) As a Stackelberg game modelling a lane change maneuver and (b) simultaneous move game modelling intersection navigation. A hierarchical game is instantiated every Δt_p seconds with action plan of Δt_h seconds.

- Set of N agents indexed by $i \in \{1, 2, 3, ..N\}$.
- A set of $\mathcal{K}$ levels indexed by $\kappa \in \{1, 2, 3, ..\mathcal{K}\}$.
- Set of actions $A_{i,\kappa}$ available to each agent i at level κ.
- A strategy s_i for agent i is a $\mathcal{K}$-tuple $s_i = (a_{i,1}, a_{i,2}, .., a_{i,\mathcal{K}})$ where $a_{i,\kappa} \in A_{i,\kappa}$ and the strategy space of s_i is $\prod_{\kappa \in \mathcal{K}} A_{i,\kappa}$.
- A set of states X_i of agent i in level 1, and an initial mapping function $f_{i,1} : X_i \to \mathcal{P}(A_{i,1})$ that maps the initial state of the agent to the available actions in level 1, where $\mathcal{P}(\cdot)$ is the power set.
- Set-valued functions $f_{i,\kappa} : \prod_{j=1}^{\kappa-1} A_{i,j} \to \mathcal{P}(A_{i,\kappa})$ for each agent i that maps a partial strategy $(a_{i,1}, a_{i,2}, .., a_{i,\kappa-1})$ to $\mathcal{P}(A_{i,\kappa})$ and gives the set of available actions to i in level $\kappa > 1$ for the partial strategy till level $\kappa - 1$.
- Set of N pay-off (utility) functions $U = \{u_i(s_i, s_{-i})\}$, where $-i$ refers to all agents other than i.

The hierarchical game imposes a total ordering in actions $A_i = \{A_{i,1}, A_{i,2}, .., A_{i,\mathcal{K}}\}$ of a given agent, and along with $f_{i,\kappa}$ induces a game tree, as shown in Fig. 3.2b. The frequency at which a hierarchical game is instantiated (Δt_p) and the time horizon of each strategy (Δt_h) are exogenous to the model. Each node is labeled $n_{i,\kappa,j}$, where i and κ are the agent and level indices, and j is the node identifier within level κ. This general formulation of a hierarchical game does not

prescribe a fixed information structure, and allows the modeller to set an information structure that is appropriate to the environment and situation they want to model. For example, Fisac et al. [70] models a lane change scenario where an AV merges into a lane occupied by a human driven vehicle. The game is modelled as a Stackelberg game with the AV being the leader and the human driven vehicle responding to the action of the AV. Fig. 3.2a shows a 2-agent 2-level the game tree for such a scenario where each decision node is a singleton information set since for every decision node, the agent who owns the decision node has perfect information on where they are in the game. At node $n_{1,1,1}$, the AV (indexed as agent 1) has the choice of either staying in its current lane (L) or merging into the adjoining lane (M) $A_{1,1} = \{L, M\}$. Conditioned on this choice, the vehicle has to generate a trajectory $a_{1,2} \in f_{1,2}(a_{1,1})$ to execute the maneuver chosen in level 1. Whereas actions in $A_{1,1}$ are discrete choices, the agent can choose from a continuum of actions (shaded region in the figure) at node $n_{1,2,2}$. The human driven vehicle after having observed the actions of the AV, can respond by deciding to speed up (U) to dissuade the merging AV cut-in the front, or slow down (D) followed by a trajectory that corresponds to the choice. In situations where assignment of a leader and a follower is unclear or that assumption is too strong, the agents might not have perfect information on the state of the play. Interactions at an intersection for example, are such scenarios. Continuing from the illustrative example from the previous section, Fig. 3.2b illustrates a 2-agent 2-level scenario as an example where an AV (indexed as 1) executes a free right turn on red at a signalized intersection (in a situation similar to id:14 in Fig. 3.3), while a human driven vehicle (id:26 and re-indexed as 2 in Fig. 3.2b) approaches cross path from left to right. The AV can either decide to turn (T) or wait (W) for the cross path vehicle to pass, i.e., $f_{1,1}(X_1) = A_{1,1} = \{T, W\}$. The human driven vehicle (id:26) can either slow down (D) or choose not to slow down (U), $f_{2,1}(X_2) = A_{2,1} = \{D, U\}$. Since either agent does not have perfect information about what the other agent is about to do next, agent 2 does not know whether they are in node $n_{2,1,1}$ or $n_{2,1,2}$ (connected by the information set $I^{(1)}$). This imperfection of information is also reflected at the trajectory level (level 2 actions), where each agent can only distinguish between the nodes in level 2 that follow from their own chosen actions in level 1, but not from the ones that follow from the other agent's level 1 decision ($I^{(2)}$-$I^{(5)}$).

It becomes apparent from this structure that the game has no proper subgame, and the game reduces to a simultaneous move game. It is well understood that a way to solve such games is by reduction to normal form. However, as we shall see, the hierarchical game has additional constraints that allow solving the game in Fig. 3.2 also through backward induction. To designate the nodes where utilities accumulate at each level in the backward induction process, we label a set of nodes in each level κ as *level roots* $\mathcal{L}(\kappa) = \{n_{i,\kappa,j} | parent(n_{i,\kappa,j}) \notin \mathcal{N}_\kappa\}$ where $\mathcal{N}_\kappa$ is the set of nodes in level κ. In other words, the set of level roots contain nodes in each level κ whose parent is not in level κ. Therefore, $\mathcal{L}(1) = \{n_{1,1,1}\}$ and $\mathcal{L}(2) = \{n_{1,2,1}, n_{1,2,2}, n_{1,2,3}, n_{1,2,4}\}$.

Algorithm 2: Backward induction for a hierarchical game

Result: S_1^*, V_1^*

for $\kappa := \mathcal{K}; \kappa = 1;\ \kappa := \kappa - 1$ **do**

  for $n \in \mathcal{L}(\kappa)$ **do**

    $S^*_{\kappa,n}, V^*_{\kappa,n} \leftarrow$ solve $\mathcal{G}_\kappa(\prod_{i=1}^{N} f_{i,\kappa}(\sigma_i(n)),$

    $\kappa = \mathcal{K}?U; V^*_{\kappa+1,\mathcal{L}(\kappa+1)})$

  end

end

Algorithm 2 shows the standard backward induction process adapted to the hierarchical game. The algorithm starts at the bottom most level ($\mathcal{K}$) and recursively moves up the tree by solving the action level games $\mathcal{G}_\kappa$ at every level. At each level, a simultaneous move *action level game* $\mathcal{G}_\kappa$ is instantiated from each node in $\mathcal{L}(\kappa)$. These action level games are constructed by first extracting $\sigma_i(n)$, which gives the partial pure strategy for agent i that lies on the branch from the root node of the game tree $\mathcal{L}(1)$ to node $n \in \mathcal{L}(\kappa)$. $f_{i,\kappa}$ gives the available actions for each agent i in the current level κ, and these actions form the domain of available strategies in the action level game $\mathcal{G}_\kappa$. The utilities depend on the level of the game; for action level game $\mathcal{G}_{\kappa=\mathcal{K}}$ the utilities are same as the game utility U, whereas for action level games $\mathcal{G}_{\kappa<\mathcal{K}}$ are solved based on the game values $V^*_{\kappa+1,\mathcal{L}(\kappa+1)}$ from the game $\mathcal{G}_{\kappa+1}$ solved in the previous iteration. Note that the pseudocode shows only the case where a single solution and game value $(S^*_{\kappa,n}, V^*_{\kappa,n})$ is propagated up the hierarchy. In the case of multiple solutions for the action level games, the strategies and values have to be tracked and repeated for each solution. The solutions and game value $S^*_{\kappa,n}, V^*_{\kappa,n}$ depend on the solution concept used for the individual action level game, and this is discussed in detail later under Solution concepts.

Due to the tree-like structure of the hierarchical game and the process of solving the game bottom up from the leaf nodes through backward induction, one can see that the process is very similar to solving for subgame perfect equilibria in multi-stage games with stages being replaced by levels in the hierarchy [213]. However, it is not a subgame perfect equilibria, since the action level games are not subgames in the game tree. The connection between subgame perfect equilibria and the solution in the hierarchical game is that in the hierarchical game, the mapping functions $f_{i,\kappa}$ impose an action selection method that is similar to action selection based on the condition of *sequential rationality* in subgame perfect equilibria. The mapping functions $f_{i,\kappa}$ eliminate strategies for all agents i that are not direct successors of the partial strategies $\sigma_i(n) \cdot \sigma_{-i}(n)$, essentially breaking any information set within a level κ that spans two separate levels in $\mathcal{L}(\kappa)$. To illustrate this more intuitively, take the example in Fig. 3.2(b) when agent

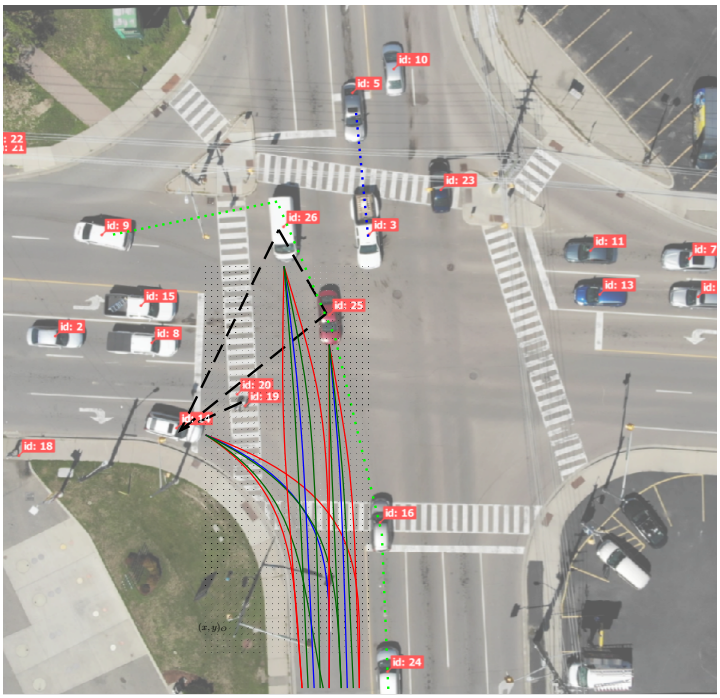


Figure 3.3: A snapshot of the intersection traffic scene. Representative trajectories based on the three sampling schemes over a R^3. The figure shows the path (R^2) projection of the trajectories and the dimension of time not represented in the figure.

1 is in the information set $I^{(2)}$ connecting the two nodes $n_{1,2,1}$ and $n_{1,2,2}$. At this information set, agent 1 could solve the game under two nodes separately only under a guarantee that after solving the game starting from node $n_{1,2,1}$ and following that strategy, under no condition would they find themselves at a leaf node of the tree starting at $n_{1,2,2}$, and vice versa. The function $f_{i,\kappa}$ provides that guarantee and eliminates hypothetical strategies where, at level 1, a vehicle may think about slowing down, but at level 2 chooses a trajectory that speeds up; and the fact that this cannot happen is part of the common knowledge among the agents in the game mediated by $f_{i,\kappa}$.

3.5 Game Structure

In this section, I describe the details of the game structure, including the number of agents, actions, strategies, and utilities.

3.5.1 Relevant agents and available actions

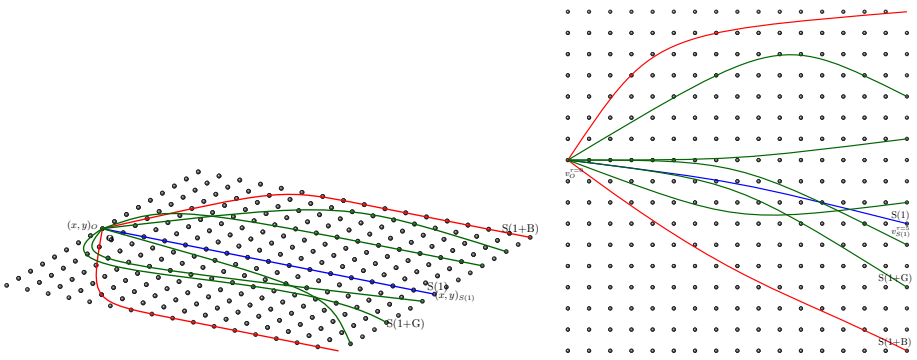
Since we are interested in investigating decision making in situations where there may be strategic reasoning involved, we extract situations where vehicles are executing a left turn across path or unprotected right turns. At each time step Δt_p=1s, we setup a hierarchical game with an action plan horizon of Δt_h=5s into the future from the perspective of each vehicle that turns left or right in the scenario. For example, in the snapshot of Fig. 3.3, the black dashed line shows the game from the perspective of the vehicle 14, which is turning right. The process of including the relevant vehicles in the game is as follows: we identify the conflict points on the map with respect to all the lanes in the intersection that cross each other. Since a game is initiated with respect to a 'subject vehicle', we first locate the conflict points corresponding to the lane that the subject vehicle is currently in, and include all agents in the scene that are on the lanes in conflict with the subject vehicle's lane. We also include the leading vehicle of these conflicting vehicles. This set of relevant vehicles along with the subject vehicle form the set of agents in each game. Pedestrian actions are not modelled explicitly in the game tree; however, their influence is modelled in the utility structure of the game, which is described later in the section. Further details about the conflict point and the process of relevant vehicle assignment is also included in appendix B.

Each game is a N-player 2-level hierarchical game where level 1 actions for each agent are high level manoeuvres that are relevant to the task under execution, and level 2 actions are the corresponding trajectories. We setup the set of manoeuvres with the help of a first order logic rules (appendix A.2) that takes into account the task of the vehicle and its situational state. The complete list of level 1 actions is documented in Table 3.4. Level 2 actions ($A_{i,2}$) are trajectories that are generated based on the actions in level 1. To generate the trajectories for each vehicle, we use a lattice sampling based trajectory generation similar to one presented in [251]. First a set of lattice endpoints are sampled on R^2 cartesian co-ordinate centered on the vehicle's current position. Each lattice sample point on R^2 is then extended with a temporal lattice which is re-sampled to form the final lattice points in R^3 that contain the (x, y) positions and the target velocity at each lattice point after accounting for acceleration and jerk limits of passenger vehicles [14]. Finally, the sampled lattice points are connected with a smooth cubic spline that adhere to the velocity, acceleration, and jerk constraints, representing the vehicle trajectory (Fig. 3.4). Appendix A.1 explains this process in more detail.

Since the trajectory generation is in continuous space with infinite actions for the drivers to reason over, combined with the time constraints to make a decision (which is in the order of milliseconds), the situation is ripe for bounded rationality to be in play — in a form where agents look at the game through samples[1] of action rather than all possible trajectories over the

[1] The usage of the word *sample* here means an *example*, and does not necessarily imply a presence of a specific statistical sampling procedure over a distribution.

continuous space. This form of bounded rationality is connected to Osborne and Rubinstein's model where agents' employ a mental process to sample other agents' actions and respond based on the imagined outcome of those samples [171]. In Osborne and Rubinstein's model, the model further develops this view of bounded rationality into an equilibrium (sampling equilibrium), whereas in our case I use this only as a method of action construction. I also use a common sampling procedure for all agents, and this enables the agents to have the same view of the action space and therefore play the same game. In the context of driving, this process is the same as when vehicles sample a set of trajectories of other agents and respond in accordance to those sampled trajectories with the additional assumption of having a common sampling scheme. Naturally, one may imagine that some sampling procedures make more sense than others, and in some procedures the assumption of a common sampling scheme is more reasonable than others. I now briefly mention the sampling procedures used in our experiments, and the intuitive reasoning behind each.



(a) (b)

Figure 3.4: Representative trajectories based on the 3 sampling schemes over a R^3 lattice showing the spatial representation of the (a) path and (b) velocity profiles. Lattice points are connected with cubic splines.

At each time step when the game tree is instantiated, agents observe the current attributes (such as position, velocity and acceleration) of other relevant agents in the game tree and use one of the following sampling methods to construct the game tree.

- S(1): In the most basic case, an agent i may sample a single trajectory (level 2 action) of every agent $-i$ that that they, i.e., i, think is most representative of the level 1 action of the agent they are currently reasoning over. To construct the trajectory sample, we select lattice endpoints along the lane centerline and use a piecewise constant acceleration model to generate the final trajectory.

- S(1+B): With a little more cognitive bandwidth, along with the $S(1)$ trajectory sample, they can also sample trajectories that form the extreme ends of the bounded level-2 action space of other agents. These trajectories are bounded spatially by the lane boundaries and temporally by the upper and lower bounds on the velocity limits of the level 1 action they correspond to. There are a total of 9 trajectory samples that are generated from this scheme; 3 velocity profiles generated over each of 3 path samples. This set of trajectories indicate what other agents might do in normative (i.e. following the rules as captured by the piecewise constant acceleration model) as well as in the extreme case but still within the physical limitations of the vehicle.

- S(1+G): The final sampling scheme lies in between the two schemes. Similar to $S(1+B)$, this scheme includes the $S(1)$ trajectory; however, the rest of the trajectories are sampled from a multivariate Gaussian distribution with $\mu = [x_{S(1)}, y_{S(1)}, v_{S(1)}]^\top$ and an unit diagonal covariance matrix, where $(x_{S(1)}, y_{S(1)}, v_{S(1)})$ is the lattice endpoint corresponding to the $S(1)$ trajectory. We refer to this scheme as $S(1+G)$ and the samples include the normative behaviour that comes from $S(1)$ along with variations in the path and velocity of the vehicle but not to the extremes that were captured in the $S(1+B)$ scheme.

One can see that S(1) and S(1+B) are methods of action construction that do not *sample* from a distribution in a statistical sense, and therefore the only assumption is that the agents share the general method of the action construction and the limits of the vehicle movement. In S(1+G) however, the specific samples of the distribution also need to be a part of the common knowledge for the agents to play the same game. In practice, this is a restrictive assumption; however, I include this method solely for comparison with other approaches.

3.5.2 Utilities

To determine the utility structure, we draw from motivational aspects of driver behaviour modelling in traffic psychology literature [209]. In general, driving motivations are multiobjective and the broad dimensions can be classified into *inhibitory* and *excitatory* motives. Whereas excitatory motivations drive a driver to make progress towards reaching the destination, inhibitory motivations are the balancing factors that account for mitigating crashes and mental stress. The three different utlities used in this work are as follows

- *Excitatory utility*. the degree of progress a driver can make based on a selected trajectory $a_{i,2}$ is the excitatory utility $u_{\text{v\_exc}}(a_{i,2})$ as determined by the trajectory length $\|a_{i,2}\|$, $u_{\text{v\_exc}}(a_{i,2}) = \min(\frac{\|a_{i,2}\|}{d_g}, 1)$, where d_g is a constant and can be interpreted as the distance

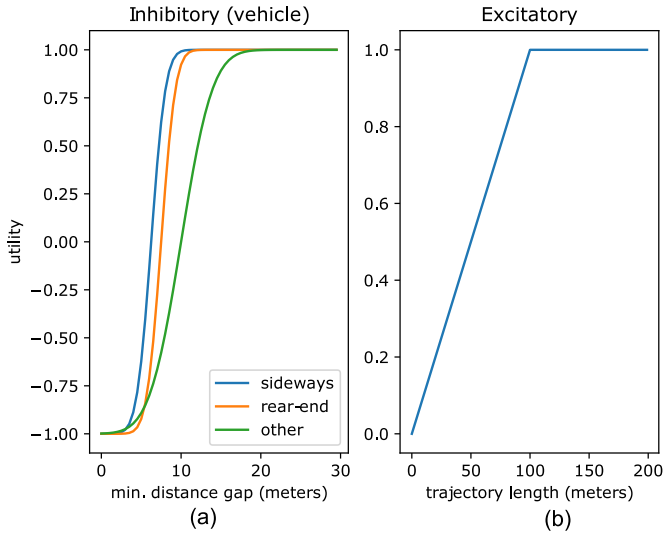


Figure 3.5: Utility function that maps a) minimum distance gap between trajectories to an utility interval [-1,1] (inhibitory utility for vehicle-vehicle interactions), and b) trajectory length to the utility interval [0,1] (excitatory utility).

to goal or crossing the intersection. We set $d_g = 100$ (Fig. 3.5b), which is approximately the distance to cross the intersection we study in our experiment. The minimum excitatory (i.e., progress) utility is set to 0 instead of -1 since even transiently waiting for another vehicle (trajectory length 0) may invoke a feeling of making some progress towards their destination.

- *Vehicle inhibitory utility*. This utility, $u_{\text{v\_inh}}$, is based on the minimum distance gap between pairs of trajectories of vehicles. The form of the utility function is a sigmoidal function (Fig. 3.5a), which are a popular family of functions that can map preferences over surrogate metrics, e.g., minimum distance gap $d(a_{i,2}, a_{-i,2})$, into an utility interval [71]. For $u_{\text{v\_inh}}$, we first fix a minimum safe distance gap $d^*_{a_{i,2},a_{-i,2}}$ based on collision type that may occur as a result of the distance gap reaching zero between the trajectories of the agents in the game. Collision types include side collisions, rear-end collisions, and other types that include angle collisions. The value of the safe distance gap determines the location θ of the sigmoidal function (erf). However, since the conception of what is considered safe may vary in a population of drivers, we let θ to be a random variable that is normally distributed with $\mu = d^*_{a_{i,2},a_{-i,2}}$ and constant variance σ determining the scale of the sigmoidal function. The choice of erf as the sigmoidal function is a mathematical convenience since the Gaussian integral of the erf in $u_{\text{v\_inh}}(a_{i,2}, a_{-i,2})$ evaluates to another

sigmoidal $\text{erf}(\frac{d(a_{i,2},a_{-i,2})-d^*_{a_{i,2},a_{-i,2}}}{2\sigma})$.

$$u_{\text{v\_inh}}(a_{i,2}, a_{-i,2}) = \int \text{erf}\left[\frac{d(a_{i,2}, a_{-i,2}) - \theta}{\sigma\sqrt{2}}\right] \mathcal{N}(\theta; d^*_{a_{i,2},a_{-i,2}}, \sigma) d\theta \tag{3.1}$$

- *Pedestrian inhibitory utility*. This utility, $u_{\text{p\_inh}}$, is a step function over [-1,1] such that $u_{\text{p\_inh}}(a_{i,2}) = -1$ if $a_{i,2}$ is a trajectory that does not wait for a pedestrian when the pedestrian is in the vicinity having a right of way, or is on the crosswalk to be traversed; and 1 otherwise.

The above multi objective utilities are aggregated using a weighted aggregation method with a weight vector $\mathbf{W}$ that combines inhibitory and excitatory utilities to produce a single real value.

$$u_i(a_{i,2}, a_{-i,2}) = \mathbf{W} \cdot \begin{bmatrix} u_{\text{v\_inh}}(a_{i,2}, a_{-i,2}) & u_{\text{p\_inh}}(a_{i,2}) & u_{\text{v\_exc}}(a_{i,2}) \end{bmatrix}^\top \tag{3.2}$$

The utilities for the actions in $\mathcal{G}_1$ depends on the solution concept (discussed in the next section), and can be calculated as follows. $u_i(a_{i,1}, a_{-i,1}) = V^*_{2,\eta}(i)$, where η is the leaf node of the branch $a_{i,1}, a_{-i,1}$ and $V^*_{2,\eta}(i)$ is the utility of agent i following the pure strategy response $a^*_{i,2}$, where $a^*_{i,2}$ is the solution to the underlying $\mathcal{G}_2$ game. The utilities presented above are the utilities at the leaf nodes of the game (as represented by U based on the formalization in Sec. 3.4.2).

3.6 Solution concepts in hierarchical games

A key element that influences solution concepts in games is the manner in which each agent reasons over the strategies of other agents. In non-strategic behaviour models, agents do not explicitly model other agents in the game and respond solely on the basis of their own utility structure. Wright and Leyton-Brown [237] refer to this property of behaviour being dependent only on the agent's own payoff as *dominance responsive*. Along with following the property of dominance responsiveness, strategic agents, on the other hand, also reason over the strategies of other agents. This means that strategic agents are not only responsive to their own utilities, but also demonstrate a behaviour that is dependent on others' utilities too; a property that is referred to as *other responsiveness* [237].

The first category of behaviour models we consider is the Quantal level-k (Qlk) model [235]. Qlk models the population of agents as a mix of strategic and non-strategic agents, with agents having a bounded iterated cognitive hierarchy of reasoning. Strategic agents in Qlk use Quantal Best Response (QBR) function, often expressed as a logit response $\pi_i^{\text{QBR}}(a_i, s_{-i}, \lambda) =$

$\frac{\exp\{[\lambda \cdot u_i(a_i, s_{-i})]\}}{\Sigma_{a_i'} \exp\{[\lambda \cdot u_i(a_i', s_{-i})]\}}$, where s_{-i} represents the pure or mixed strategies of other agents and λ is the *precision* parameter that can account for errors in agent response with respect to utility differences[2]. When $\lambda \to 0$, the mixed response is a uniform random distribution, whereas $\lambda \to \infty$ makes the response equivalent to best response. Level-0 agents are non-strategic (NS) agents who choose their actions uniformly at random, whereas Level-1 agents are strategic (S) agents who believe that the population consists solely of Level-0 agents, and their response is a QBR response to Level-0 agents' actions. In the original Qlk model, level-0 agents follow a mixed uniform distribution strategy; however, in [236], the behaviour of level-0 agents is formalised by noting that any behaviour that is *dominance responsive only* can be considered a model of level-0 behaviour. Additionally, the behaviour that is *other responsive*, are level-$k \geqslant 1$ behaviour. In this chapter, I use this expanded definition, and similar to [236], the level-0 agents' strategies follow more intuitive, yet nonstrategic response, such as the maxmax response (MX) or the maxmin (MM) response. Note that for both models, an agent only needs to perform operations on their own utilities, thus adhering to the level-0 constraint of not being *other responsive*. We believe that the expanded definition of the level-0 agents suits our situation much better, since it is unrealistic to expect a driver to choose actions purely at random from their available actions. Even with this expanded definition, these are still *non-strategic* since level-0 agent responses depend purely on their own utilities and do not rely on strategic reasoning over other agents' utilities [237].

Another category of non-strategic behaviour that I consider is rule following behaviour. Under a rule following behaviour level-0 agents strictly adhere to the traffic rules regardless of what the utilities may suggest. Such a strict rule following behaviour is not even *dominance responsive* because based on the utilities constructed in Section 3.5.2, the action that the rule suggests can be strictly dominated by all other actions, and the agent would still follow the rule. On the other hand, if the utilities are constructed in an alternate way that captures the preference of the rule following, then our level-0 agent can be deemed to be dominance responsive. Based on these characteristics, the rule following can be said to be a nonstrategic behaviour, therefore in the category of level-0.

In a hierarchical game, since the agent strategies are factored into levels $s_i = (a_{i,1}, a_{i,2})$, the manner in which an agent reasons over strategies in one level might not be the same as the reasoning process in another level. Therefore, instead of a single solution concept in the game of Fig. 3.2, action level games $\mathcal{G}_2$ can have a different solution concept than the one in the game $\mathcal{G}_1$. In our models, we let agents have a cognitively less demanding non-strategic response in $\mathcal{G}_2$, and a more deliberative strategic response in $\mathcal{G}_1$. This choice is similar to one taken in [70], and

[2] In this formulation, the symbols s_i and a_i are strategies and actions of a game in a general sense, respectively, and not related to the symbols used specifically in the formulation of hierarchical games earlier.

QL0		QL1		PNE-QE		QlkR	
$\mathcal{G}_1$	$\mathcal{G}_2$	$\mathcal{G}_1$	$\mathcal{G}_2$	$\mathcal{G}_1$	$\mathcal{G}_2$	$\mathcal{G}_1$	$\mathcal{G}_2$
NS	NS	S+NS	NS	S	NS	S	NS

Table 3.2: Distribution of strategic (S) and non-strategic (NS) behaviour in action level games $\mathcal{G}_1$ and $\mathcal{G}_2$ in four metamodels QL0, QL1, PNE-QE, and QlkR.

reflects the natural process where it is easier for drivers to reason strategically over the strategy space of discrete manoeuvres than over the space of infinitely many trajectories.

We consider three metamodels of behaviour under Qlk: Ql0, Ql1, and QlkR. We refer to them as metamodels, since they can be further refined based on the choice of response function and sampling schemes to create concrete models (see Table 3.3).

Ql0 metamodel. In QL0 metamodel, we restrict the population to be solely level-0 responders in both $\mathcal{G}_1$ and $\mathcal{G}_2$. Level-0 agents follow non-strategic behaviour with one of two solution concepts, maxmax response (MX), and best worst-case or maxmin response (MM). The model of MX is:

$$a^*_{i,\kappa} = \underset{\forall a_{i,\kappa}, a_{-i,\kappa}}{\operatorname{argmax}}\ u_i(a_{i,\kappa}, a_{-i,\kappa}) \tag{3.3}$$

$$\pi_i(a_{i,\kappa}) = \frac{\exp\Big[\lambda_i \cdot u_i(a_{i,\kappa}, \operatorname{argmax}_{\forall a_{-i,\kappa}} u_i(a_{i,\kappa}, a_{-i,\kappa}))\Big]}{\Sigma_{\forall a_{i,\kappa}} \exp\Big[\lambda_i \cdot u_i(a_{i,\kappa}, \operatorname{argmax}_{\forall a_{-i,\kappa}} u_i(a_{i,\kappa}, a_{-i,\kappa}))\Big]} \tag{3.4}$$

where a^*_i is the pure-strategy utility-maximizing action for i. The model for nonstrategic MM response is:

$$a^*_{i,\kappa} = \underset{\forall a_{i,\kappa}}{\operatorname{argmax}}\ \underset{\forall a_{-i,\kappa}}{\operatorname{argmin}}\ u_i(a_{i,\kappa}, a_{-i,\kappa}) \tag{3.5}$$

$$\pi_i(a_{i,\kappa}) = \frac{\exp\Big[\lambda_i \cdot u_i(a_{i,\kappa}, \operatorname{argmin}_{\forall a_{-i,\kappa}} u_i(a_{i,\kappa}, a_{-i,\kappa}))\Big]}{\Sigma_{\forall a_{i,\kappa}} \exp\Big[\lambda_i \cdot u_i(a_{i,\kappa}, \operatorname{argmin}_{\forall a_{-i,\kappa}} u_i(a_{i,\kappa}, a_{-i,\kappa}))\Big]} \tag{3.6}$$

Equations 3.4 and 3.6 are relaxations that translate the pure strategy action to a noisy response $\pi_i(a_{i,\kappa})$ based on the precision parameter λ_i and sensitivity to i's utility difference with respect to opponent actions that maximizes i's utility for MX and minimizes for MM.

Ql1 metamodel. In QL1, the population consists of a mix of level-0 and level-1 responders in $\mathcal{G}_1$ and level-0 responders in $\mathcal{G}_2$ (Table 3.2). Level-0 agents in this population follow MX and MM nonstrategic behaviour as formulated earlier, and level-1 agents best respond quantaly to level-0

agents' behaviour. With the expanded definition of level-0 agents as nonstrategic bounded rational agents, there is a design choice to be made on what level-1 agents believe about level-0 agents. They can consider level-0 agents to be bounded rational responders having mixed response of Equations 3.4 and 3.6, or level-1 agents can consider level-0 agents to be pure strategy rational responders based on Equations 3.3 and 3.5. We choose the latter to align with the original Qlk model, where agents modelling other agents as bounded rational agents are observed only at a higher cognitive level (level-2 and above). In Qlk models, the mixed population is modelled as a bimodal mixture distribution. Therefore, if the proportion of level-0 and level-1 agents is α and $1-\alpha$, respectively, then the QL1 model response in $\mathcal{G}_1$ is the mixed strategy response.

$$\pi_i^{\text{QL1}}(a_{i,1}) = \alpha \cdot \pi_i^{\text{QL0}}(a_{i,1}) + (1-\alpha) \cdot \pi_i^{\text{QBR}}(a_{i,1}, a^*_{-i,1}, \lambda_i) \tag{3.7}$$

where $\pi_i^{\text{QL0}}(a_{i,1})$ is the left hand side of the equation 3.4 or 3.6 and $a^*_{-i,1}$ is the solution set to equations 3.3 or 3.5 for each of the other agents.

QlkR metamodel. In the QlkR metamodel, the population consists of level-1 agents who believe that everyone else follows a rule following, and the agents in the QlkR model best respond quantaly with precision parameter λ_i. The table of rules that determine the behaviour of rule-following is included in Appendix A.2. The rule following behaviour in this case can be considered as an alternate model of level-0 behaviour. However, the only property that a level-0 model adheres to is dominance responsive. One can argue that such a strict rule following behaviour is not even *dominance responsive* because based on the utilities constructed in Section 3.5.2, the action that the rule suggests can be strictly dominated by all other actions, and the agent would still follow the rule. On the other hand, if the utilities are constructed in an alternate way that captures the preference of the rule following, then our level-0 agent can be deemed to be dominance responsive. Let $\mathcal{R}_{-i}(X_{-i})$ be the action corresponding to the traffic rule that agent $-i$ should follow in state X_{-i}, then the model of QlkR behaviour is as follows.

$$a^*_{i,\kappa} = \underset{\forall a_{i,\kappa}, \mathcal{R}_{-i}(X_{-i})}{\text{argmax}} \; u_i(a_{i,\kappa}, \mathcal{R}_{-i}(X_{-i})) \tag{3.8}$$

$$\pi_i^{\text{QlkR}}(a_{i,\kappa}) = \frac{\exp[\lambda_i \cdot u_i(a_{i,\kappa}, u_i(a_{i,\kappa}, \mathcal{R}_{-i}(X_{-i}))]}{\Sigma_{\forall a_{i,\kappa}} \exp[\lambda_i \cdot u_i(a_{i,\kappa}, \mathcal{R}_{-i}(X_{-i}))]} \tag{3.9}$$

PNE metamodel. The final metamodel we consider is a generalization of pure strategy Nash equilibrium with noisy response. In this metamodel, agents follow a non-strategic model in $\mathcal{G}_2$,

and a strategic model in $\mathcal{G}_1$ as described below.

$$a^*_{i,1} = \underset{\forall a_{i,1}}{\operatorname{argmax}}\, u_i(a_{i,1}, a^*_{-i,1}) \tag{3.10}$$

$$\pi_i^{\text{PNE-QE}}(a_{i,1}) = \frac{\exp\Big[-\lambda_i \cdot \min_{\forall (a^*_{i,1}, a^*_{-i,1})}(u^*_i - u_i(a_{i,1}, a^*_{-i,1}))\Big]}{\Sigma_{\forall a_{i,1}} \exp\Big[-\lambda_i \cdot \min_{\forall (a^*_{i,1}, a^*_{-i,1})}(u^*_i - u_i(a_{i,1}, a^*_{-i,1}))\Big]} \tag{3.11}$$

where $u^*_i = u_i(a^*_{i,1}, a^*_{-i,1})$. In the above model, agents respond according to pure strategy Nash equilibria $a^*_{i,1}$, but in error may choose actions $a_{i,1} \notin a^*_{i,1}$ based on the sensitivity to the difference in the utility of the action and an equilibrium action. We refer to this model as pure strategy Nash equilibria with quantal errors (PNE-QE). The formulation is similar to Quantal Response Equilibrium (QRE), yet with key differences. In QRE, strategic reasoning occurs in a space of mixed responses and the precision parameter is part of common knowledge in the game. In our model, reasoning over opponent strategies is in pure strategy action space and the precision parameter is endogenous to each agent; therefore, when an agent reasons about the strategies of other agents, their parameters do not play a role [55]. Based on the choice of the metamodel, the response function, and the sampling scheme, we get 30 different behaviour models ($\mathcal{B}$), cf. Table 3.3, which we evaluate in the next section.

Estimation of game parameters

The dataset used in this chapter includes $\mathcal{D}$ (˜23k) hierarchical games, instantiated with planning frequency $\Delta t_p = 1s$, and planning horizon $\Delta t_h = 5s$ and with the state variables X_i along with the observed strategy $s^o_i = (a^o_{i,1}, a^o_{i,2})$ for every agent i in the game. For each behaviour model $b \in \mathcal{B}$, we note the errors in actions with respect to the pure strategy responses in the games as $\Delta\mathcal{U}_b = \{\epsilon_{i,b} | \epsilon_{i,b} = \min_{\forall a^*_i}[u_i(a^*_i, a^*_{-i}) - u_i(a^o_i, a^*_{-i})]\}$, where a^*_i are the solutions to Equations 3.3 or 3.5 for non-strategic models, Equation 3.8 for QlkR model, and 3.10 for PNE-QE model (we verified the existence of pure strategy NE for all $\mathcal{G}_1$ games in $\mathcal{D}$). Within the context of a game, we assume that all players follow a common behaviour model, and the precision parameters ($\lambda_{i,b}$) in an individual game is a function of the agent's state X_i (see Tables 3.6 and 3.7 for the list of state factors) from whose perspective the game is initiated as well as the behaviour model b of the game. Therefore, for a given state factor X_i, $\epsilon_{i,b}$, or the error value that captures the utility difference follows an exponential distribution based on the game's precision parameter for Ql0, QlkR and PNE-QE metamodels, and a mixed exponential distribution (3.7) for QL1 in $\mathcal{G}_1$. The exponential distribution of the errors and the (assumed) dependency based on state factors lend well for the model to be fit based on a *generalized linear model* [86]. Additionally, since the mean of an exponential distribution is just the inverse of the distribution parameter, the estimate

that the *glm* model gives is the inverse of the precision parameter estimate that we wish to infer. Therefore, to estimate the value of $\lambda_{i,b}$ we fit a generalized linear model $glm(\epsilon_{i,b} \sim \boldsymbol{\beta} X_i)|_{\Delta\mathcal{U}_b}$ with Gamma($k = 1$) family and inverse link, which models $\epsilon_{i,b}$ as an exponentially distributed random variable with $\mathrm{E}[\epsilon_{i,b}] = \frac{1}{\lambda_{i,b}}$ and $\mathrm{Var}[\epsilon_{i,b}] = \frac{1}{\lambda_{i,b}^2}$. $\boldsymbol{\beta}$ is the model co-efficient, solved through maximum likelihood estimate based on the data in $\Delta\mathcal{U}_b$. The prediction of the glm model gives the mean and standard error of $\lambda_{i,b}^{-1}$ based on the state observation X_i. For the mixed exponential distribution in QL1 model, once we estimate the individual precision parameters of 3.7, we estimate α by maximizing the likelihood function $\Sigma_{\forall a_{i,1}^o} \ln\Big(\pi_i^{\mathrm{QL1}}(a_{i,1}^o)\Big)$.

3.7 Experiment and evaluation

Dataset. I used the intersection dataset of Waterloo Multi-Agent Traffic Dataset, which contains a total of 3649 vehicles and 264 pedestrians, including their centimetre-accurate trajectory estimates. We analyse the decision making in right turning and left turning vehicles, which results in a total of 12526 hierarchical games. The detailed process of data collection, labelling, and description is included in Appendix B. Table 3.4 shows the manoeuvres that are used in the construction of the level-1 games. The manoeuvres are context specific, and I use a rule based method (appendix A.2) to generate the set of available manoeuvres to each agent in the game. The situational context in which each maneuver is available to an agent is shown in the description column of Table 3.4. Relevant code for the experiments is available at https://git.uwaterloo.ca/a9sarkar/traffic_behavior_modeling.

In this experiment I study naturalistic driving behaviour and evaluate which behaviour model captures human driving better, both in terms of model fit and predictive accuracy. I set $\mathbf{W} = \begin{bmatrix}0.25 & 0.5 & 0.25\end{bmatrix}$, thereby giving more importance to pedestrian inhibitory actions and set the value of $d_g = 100$ m. In particular I answer the following research questions:

- *RQ1*. Which solution concept provides the best explanation for the observed naturalistic data?
- *RQ2*. How do state factors influence the precision parameters in the games?
- *RQ3*. How does the choice of the response function in the lower action level game $\mathcal{G}_2$ affect the higher level solutions in $\mathcal{G}_1$?

Table 3.3 shows a synopsis of all the behaviour models included in the evaluation. The model names are indexed by their metamodel followed by the choices of the response functions in $\mathcal{G}_1$:$\mathcal{G}_2$

Model name	**Metamodel**	**Action level game $\mathcal{G}_1$**	**Action level game $\mathcal{G}_2$**	**Trajectory sampling**
PNE-QE:MM_S(1+B)	PNE	Pure strategy NE	Maxmin	Bounds, S(1+B)
PNE-QE:MM_S(1+G)	PNE	Pure strategy NE	Maxmin	Truncated Gaussian, S(1+G)
PNE-QE:MX_S(1+B)	PNE	Pure strategy NE	Maxmax	Bounds, S(1+B)
PNE-QE:MX_S(1+G)	PNE	Pure strategy NE	Maxmax	Truncated Gaussian, S(1+G)
PNE-QE_S(1)	PNE	Pure strategy NE	NA	Prototype trajectory, S(1)
Ql0:MM:MM_S(1+B)	Ql0	Maxmin	Maxmin	Bounds, S(1+B)
Ql0:MM:MM_S(1+G)	Ql0	Maxmin	Maxmin	Truncated Gaussian, S(1+G)
Ql0:MM:MX_S(1+B)	Ql0	Maxmin	Maxmax	Bounds, S(1+B)
Ql0:MM:MX_S(1+G)	Ql0	Maxmin	Maxmax	Truncated Gaussian, S(1+G)
Ql0:MM_S(1)	Ql0	Maxmin	NA	Prototype trajectory, S(1)
Ql0:MX:MM_S(1+B)	Ql0	Maxmax	Maxmin	Bounds, S(1+B)
Ql0:MX:MM_S(1+G)	Ql0	Maxmax	Maxmin	Truncated Gaussian, S(1+G)
Ql0:MX:MX_S(1+B)	Ql0	Maxmax	Maxmax	Bounds, S(1+B)
Ql0:MX:MX_S(1+G)	Ql0	Maxmax	Maxmax	Truncated Gaussian, S(1+G)
Ql0:MX_S(1)	Ql0	Maxmax	NA	Prototype trajectory, S(1)
Ql1:MM:MM_S(1+B)	Ql1	BR to Maxmin	Maxmin	Bounds, S(1+B)
Ql1:MM:MM_S(1+G)	Ql1	BR to Maxmin	Maxmin	Truncated Gaussian, S(1+G)
Ql1:MM:MX_S(1+B)	Ql1	BR to Maxmin	Maxmax	Bounds, S(1+B)
Ql1:MM:MX_S(1+G)	Ql1	BR to Maxmin	Maxmax	Truncated Gaussian, S(1+G)
Ql1:MM_S(1)	Ql1	BR to Maxmin	NA	Prototype trajectory, S(1)
Ql1:MX:MM_S(1+B)	Ql1	BR to Maxmax	Maxmin	Bounds, S(1+B)
Ql1:MX:MM_S(1+G)	Ql1	BR to Maxmax	Maxmin	Truncated Gaussian, S(1+G)
Ql1:MX:MX_S(1+B)	Ql1	BR to Maxmax	Maxmax	Bounds, S(1+B)
Ql1:MX:MX_S(1+G)	Ql1	BR to Maxmax	Maxmax	Truncated Gaussian, S(1+G)
Ql1:MX_S(1)	Ql1	BR to Maxmax	NA	Prototype trajectory, S(1)
QlkR:BR-R:MM_S(1+B)	QlkR	BR to traffic rule	Maxmin	Bounds, S(1+B)
QlkR:BR-R:MM_S(1+G)	QlkR	BR to traffic rule	Maxmin	Truncated Gaussian, S(1+G)
QlkR:BR-R:MX_S(1+B)	QlkR	BR to traffic rule	Maxmax	Bounds, S(1+B)
QlkR:BR-R:MX_S(1+G)	QlkR	BR to traffic rule	Maxmax	Prototype trajectory, S(1)
QlkR:MX_S(1)	QlkR	BR to traffic rule	NA	Prototype trajectory, S(1)

Table 3.3: Synopsis of the thirty behaviour models included in the evaluation. 'BR' stands for Best response.

Level-1 action (maneuver)	Description
`wait-for-oncoming (aggressive)` `wait-for-oncoming (normal)`	Applies to left and right turning vehicles. The action of waiting for a vehicle that has the right of way. Generates a trajectory with terminal velocity of zero.
`proceed-turn (aggressive)` `proceed-turn (normal)`	Applies to left and right turning vehicles. Action of executing the turn.
`track-speed (aggressive)` `track-speed (normal)`	Applies to straight through vehicles. Trajectory accelerates or decelerates to road speed limit.
`follow-lead (aggressive)` `follow-lead (normal)`	Applies to straight through vehicles vehicles with a lead vehicle. Generates a trajectory with same target velocity as leading vehicle.
`decelerate-to-stop (aggressive)` `decelerate-to-stop (normal)`	Applies to all vehicles. Indicates vehicles coming to a stop on change of traffic light from green to amber/red. Generates a trajectory with terminal velocity of zero.
`wait-for-lead-to-cross (aggressive)` `wait-for-lead-to-cross (normal)`	Applies to left and right turning vehicles with a lead vehicle. Indicates vehicle waiting for a lead vehicle to finish executing its turn. Generates a trajectory with terminal velocity of zero.
`follow-lead-into-intersection (aggressive)` `follow-lead-into-intersection (normal)`	Applies to left and right turning vehicles with a lead vehicle which is yet to execute the turn. Indicates a vehicle following the its vehicle into the intersection while the lead vehicle executes a turn. Generates a trajectory with same target velocity as leading vehicle.
`wait-on-red (aggressive)` `wait-on-red (normal)`	Applies to all vehicles. Indicates vehicles waiting on red light.
`wait-for-pedestrian (aggressive)` `wait-for-pedestrian (normal)`	Applies to left and right turning vehicles. Indicates waiting for a pedestrian to cross a crosswalk.

Table 3.4: Description of actions used in $\mathcal{G}_1$ of the hierarchical game. `aggressive` actions generate trajectories with maximum absolute acceleration/deceleration $\geq 2\,\mathrm{ms}^{-2}$.

followed by the sampling scheme used in $\mathcal{G}_2$. For QlkR metamodels, recall that level-1 agents believe that other agents will follow the traffic rules. Therefore, level-1 agents when solving their own action level game $\mathcal{G}_2$, would only solve the game under the manoeuvre that corresponds to the rule following behaviour on the part of level-0 agents. The MX or MM solution concept noted for QlkR metamodels is the one used to solve the $\mathcal{G}_2$ games. For models using S(1) sampling of trajectories, the response function in $\mathcal{G}_2$ is omitted, since the hierarchical game consists only of

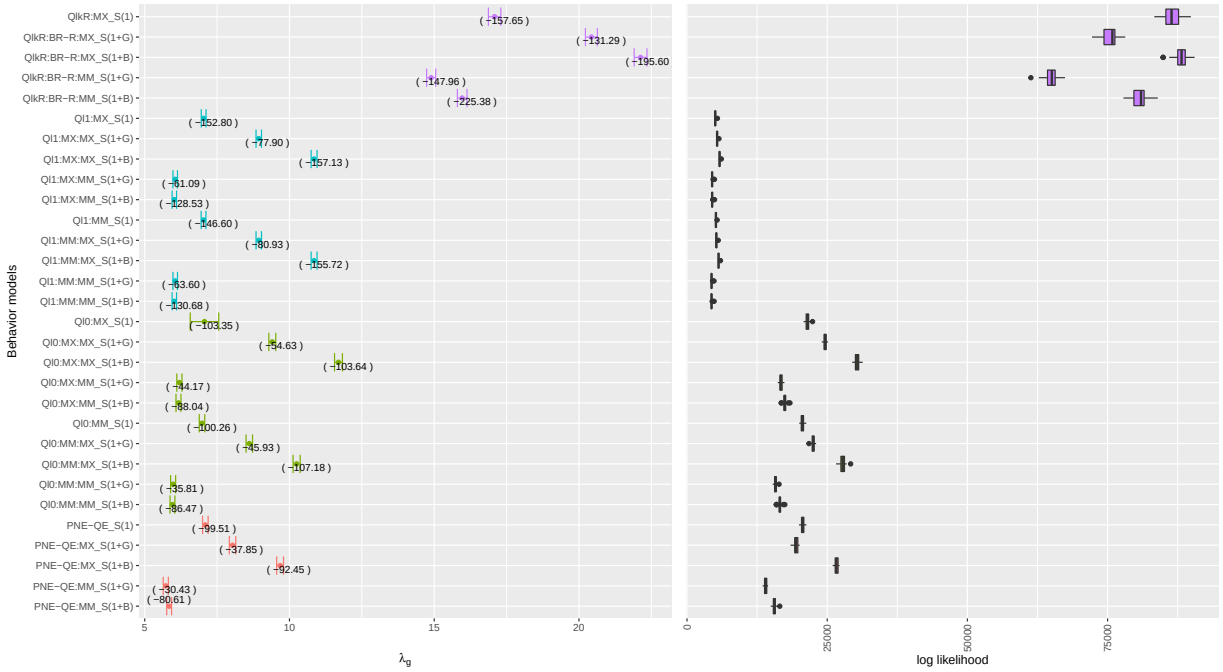


Figure 3.6: Comparison of models based on (a) precision parameters ($\lambda_{i,b}$), fit (AIC values marked in brackets), and (b) predictive accuracy (log likelihood of observations in test data after 30 runs). The first plot shows the mean estimate along with the standard error of the precision parameter across every state, and the second plot shows the boxplot of the likelihood estimates across 30 runs.

$\mathcal{G}_1$ games; and in those cases, each agent has a single choice under each $\mathcal{G}_2$ roots. We perform our analysis of RQs 1 and 2 based on $\mathcal{G}_1$, and discuss the impact of the choice of $\mathcal{G}_2$ solution concepts as part of RQ3. The list of manoeuvres for $\mathcal{G}_1$ is shown in Table 3.4 along with their descriptions. Each manoeuvre is further divided into aggressive and normal modes, thus giving a total of 18 $\mathcal{G}_1$ actions.

RQ1. Which solution concept provides the best explanation for the observed naturalistic data? We address this question in three ways; with respect to (i) parameter values in the model, (ii) predictive accuracy in unseen data, and (iii) model fit. Fig. 3.7 shows the box plot of $\Delta\mathcal{U}_b$ or the utility difference between the true utility (i.e. utility of observed manoeuvre) and the utility of the manoeuvre predicted by the game solutions for each behaviour model. Therefore, a lower value indicates that the solutions are closer to the true manoeuvre executed by the vehicle. For cases where there are multiple solutions, the one with the minimum $\Delta\mathcal{U}_b$ is chosen. The models in the figure are sorted based on the mean $\Delta\mathcal{U}_b$. We see that the QlkR metamodel consistently

$\mathcal{R}(\mathcal{G}_1)$:	MX		MM	
$\mathcal{R}(\mathcal{G}_2)$:	MX	MM	MX	MM
QL0	+5.5	–	+4.5	–
QL1	+4.9	–	+4.9	–

$\mathcal{R}(\mathcal{G}_1)$:	PNE-QE		QlkR	
$\mathcal{R}(\mathcal{G}_2)$:	MX	MM	MX	MM
	+3.9	–	+6.3	–

Table 3.5: Impact of response function choice in $\mathcal{G}_2$ on rationality parameters in $\mathcal{G}_1$. Maxmax (MX) as the solution concept in $\mathcal{G}_2$ lead to higher precision parameter estimates.

shows lower values in the utility difference compared to the Ql0, Ql1, and PNE-QE metamodels. Within the QlkR metamodels, utilities of the actions selected by the QlkR:$\text{MM}_{\text{S(1+B)}}$ model are closest to the utilities of real action selected by the vehicles, however, the difference among the QlkR models is not as distinct.

In general, similar to the results in Fig. 3.7, QlkR models show higher values of the precision parameter as well, thus reflecting better performance as a model of behaviour in level-1 games, i.e. for selection of manoeuvres. QlkR:$\text{MX}_{\text{S(1+B)}}$ (QlkR model with maxmax response in $\mathcal{G}_2$ with bounds sampling) show highest value of the precision parameter, $\lambda = 22.1 \pm 0.22$ (Fig. 3.6a). Next, we evaluate model fit using Akaike information criterion (AIC) values, which are noted in Fig. 3.6(a) in brackets. Since AIC is an evaluation of model fit rather than predictive accuracy, the log likelihood values used in AIC calculation was performed over the entire dataset instead of just the training set. QlkR model with bounds sampling of trajectories have lowest AIC values (-225.3 and -195.6 for QlkR:$\text{MM}_{\text{S(1+B)}}$ and QlkR:$\text{MX}_{\text{S(1+B)}}$ respectively), indicating the best fit among the models based on this criterion.

Alternatively, model selection can also be guided by their predictive power in unseen situations. For evaluation based on this criterion, we use random subsampling with 75:25 training and testing split and 30 runs. The model parameters are estimated based on the observations in the training set, and the predictive accuracy is measured in two ways. First, the predictive accuracy of the models is measured when the solution is in mixed strategies; which is evaluated on the basis of the log likelihood of the observed actions in the testing set. Second, the models are also evaluated based on their accuracy of pure strategy solutions. Fig. 3.6(b) shows the boxplot of sum log likelihood of the observed $\mathcal{G}_1$ actions in the testing set as predicted by each model across 30 runs. The sum log likelihood is calculated based on the likelihood of the observed actions in the test set as predicted by the Quantal Response model with the estimated precision parameter ($\hat{\lambda}$). The set of available actions for the games are often different (since the available actions depend on the state of the road user), and therefore to standardize the analysis process, the likelihood was calculated over the domain of utilities based on a continuous negative exponential distribution rather than over the actions as is often done in estimation of parameter of Quantal Best Response. This process of estimating the likelihood is invariant to different sets of actions, but

still keeps the main model of Quantal Best Response intact. However, note that the likelihood values in this transformed model can be greater than 1. More specifically, the likelihood value of an individual observed action was calculated using the formula $\pi^{\text{QBR}(a^*_{-i},\hat{\lambda_i})}(a^o_i) = \hat{\lambda}e^{-\hat{\lambda}\Delta u}$ where $\Delta u = u_i(a^*_i, a^*_{-i}) - u_i(a^o_i, a^*_{-i})$ is the difference in utility between the game's solution and the selected action, $\hat{\lambda}$ is the estimate of the precision parameter based on the generalized linear model, and the sum log likelihood was calculated with the formula $\sum \log\left(\pi^{\text{QBR}(a^*_{-i},\hat{\lambda_i})}(a^o_i)\right)$.

Next, I further compare the models with respect to their pure strategy solutions. Although the mixed strategy solutions of the models (as expressed through the precision parameter) give a good understanding of how well the models capture naturalistic behaviour, when it comes to using the behaviour models as a behaviour planner in an AV, it is important to evaluate them also with respect to the pure strategy solutions, since an AV can only execute a single action at a time rather than a mixed one. Fig. 3.8 shows the multi-class confusion matrix of the predicted manoeuvres ($\mathcal{G}_1$ solutions) of each model. Instead of all 30 models, I select a cross sectional sample of the five models that are based on S(1) sampling of trajectories. In addition to the models already presented in the chapter, I also include a model solely for the sake of comparison. In this model, labelled 'best response to observed action', as the name suggests, agents simply best respond to the observed manoeuvre of other agents in the game that was played in the previous time step. Based on the data, we see that the accuracy varies significantly across different manoeuvres. The mean accuracy is highest for the rule-based model (78%) followed by QlkR model (75%), which is not surprising since the dataset contains many situations that do not involve strategic reasoning which the strategic behaviour models are good for, for example, approaching the intersection before deciding whether to take the turn. With respect to specific manoeuvres, the behaviour models fare better for the decision of whether to tail a lead vehicle into an intersection or not (follow-lead-into-intersection). On the other hand, the rule based model does better with respect to waiting for oncoming vehicles and pedestrians. In the next chapter, we will revisit this comparison again by focusing on situations that involve higher chance of strategic reasoning as well as some techniques that improve the overall accuracy of the models.

Overall, these results indicate that based on the three evaluation criteria combined (precision parameter, AIC, and predictive performance) and all the models studied, QlkR model where players best respond to the belief that others will follow the traffic rules, especially with bounds sampling of trajectories, is the better model of decision making at the level of manoeuvres for both pure strategy (comparison based on best response) and mixed strategy responses (comparison based on Quantal response).

RQ2. How do state factors influence the precision parameters in the games? In this research question, I study the impact of the state factors on the precision parameter. The state factors are shown in the first column of Tables 3.6 and 3.7. Most state factors are self explanatory;

NEXT_CHANGE refers to the next change in the traffic signal and time in seconds till the change occurs, RELEV_VEHICLE refers to the type of relevant vehicle in the game, for example, whether there is a lead vehicle present or other vehicles in conflict which are not lead vehicles. The table shows the mean precision parameter of the behaviour models for each state factor variable. Since $\lambda_{i,b}$ depends on the state X_i, which is a vector of the five categorical state factors, each row in the table shows the mean precision parameter for situations with the corresponding state factor value, but in isolation; i.e. without taking into account the interaction between the state factors like in the predictive *glm* model. For each state factor value, the value corresponding to the highest precision parameter is underlined. As expected from the previous results, for most state factor variables, QlkR models have the highest precision parameter values. When we compare the values of the precision parameter in Tables 3.6 and 3.7 with the values of Fig. 3.6 (b), we observe that there is much more variation within individual models depending on the agent's state compared to the variation between different models.

RQ3. How does the choice of the response function in the lower action level game $\mathcal{G}_2$ affect the higher level solutions in $\mathcal{G}_1$? As part of this research question, I analyse the impact on the estimate of the precision parameter based on the choice of the solution concept in the lower action level game $\mathcal{G}_2$. For the six possible combinations of the metamodel and the solution concept in $\mathcal{G}_1$, namely, Ql0:MX, Ql0:MM, Ql1:MX, Ql1:MM, PNE-QE, and QlkR, Table 3.5 shows the relative change in the precision parameter of action level games $\mathcal{G}_1$ based on the choice of the response function in $\mathcal{G}_2$. All estimates were found to be significant at $p = 0.05$ based on Dunn's pairwise comparison test after Kruskal-Wallis test indicated significant within group difference. We see that choosing maxmax as the solution concept in $\mathcal{G}_2$ consistently results in a better precision parameter after controlling for the model and the solution concept in $\mathcal{G}_1$.

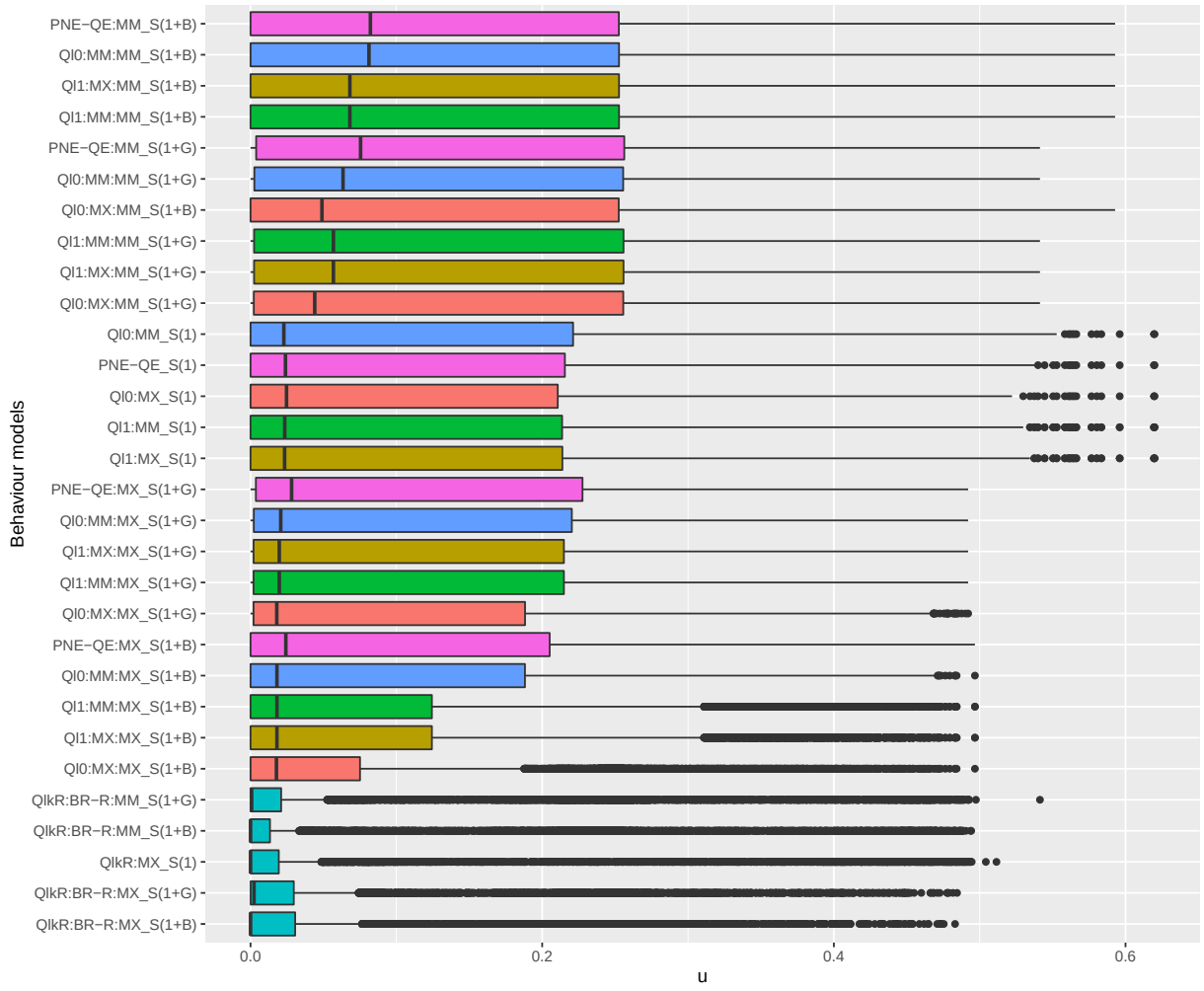


Figure 3.7: Comparison of the models based on spread of utility differences (Δu) between selected action ($a^o_{i,1}$) and the solution ($a^*_{i,1}$) in $\mathcal{G}_1$ games for each behaviour model, sorted by mean Δu.

STATE FACTOR	Ql0: BR_S(1)	PNE-QE: BR_S(1+B)	PNE-QE: BR_S(1+G)	PNE-QE: MM_S(1+B)	PNE-QE: MM_S(1+G)	PNE-QE_S(1)	Ql0: BR: BR_S(1+B)	Ql0: BR: BR_S(1+G)	Ql0: BR: MM_S(1+B)	Ql0: BR: MM_S(1+G)	Ql0: MM: BR_S(1+B)	Ql0: MM: BR_S(1+G)	Ql0: MM: MM_S(1+B)	Ql0: MM: MM_S(1+G)	Ql0: MM_S(1)
SEGMENT															
exec-left-turn	9.61	11.31	10.03	7.05	7.60	9.54	13.22	11.31	7.56	7.82	12.37	10.77	7.32	7.62	9.16
exec-right-turn	10.57	9.19	9.31	6.73	7.38	10.64	32.04	28.85	16.34	15.88	10.26	9.87	7.08	7.05	10.53
OTHER LANES	9.07	12.20	10.03	7.50	7.31	9.02	14.12	11.56	7.72	7.86	12.72	10.94	7.55	7.71	9.04
prep-left-turn	5.29	10.33	8.84	4.25	4.76	5.43	10.93	8.88	4.11	4.45	10.57	8.61	4.07	4.41	5.28
prep-right-turn	9.69	9.82	8.37	5.29	5.44	9.20	21.37	15.04	6.73	6.99	10.79	8.64	5.40	5.47	9.17
NEXT CHANGE															
<10-G	26.68	28.29	20.29	18.77	15.31	25.25	30.78	24.15	19.09	17.75	29.73	23.53	18.95	17.50	24.46
<10-Y/R	5.81	10.86	9.70	5.10	5.46	5.87	14.41	11.12	5.29	5.44	11.61	9.71	5.02	5.22	5.75
>EQ 10-G	16.04	21.68	16.54	16.61	14.61	16.17	23.07	19.42	17.22	16.46	22.11	18.98	16.85	16.07	15.97
>EQ 10-Y/R	5.61	8.03	7.00	4.29	4.64	5.56	9.99	8.08	4.52	4.81	8.52	7.22	4.31	4.57	5.56
SPEED															
HIGH	3.58	6.79	6.74	6.85	6.77	3.58	6.79	6.74	6.85	6.79	6.79	6.75	6.85	6.79	3.58
LOW	8.06	11.47	9.57	6.48	6.59	8.05	13.64	11.17	6.76	7.01	12.05	10.26	6.50	6.73	8.02
MEDIUM	23.08	13.83	13.35	9.81	10.90	21.96	20.22	16.57	11.62	12.16	14.80	13.40	10.05	10.73	19.55
PEDESTRIAN															
N	8.30	12.42	10.43	6.89	7.10	8.35	14.18	11.90	7.18	7.52	12.98	11.16	6.92	7.24	8.33
Y	8.38	10.66	8.93	6.25	6.32	8.26	13.52	10.76	6.62	6.79	11.26	9.53	6.29	6.46	8.18
NO. RELEV. VEHICLE															
$\leqslant 2$	9.56	8.80	8.70	6.74	7.09	8.97	11.91	10.19	7.95	7.59	10.46	9.34	7.53	7.23	9.45
> 2	4.58	20.21	21.07	6.00	3.40	6.47	32.81	24.70	3.40	3.23	29.75	22.15	3.37	3.19	4.54
$= 0$	19.74	16.77	13.94	7.60	8.12	18.98	**22.96**	20.56	15.44	16.03	18.99	17.84	14.00	14.61	19.56

Table 3.6: Mean precision parameter ($\lambda_{i,b}$) of the behaviour models for each state variable across all games.

STATE FACTOR	Ql1: BR: BR_S(1+B)	Ql1: BR: BR_S(1+G)	Ql1: BR: MM_S(1+B)	Ql1: BR: MM_S(1+G)	Ql1: BR_S(1)	Ql1: MM: BR_S(1+B)	Ql1: MM: BR_S(1+G)	Ql1: MM: MM_S(1+B)	Ql1: MM: MM_S(1+G)	Ql1: MM_S(1)	QlkR: BR-R: BR_S(1+B)	QlkR: BR-R: BR_S(1+G)	QlkR: BR-R: MM_S(1+B)	QlkR: BR-R: MM_S(1+G)	QlkR: BR_S(1)
SEGMENT															
exec-left-turn	12.66	10.93	7.39	7.67	9.56	12.66	10.93	7.39	7.67	9.56	15.43	13.43	11.70	10.49	17.83
exec-right-turn	15.47	13.97	9.00	8.84	10.67	15.47	13.97	9.00	8.84	10.68	**22.67**	21.20	13.40	12.69	21.67
OTHER LANES	13.30	11.21	7.61	7.77	9.10	13.30	11.21	7.61	7.77	9.09	31.16	**29.36**	24.03	22.45	22.44
prep-left-turn	10.76	8.72	4.08	4.42	5.27	10.76	8.72	4.08	4.42	5.27	**20.06**	19.13	14.48	14.17	14.84
prep-right-turn	14.07	10.64	5.72	5.87	9.41	14.07	10.64	5.72	5.87	9.41	23.85	22.60	13.53	13.03	**25.08**
NEXT CHANGE															
<10-G	30.06	23.96	19.02	17.57	26.19	30.06	23.96	19.02	17.57	26.17	103.14	87.35	81.60	67.60	**93.19**
<10-Y/R	12.77	10.24	5.08	5.29	5.70	12.77	10.24	5.08	5.29	5.70	**24.59**	21.73	13.75	12.86	12.24
>EQ 10-G	22.51	19.18	17.09	16.23	16.27	22.51	19.18	17.09	16.23	16.21	**76.81**	69.05	73.47	62.97	62.67
>EQ 10-Y/R	9.13	7.57	4.38	4.65	5.60	9.13	7.57	4.38	4.65	5.60	15.30	**14.52**	11.28	10.78	12.22
SPEED															
HIGH	6.79	6.74	6.85	6.77	3.58	6.79	6.74	6.85	6.77	3.58	9.31	**9.43**	6.03	7.35	4.34
LOW	12.72	10.64	6.58	6.83	8.07	12.72	10.64	6.58	6.83	8.07	**25.42**	23.70	18.61	17.50	20.42
MEDIUM	16.77	14.63	10.59	11.20	22.28	16.77	14.63	10.59	11.20	22.28	28.99	24.97	26.33	22.19	**31.33**
PEDESTRIAN															
N	13.50	11.46	7.01	7.33	8.34	13.50	11.46	7.01	7.33	8.34	**27.87**	25.47	19.67	18.44	20.63
Y	12.18	10.04	6.38	6.57	8.33	12.18	10.04	6.38	6.57	8.33	23.02	**21.77**	17.96	16.79	20.59
NO. RELEV. VEHICLE															
$\leqslant 2$	11.10	9.70	7.67	7.36	9.59	11.10	9.70	7.67	7.36	9.59	**23.27**	21.03	18.88	17.33	20.16
> 2	30.80	23.14	3.38	3.20	4.55	30.80	23.14	3.38	3.20	4.55	48.66	**56.83**	18.90	19.07	15.62
$= 0$	20.16	18.63	14.19	14.94	19.30	20.16	18.63	14.19	14.94	19.13	22.67	21.61	19.17	18.21	**80.47**

Table 3.7: Mean precision parameter ($\lambda_{i,b}$) of the behaviour models for each state variable across all games (continued).

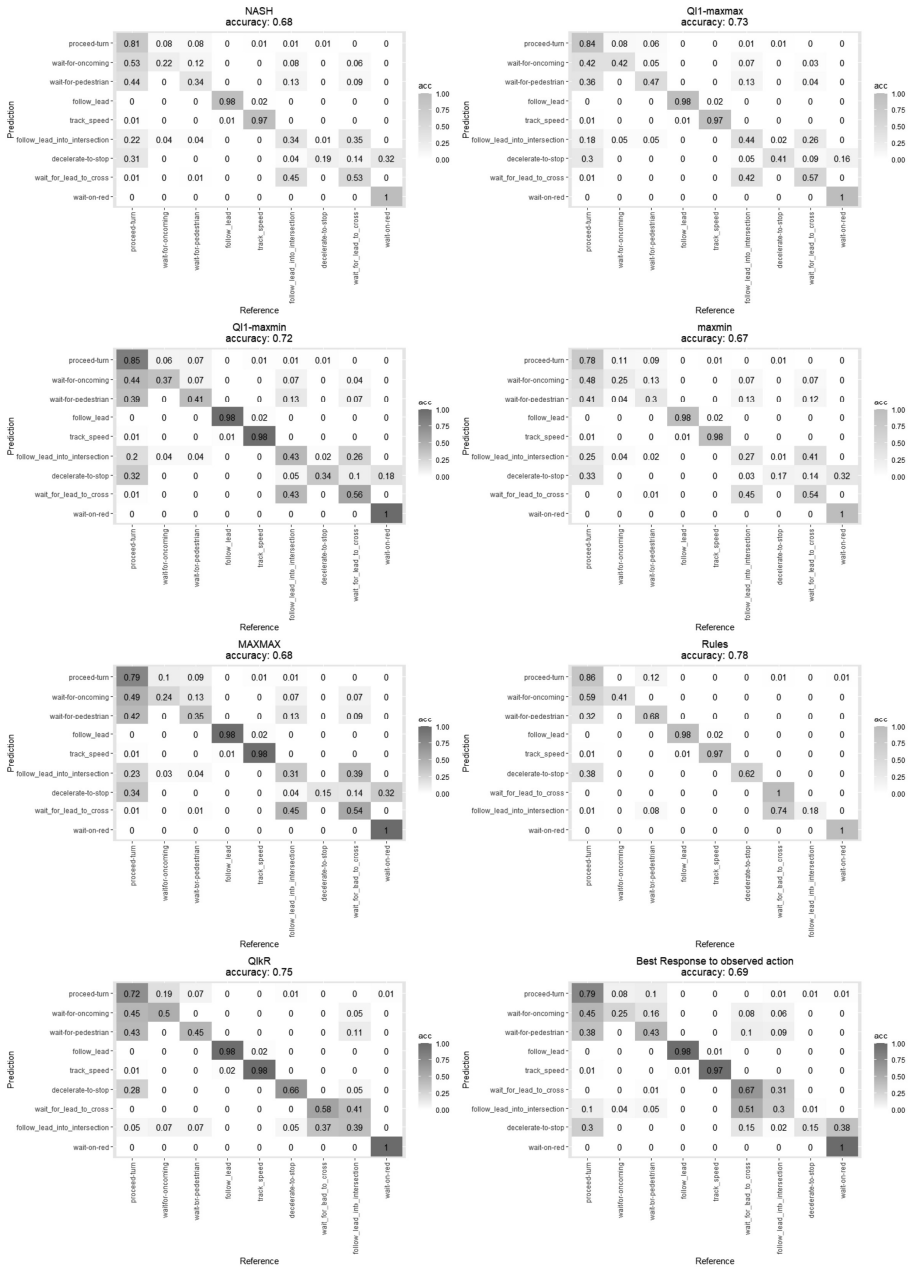


Figure 3.8: Confusion matrix of the pure strategy solutions of level-1 games with S(1) sampling of models with respect to the ground truth maneuver.

3.8 Conclusion

We formalise the concept of a hierarchical game and develop various solution concepts that can be applied to a hierarchical game by adapting popular behavioural game theoretic metamodels (Qlk and PNE-QE). In the context where games are constructed to model naturalistic scenarios, modellers are faced with multiple choices, and this chapter shows different ways in which strategic and non-strategic models can be applied to solve a hierarchical game. We evaluated the behaviour models based on a large dataset of human driving at a busy urban intersection. The results show that among the behaviour models evaluated, modelling driving behaviour as a model where drivers best respond to other drivers with the belief that everyone else will follow the rules is the superior model of manoeuvre selection. As a design choice, constructing the action space of the games with bounds sampling of trajectories provides the best fit to naturalistic driving behaviour. However, if computational efficiency is a concern, then modellers do not lose much performance if they use a single prototype trajectory as a method of constructing actions in a hierarchical game. Furthermore, choosing maxmax as a solution concept for solving the game of trajectories results in higher precision parameter values compared to a maxmin model. The work undertaken in this chapter provides practical insight for practitioners interested in modelling interactive human decision making in traffic for autonomous vehicles.

Chapter 4

Revealed multi-objective utility aggregation

4.1 Introduction

Construction of utilities of agents is one of the main steps involved in the design of game theoretic models based on observational data. During driving, humans balance different potentially conflicting objectives, such as safety, progress, and comfort, in the process of selecting their desired action. The manner in which a driver aggregates these objectives is often context dependent and individual specific. Developing a methodology for estimation of the parameters involved in the aggregation process is a necessary first step towards constructing the agent utilities in a game. One way to estimate the parameters is from naturalistic observational data using the concept of *rationalisability*, that is, the aggregation parameters that would make the observed agent decision optimal. However, the process is nontrivial since not only are there different modalities of aggregation, but also the definition of optimality depends on the various strategic and nonstrategic reasoning models involved. To solve the aforementioned problem, in this chapter I develop a methodology based on the ideas of *revealed preference* to estimate parameters involved in multiobjective aggregation that are specific to the underlying reasoning model and the aggregation method used by the agent.

4.2 Motivation

The general form of a driver behaviour model presented in the problem statement of this thesis is $f(S, U, \mathcal{B}, \epsilon)$, where the driver behaviour is a function of the situational traffic state (S), the utility of the agent (U), the behaviour model ($\mathcal{B}$) and the error model (ϵ) involved in the decision-making process. The focus of the previous chapter was mainly on components S, $\mathcal{B}$, ϵ, where we evaluated different behaviour models based on accuracy, and also analysed the impact of situational state S on error ϵ which was modelled by the precision parameter. The utility U was multi-objective, with safety and progress being the two dimensions, and these utilities were scaled using weighted aggregation using fixed weights. This process of fixing the aggregation parameter to a specific value is commonly encountered in the literature that uses multiobjective utilities [229, 215, 153], especially since it helps focus the game construction and evaluation on models rather than introduce another free variable. However, addressing the question of how drivers aggregate safety and progress in their decision making is a critical question in the context of driving and requires further investigation. To this end, the goal of this chapter is to develop methods that can estimate the aggregation parameters involved in U that are rationalisable based on a given model of reasoning and empirical observation of naturalistic behaviour.

Analysing the aggregation process helps us gain a basic understanding of drivers' preferences under different driving situations, and answering questions such as observed association from the state S and the agent preferences encoded in the utilities U. For example, if a driver is observed to have a higher than expected speed close to an intersection, can we infer something about their aggregation parameter based on that observation — maybe that a driver in that context may weigh progress more than safety? From a modelling point of view, we obtain a more accurate identification of agent preferences, which is especially relevant for behavioural models, since in the absence of such an analysis, any behavioural model, no matter how incorrect the preferences are, can explain away deviations of observed behaviour under the error term ϵ. The line of inquiry of estimating some aspect of agent preferences from observations is related to the problem of inverse reinforcement learning [190, 111], inverse game theory [121], theory of revealed preference [54], and multi-criteria decision making in operations research [61, 102]. In the related work section, I highlight the contribution of this chapter in light of the extensive literature on these topics.

In the context of estimating preferences, it is important to note that there are two separate questions. First, we have the question of form, that is, estimating the form and parameters of the utility function, say $u_{\text{safety}}(\delta) : \mathbf{R}^+ \rightarrow [-1, 1]$ that maps the choices (the distance gap in this example) into a utility interval ([-1,1]). This has been well studied within the literature of revealed preference with specific behavioural theories such as time discounting of utilities [64], risk aversion [65], and prospect theory of loss aversion [230], some of which have been applied

to driving [199] and robotics [122]. However, a second question around multi-objective utilities, i.e. how to estimate aggregation parameters of multi-objective utilities based on consistency of observations and reasoning model, has received comparatively less attention in the context of AV or human driving— and that is the focus of this work. Specifically, I address the following questions.

- **Aggregation**: Given a multi-objective utility U and a parametric scalarization function $S(U, \theta)$, how do we estimate θ that is *rationalisable* with a set of observed choices of all agents conditioned on a model of reasoning?
- **Bounded rationality**: How can the estimation of θ accommodate nonstrategic reasoning models such as maxmax or maxmin?
- **State association**: Is there an observed association between state factors such as velocity, traffic situation, etc., and θ. In other words, are the parameter values stable across different traffic situations?
- **Model performance**: How does the performance of different behaviour models change when utilities U are constructed based on a learning-based technique that infers the aggregation parameter θ from the data?

The question of aggregation is addressed by constructing axiomatic conditions under which a set of observations is *rationalisable* using a given parameterized aggregation method, namely weighted aggregation and satisficing aggregation. As a part of the second question, I show that such a construction is different for strategic and nonstrategic models, where the former can be formulated by a set of linear constraints and the latter as a set of nonlinear constraints. For the third question, I estimate the rationalisable parameters for different traffic situations and evaluate whether there are significant situational differences in how drivers aggregate the utilities, namely safety and progress. Finally, by treating the state factors as independent variables and the aggregation parameter as dependent, I use the data to learn a regression model (CART) that can predict the aggregation parameter in new situations, and use that method to evaluate the performance of the behaviour models developed in the previous chapter based on predictive accuracy of driver manoeuvres.

4.3 Related work

This section spans three different fields of research that deal with similar problems in their own right. Namely, literature on the theory and applications of reveled preference from eco-

nomics, multicriteria decision making literature from operations research, and inverse reinforcement learning from robotics and computer science.

Theory of revealed preferences: The main problem addressed in this chapter, i.e., estimation of agent preferences given a set of observations and a model, falls under the scope of the theory of revealed preferences. Samuelson's [193] classical work on the theory of revealed preference led to an axiomatic characterisation of preferences with the simple observation that all other factors remaining constant, *"if an individual selects batch one over batch two, he [sic] does not at the same time select two over one"*, which was later defined as the Weak Axiom of Revealed Preference[1]. Ironically, although a primary motivation of Samuelson's work was to construct a model of consumer choice behaviour without reference to utility functions, most of recent literature on revealed preference has built upon Afriats's approach [4], which defines axioms of existence of a utility function u that can *rationalise* a set of observed behaviours. Although most of the literature is focused on aggregate consumer demand problems [222, 59], revealed preference conditions can also be constructed for noncooperative strategic models such as Nash equilibrium [46]; and Chambers et al. [42] lay the universality and existence conditions of such a construction for any model beyond just equilibrium. Covering the extensive literature on revealed preference in economics is outside of the focus and scope of this work; therefore, I refer to [59] as a good reference for that general literature.

Most economics models are based on rational choices, and given that my work builds models that include non-strategic behaviour and boundedly rational agents, it is relevant to include literature on revealed preference that is based on behavioural economics. Crawford [54] presents a review of the literature on revealed preference that covers behavioural theories and links to empirical evidence. Dziewulski [64] constructs the revealed preference conditions based on a model in which a single agent uses time-based discounting of their utilities with various discounting models such as quasi-hyperbolic and exponential. In contrast, this chapter uses a model that is simpler in some way (one-shot game as opposed to dynamic game) and complex in other way (multi-agent behaviour). Application of the construction from [64] for the case of driving in dynamic semi-cooperative setting is an interesting future direction of research, especially since discounted utilities are standard in reinforcement learning (RL) based methods, and RL has received a lot of attention from the AV community in recent years. Another behavioural attribute, altruism, has been consistently observed in an empirical setting, especially in the context of dictator games [18]. Andreoni and Miller [9] set the construction of the revealed preference with respect to altruistic behaviour in a dictator game and find that only a quarter of the participants were selfish money maximisers and the rest passed the test of altruistic behaviour. More recently, Porter and Adams [178] study revealed preference with respect to altruistic behaviour in the con-

[1] *Weak* refers to the fact that the statement does not say anything about transitive relations, i.e., a preferred to b and b preferred to c contain no information about the preference relation between a and c.

text of intergenerational wealth transfer, that is, transfer of money from an adult child to ageing parents. Not finding an apparent strategic motive of why one may wish to do that has, unsurprisingly, if I may say so, intrigued some economists. The study in [178] varies different models of utility, from pure selfish behaviour to pure altruism, and finds that although more than 90% of the participants pass the test of revealed preference (i.e., behaviour consistent with the models and utilities), there were differences observed based on whom they were playing the game against, whether parents or strangers. Similarly to [178], I vary the utility construction (different models of aggregation), construct the revealed preference conditions, and test on empirical data ([178] is based on a laboratory experiment) to evaluate what proportion of behaviour passes those conditions. However, the models and applications in this chapter are, of course, quite different. Overall, although the above works have treated different behavioural attributes with respect to theory of revealed preference well, to my understanding there is no existing work on revealed preference that is based on multi-objective utilities and non-strategic reasoning models especially in the context of driving behaviour.

Multi-criteria decision analysis (MCDA) Another strand of literature that is related to this chapter is on multi-criteria decision making from operations research [61]. Compared to theory of revealed preference, where the focus is more on the model of decision making, in MCDA, the focus is on multiobjective nature of the utilities. The process of estimating the parameters of the aggregation process that an agent uses is called preference disaggregation (a terminology I retain in the chapter), and Jacquet-Lagreze and Siskos [102] provide a review of the tools and techniques for that purpose until 2000. From a set of datapoints of ranked choices made by an individual, typical algorithms solve the general minimisation problem $\arg\min_{\mathbf{w}} ||\mathcal{R}(X)^o, \mathcal{R}(X, \mathcal{A}_{\mathbf{w}})||$, where $\mathcal{R}(X)^o$ is the observed ranking of the alternatives by the agent, $\mathcal{R}(X, \mathcal{A}_{\mathbf{w}})$ is the ranking based on the aggregation model $\mathcal{A}$ parameterized on $\mathbf{w}$ on the same set of alternatives S. Standard algorithms, such as UTA [101] formulate the solution as a mathematical programming problem, and in recent years, statistical learning methods similar to those I use (CART) have also been used [61]. There are few differences between the MCDA methods and those in this chapter. First, in my case, the models I study are strategic (and non-strategic) decisions, thereby adding another layer of complexity. Second, our problem in this chapter is also less well-defined, since we do not have access to the drivers' ranking of the preferences, but rather only a singular choice of the observed action. Finally, the preferences in this chapter is taken to be dependent on the situational state.

Inverse reinforcement learning (IRL): Although IRL [163] is conceptually different from the methods presented in this chapter, it is relevant to include some recent works in the literature due to the interest and application of IRL for autonomous driving. IRL formulates the problem of estimating an agents' behaviour as a single agent problem as opposed to the game theoretic approach of treating the problem as one of multi-agent behaviour with support for different rea-

soning processes. Another salient distinction in IRL is that it typically retrieves the utility U that fits the observed behaviour best without referencing utility to prespecified dimensions of safety, progress, comfort, etc., but rather uses a single objective function that may or may not have a semantic meaning. Sadigh et al. [190] use IRL to first learn a policy of behaviour from demonstrations and subsequently use that in a game theoretic based planning module using a level-k (k = 2) type solution concept, although the solution concept is not explicitly stated as such in the paper. As a mathematical formulation, such an approach works well in practise because IRL can provide a best-response type behaviour to the (other) agent action; however, the implicit assumption that the agents adhere to a single model of reasoning throughout every interaction might be a strong one. Nevertheless, the authors show practical ways to integrate a single agent method such as IRL into a game-theoretic setting.

A recent work on learning preference along multiple criteria with a game theoretic view and also in the context of driving is by Bhatia et al. [21], where agent preferences are learnt with respect to a solution concept developed based on the Blackwell approachability theorem [23]. Compared to [21], in this work, I use non-zero sum games and pure strategies in terms of the game constructs, as well as focus on multiple solution concepts. Additionally, in this chapter, I also learn the preferences of drivers based on real-world observational data.

4.4 Aggregation

The general problem of aggregation for an agent is the transformation of a vector valued utility function U_i to a scalar valued function u_i in order to solve the game in question. In other words, this involves the construction of a scalarization function $\mathcal{S}(U_i(a_i, a_{-i}), \theta_i)$ that maps the multi-objective vector of utilities for agent i, $U_i(a_i, a_{-i})$, to the real value utility $u_i(a_i, a_{-i})$ based on the parameter θ_i.

4.4.1 Weighted aggregation

Weighted aggregation is a linear combination of individual utility objectives as follows.

$$\mathcal{S}(U_i(a_i, a_{-i}), \mathbf{w}_i) = \mathbf{w}_i \cdot U_i(a_i, a_{-i}) \tag{4.1}$$

The above equation is simply the dot product between the aggregation parameter ($\mathbf{w}_i$ in this case) and the vector valued utility function. The disaggreagation process involves the estimation of the weight vector $\mathbf{w}_i$ based on the observed actions of the agents in the game.

4.4.2 Satisficing aggregation

A driver always operating at their own subjective tolerance level of risk has been a well established model of behaviour in traffic psychology [233, 77], and has also been empirically validated [133]. Lexicographic thresholding is a method of aggregation that is based on *satisficing* and encapsulates two concepts, namely, ordered criteria of objectives and a thresholding effect [135]. In lexicographic thresholding, an agent ranks the objective criteria based on a fixed and strict total order, for example, safety >progress >comfort. In this work, I focus on safety and progress with a lexicographic ordering of safety >progress. The aggregation of the two utilities into a scalar value is given by

$$\mathcal{S}(U_i(a_i, a_{-i}), \gamma_i) = \begin{cases} u_{s,i}(a_i, a_{-i}), & \text{if } u_{s,i}(a_i, a_{-i}) \leq \gamma_i \\ u_{p,i}(a_i, a_-), & \text{otherwise} \end{cases} \tag{4.2}$$

where γ_i is the *safety aspiration level* of agent i; $u_{s,i}(a_i, a_i)$ is the safety component of the vector valued function U_i; and $u_{p,i}(a_i, a_i)$ is the progress component. Based on the above formulation, an agent evaluates an action of multivalued utility based on progress rather than safety only when the safety utility of that action is greater than γ_i. The disaggregation process for the lexicographic thresholding method involves estimating the parameter γ_i based on the observed action of the agent i in the game.

4.5 Model specific estimation of multiobjective aggregation

The problem we are interested in solving in this section is the estimation of the parameters of the aggregagation for an agent given their observed action in a game. This involves estimating the weight parameters $\mathbf{w}$ for the case of the weighted aggregation method, and the safety aspiration level parameter γ for the case of the satisficing aggregation method. Additionally, since the choice of reasoning model (strategic or non-strategic) influences the behaviour of the agent in a game, we will develop separate methods based on strategic and non-strategic reasoning assumptions.

We start with the following definition of what *rationalisability* means in the context of disaggregation of multiobjective utilities.

Definition 1. *Given a normal form game $\mathcal{G}$, a vector-valued utility U_i, a solution concept $\mathcal{B}$, and a tuple of observed action (a_i^o, a_{-i}^o), an aggregation parameter θ is rationalisable iff (a_i^o, a_{-i}^o) is in the solution set of $\mathcal{G}$ solved with the solution concept $\mathcal{B}$ with scalarized utility $\mathcal{S}(U_i, \theta_i)$.*

		U (a_{-i}^o)		D	
W (a_i^o)		0.5	0.5	0.8	0.8
		0.1	0.8	0.1	-0.5
T		-0.9	-0.9	0.2	0.2
		1	0.8	1	-0.5

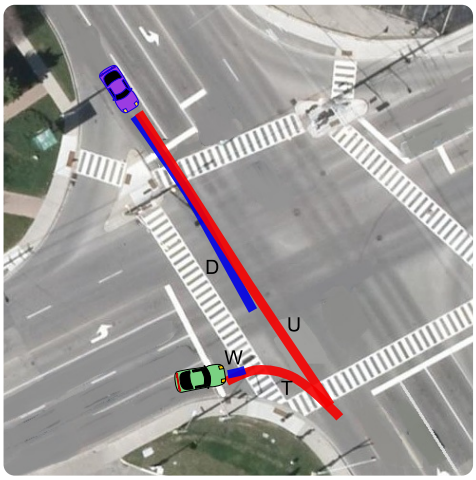


Figure 4.1: An example of right turning scenario with actions *Turn* (T) and *Wait* (W) with oncoming vehicle *Speed up* (U) or *Slow Down* (D). The first row in each cell is safety utility and second row is progress utility.

Based on a dataset of observations (a_i^o, a_{-i}^o) in various game situations, the goal is to estimate the rationalisable θ for each agent in each game situation. We first do this for the case where $\mathcal{S}$ is the weighted aggregation function followed by the case where it is the satisficing aggregation function.

4.5.1 Weighted aggregation

Strategic models

For strategic models, for the observed action, a_i^o, of a strategic agent i to be in the solution set, the action needs to be the best response to the action that i believes $-i$ will play. We use the case of Nash equilibrium in this section where a_i^o and a_{-i}^o are best responses to each other. A running example of a right-turning scenario (Fig. 4.1) elaborates the estimation process. The maneuver level game for the scenario is shown on the right with values derived from the prototype trajectories for each maneuver combination. For each combination of maneuvers, the top row utility values in each cell represent the safety utility and the bottom row represent the progress utility of the right turning vehicle (row player) and straight through (column player) vehicle respectively. Let's say the observed action in this game was (W, U). In that case, for the right turning vehicle, for W to be the best response to the observed action of $-i$, (i.e., U), the necessary and sufficiency conditions are $w_{i,s} \times 0.5 + w_{i,p} \times 0.1 \geq w_{i,s} \times -0.9 + w_{i,p} \times 1$ and $w_{i,s} + w_{i,p} = 1$,

where $w_{i,s}$ and $w_{i,p}$ are agent i's weights for safety and progress utilities respectively. In the general case of arbitrary number of finite discrete actions and $|O|$ is the number of objectives, this can be formulated as a linear program (LP) and the rationalisable weights of agent i can be estimated as the solution to the following LP

$$\begin{aligned}
&\text{maxmize} \sum_{j=1}^{|O|} w_{i,j} u_{i,j}(a_i^o, a_{-i}^o) \\
&\text{subject to} \sum_{j=1}^{|O|} w_{i,j}(u_{i,j}(a_i^o, a_{-i}^o) - u_{i,j}(a_i^{'}, a_{-i}^o)) \geq 0, \quad \forall a_i^{'} \neq a_i^o \\
&\sum_{j=1}^{|O|} w_{i,j} = 1
\end{aligned}$$

In the above LP, we select weights in the feasible set that maximise the utility of the chosen action; however, any combination of weights that fall into the feasible set based on the constraints would be consistent with the conditions of rationalisability.

Non-strategic models

In the case of a non-strategic model, an agent is not *other responsive* but only *dominant responsive* [237]. In other words, since they do not reason about the actions of the other agents, it is not possible to pin down a specific action of the other agent (a_{-i}^o) with respect to which agent i' calculates its best response. This follows from the discussion in Sec. 3.6 in Chapter 3 about the nature of non-strategic agents. To recap, this means that they do not best respond based on a specific belief about other agents' actions, but rather evaluate actions based only on their own utility values, and in our case, choose an action based on an elementary `maxmax` or `maxmin` model. This makes the process of estimating the weights slightly more complicated (read nonlinear) compared to the strategic case. Following from the example of Fig. 4.1, for action W to be the optimal action for agent i (based on the non-strategic model `maxmax`), the maximum utility for agent i that can be realised by choosing W in the aggregate form post scalarization should be greater or equal to the maximum utility that can be realized by choosing T. Therefore, the necessary and sufficiency conditions for the non-strategic case in this example are $\max\{w_{i,s} \times 0.5 + w_{i,p} \times 0.1, w_{i,s} \times 0.8 + w_{i,p} \times 0.1\} \geq \max\{w_{i,s} \times -0.9 + w_{i,p} \times 1, w_{i,s} \times 0.2 + w_{i,p} \times -0.5\}$ and $w_{i,s} + w_{i,p} = 1$. The left term in the inequality gives the maximum realised utility for the action W and the right term is the maximum realised utility for the action T. The process of estimating the weights in the non-strategic case

can therefore be formulated as a nonlinear optimisation problem as follows.

$$
\begin{aligned}
\text{maxmize} \sum_{j=1}^{|O|} & w_{i,j} u_{i,j}(a_i^o, \arg\max_{a_{-i}} w_{i,j} u_{i,j}(a_i^o, a_{-i})) \\
\text{subject to} \sum_{j=1}^{|O|} & w_{i,j}(u_{i,j}(a_i^o, \arg\max_{a_{-i}} w_{i,j} u_{i,j}(a_i^o, a_{-i})) \\
& - u_{i,j}(a_i^{'}, \arg\max_{a_{-i}} w_{i,j} u_{i,j}(a_i^{'}, a_{-i}))) \geq 0, \quad \forall a_i^{'} \neq a_i^o \\
& \sum_{j=1}^{|O|} w_{i,j} = 1
\end{aligned}
$$

Due to the presence of the `argmax` operator, the above problem changes to a nonlinear optimisation problem. Similarly, for `maxmin` non-strategic models, the process of estimating the weights is identical except that the `argmax` operator is replaced by the `argmin` operator. In the latter case, the `argmin` operator gives the minimum realisable utility of the observed action. In our experiments, for both `maxmax` and `maxmin` models, we solve the above optimisation problem using a trust region based method [56].

4.5.2 Satisficing disaggregation

The estimation process for the satisficing method involves estimating the parameter γ_i based on the observed action of the agent i in the game. Similar to the weighted aggregation case, the method of estimation depends on the underlying model due to the assumption an agent has over other agents' behaviour and the subsequent impact on the optimality calculations based on that agent's perspective. However, unlike in the weighted aggregation case, due to the thresholding effect, it is not straightforward to construct a functional form of the scalar utility over which the optimality conditions can be built. Instead, we develop an algorithmic estimation process that helps estimate the complete set of rationalisable values of the aggregation parameter γ.

Strategic model

Based on Eqn. 4.2, the aggregation process for lexicographic thresholding can be expressed in a parametric form as $u_i(a_i, a_{-i}) = \mathcal{S}(U_i(a_i, a_{-i}), \gamma_i)$ where $U_i(a_i, a_{-i}) = [u_{i,s}(a_i, a_{-i}), u_{i,p}(a_i, a_{-i})]$ and $\mathcal{S}$ is the scalarization function of Eqn. 4.2. We present an adapted definition of rationalisability for strategic models as follows:

Algorithm 3: Estimation of Γ_i based on consistency with respect to satisficing aggregation

Algorithm
  Result: Γ_i
  Input: (a_i^o, a_{-i}^o)
  $P \leftarrow$`partition`$([-1, 1], <)$
  $\Gamma_i \leftarrow \{\varnothing\}$
  for $I \in P$ **do**
    $\gamma \leftarrow$`sample`(I)
    if *`is_rationalisable`*(γ) **then**
      $\Gamma_i \leftarrow \Gamma_i \cup I$
    end
  end

Definition 2. *For any agent i, a safety aspiration level $\gamma_i \in [-1, 1]$ is equilibrium rationalisable with strategy profile (a_i^o, a_{-i}^o) iff $\mathcal{S}(U_i(a_i^o, a_{-i}^o), \gamma_i) \geqslant \mathcal{S}(U_i(a_i^{'}, a_{-i}^o), \gamma_i)\ \forall a_i^{'} \neq a_i^o$*

The above definition follows from the definition 1 with an explicit reference to the condition of optimality of the equilibrium solution, that is, for the safety aspiration level of the agent i to be rationalisable, their observed action a_i^o must be the best response to the action of the other agents a_{-i}^o. Algo. 3 presents the general algorithm to estimate the rationalisable parameter. The intuition behind the algorithm is as follows: the value of the parameter γ_i lies within the utility interval [-1,1]. Let $P = \{I_1, I_2, ..., I_P\}$ be an ordered partition of the interval [-1,1]; the process of constructing the partition depends on the underlying models of reasoning, and is explained later. We sample a single value of $\gamma \in I$ and check if the scalarization $\mathcal{S}$ based on that sampled value is rationalisable with respect to the definition 2. If so, we include the partition I from which γ was sampled in the set of rationalisable parameter set Γ, and the union of these sets is the set of rationalisable γ. Next, we set the condition under which the algorithm will be sound and complete.

Proposition 1. *Algorithm 3 is sound and complete based on a partition P iff $\forall I \in P$, is_rationalisable(γ) $\leftrightarrow$ is_rationalisable(γ') $\forall \gamma, \gamma' \in I$*

Implementing the *is_rationalisable* method based on the definition 2 ensures soundness; this is because the utility maximizing action (which is checked in the condition of definition 2) in response to an equilibrium action means that the said action is in equilibrium, and therefore (correctly) rationalisable. The bidirectional implication condition of proposition 1 ensures that if we sample only a single value γ from an interval $I \in P$ and check for rationalisability, then

any γ' in that interval that was not sampled is also rationalisable. Next, I construct a partition for which the double-implication condition of proposition 1 holds. Given a game, let the partition P_{eq} consist of the ordered safety utility of the agent i's action as follows

$$P_{\text{eq}} = \{[-1, u_{i,s}(a_{i,1}, a^o_{-i})), [u_{i,s}(a_{i,2}, a^o_{-i}), ...), [u_{i,s}(a_{i,|A_i|}, a^o_{-i}), 1]\}$$

where $u_{i,s}(a_{i,1}, a^o_i) \leqslant u_{i,s}(a_{i,2}, a^o_i) \leqslant ... \leqslant u_{i,s}(a_{i,|A_i|}, a^o_i)$ is the ordered sequence of the safety utility values of agent i. An example partition for the game with respect to the row player (right turning vehicle) in response to the action U (the observed action) of the column player of Fig. 4.1 is shown in Fig. 4.2.

Figure 4.2: The partition intervals $I_0, I_1, ..$ of P based on the game of Fig. 4.1.

Theorem 1. *For any interval $I \in P_{eq}$, if $\gamma \in I$ is equilibrium rationalisable, then $\forall \gamma' \in I$, γ' is equilibrium rationalisable. Conversely, if $\gamma \in I$ is not equilibrium rationalisable, then $\forall \gamma' \in I$, γ' is not equilibrium rationalisable.*

The proof is based on the intuition that if the partitions are constructed using ordered safety values of different actions, then for any given threshold that falls between two such utilities continue to impose the same ordering of actions after scalarization since the conditions of Eqn. 4.2 remain unchanged.

Proof. Since the equilibrium rationalisability is based on the condition $\mathcal{S}(U_i(a^o_i, a^o_{-i}), \gamma_i) \geqslant \mathcal{S}(U_i(a^{'}_i, a^o_{-i}), \gamma_i)$, I first show that $\mathcal{S}(U_i(a_i, a_{-i}), \gamma_i) = \mathcal{S}(U_i(a_i, a_{-i}), \gamma'_i)\ \forall \gamma, \gamma' \in I$, i.e., the scalarized value based on any two parameters that fall in the same interval is equal.
Consider any $U_i(a_i, a^o_{-i})$,

Case $u_{s,i}(a_i, a^o_{-i}) \leqslant \min I$: In this case, $u_{s,i}(a_i, a^o_{-i}) \leqslant \gamma,\ \forall \gamma \in I$. Therefore, $\mathcal{S}(U_i(a_i, a_{-i}), \gamma_i) = \mathcal{S}(U_i(a_i, a_{-i}), \gamma'_i)$ since both evaluate to $u_{s,i}(a_i, a^o_{-i})$ based on Eqn. 4.2.

Case $u_{s,i}(a_i, a^o_{-i}) > \min I$: In this case, $u_{s,i}(a_i, a^o_{-i}) \geqslant \sup I$, since for any $u_{s,i}(a_i, a^o_{-i}) \neq 1$, $u_{s,i}(a_i, a^o_{-i}) = \min I$ when $u_{s,i}(a_i, a^o_{-i}) \in I$ based on the construction of P_{eq}. Therefore, $\forall \gamma \in I$, $\gamma < u_{s,i}(a_i, a^o_{-i})$, and $\mathcal{S}(U_i(a_i, a_{-i}), \gamma_i) = \mathcal{S}(U_i(a_i, a_{-i}), \gamma'_i)$ since both evaluates to $u_{p,i}(a_i, a^o_{-i})$

based on Eqn. 4.2
Therefore for any $I \in P_{\text{eq}}$, the condition $\mathcal{S}(U_i(a_i, a_{-i}), \gamma_i) = \mathcal{S}(U_i(a_i, a_{-i}), \gamma_i')$ holds true for all $\gamma, \gamma' \in I$.
By the above equality condition, $\mathcal{S}(U_i(a_i^o, a_{-i}^o), \gamma_i) \geqslant \mathcal{S}(U_i(a_i^{'}, a_{-i}^o), \gamma_i) \leftrightarrow \mathcal{S}(U_i(a_i^o, a_{-i}^o), \gamma_i') \geqslant \mathcal{S}(U_i(a_i^{'}, a_{-i}^o), \gamma_i')$, which establishes the biconditional relationship of the theorem based on the definition of equilibrium rationalisability (Defn. 2). □

Theorem 1 helps significantly reduce the number of consistency checks that we need to perform, since we need to check only one value in each interval in P to determine whether all the values in that interval are rationalisable or not. This keeps the run-time complexity of Algo. 3 linear in the number of actions of the agent in the worst case (that is, $O(|A_i|)$), since the run time depends on the size of the partition P_{eq}, which in turn depends on the number of unique safety utilities, i.e., $|A_i|$ in the worst case.

Non-strategic models

Recall that for non-strategic models, an agent i does not hold a specific belief about the action another agent might play, and therefore, similar to the weighted aggregation case, we cannot pin down a specific action a_{-i}^o in response to which the parameters can be estimated. This leads to a revision of the rationalisability definition of Def. 2 to make it independent of the actions of other agents for the maxmax and maxmin models.

Definition 3. *For any agent i, a safety aspiration level $\gamma_i \in [-1, 1]$ is maxmax rationalisable with action a_i^o iff* $\max_{a_{-i}} \mathcal{S}(U_i(a_i^o, a_{-i}), \gamma_i) \geqslant \max_{a_{-i}} \mathcal{S}(U_i(a_i^{'}, a_{-i}), \gamma_i) \forall a_i^{'} \neq a_i^o$.

Definition 4. *For any agent i, a safety aspiration level $\gamma_i \in [-1, 1]$ is maxmin rationalisable with action a_i^o iff* $\min_{a_{-i}} \mathcal{S}(U_i(a_i^o, a_{-i}), \gamma_i) \geqslant \min_{a_{-i}} \mathcal{S}(U_i(a_i^{'}, a_{-i}), \gamma_i) \forall a_i^{'} \neq a_i^o$.

For the strategic case, we needed to check a_i^o for rationalisability only as a response to a fixed action a_{-i}^o, and therefore it sufficed to construct the partition P_{eq} based only on the safety utilities for all actions that were in response to a_{-i}^o. However, for non-strategic models, the rationalisability of a_i^o involves comparison with all entries of the safety utilities of agent i in the game matrix. Therefore, to apply Prop. 1 for the non-strategic case, the partition points of P need to include all the entries of the table as follows:

$$P_{\text{ns}} = \{[-1, u_{i,s}(a_{i,1}, a_{-i})), [u_{i,s}(a_{i,2}, a_{-i}), u_{i,s}(a_{i,k}, a_{-i})), ..., [u_{i,s}(a_{i,|A_i|}, a_{-i}), 1]\}$$

where $u_{i,s}(a_{i,1}, a_{-i}) \leqslant u_{i,s}(a_{i,2}, a_{-i}) \leqslant ... \leqslant u_{i,s}(a_{i,|A_i|}, a_{-i})$ is the ordered sequence of the safety utility values of agent i, and (with a minor abuse of notation) a_{-i} steps through all the corresponding actions of the other agents based on that ordering. The only difference between P_{eq} and P_{ns} is that P_{ns} is partitioned based on the safety utilities of i in the entire game matrix, whereas P_{eq} was based on the column corresponding to a^o_{-i}. This also has an impact on the runtime of the algorithm, which is $O(|A|^N)$, where N is the number of players in the game and $|A|$ is the number of actions for a player. This value is the same as the size of the game matrix since the partition is constructed from each safety utility value for each agent. The corresponding corollaries of Theorem 1 for the non-strategic case are as follows:

Corollary 1. *For any interval $I \in P_{ns}$, if $\gamma \in I$ is maxmax rationalisable, then $\forall \gamma' \in I$, γ' is maxmax rationalisable. Conversely, if $\gamma \in I$ is not maxmax rationalisable, then $\forall \gamma' \in I$, γ' is not maxmax rationalisable.*

Corollary 2. *For any interval $I \in P_{ns}$, if $\gamma \in I$ is maxmin rationalisable, then $\forall \gamma' \in I$, γ' is maxmin rationalisable. Conversely, if $\gamma \in I$ is not maxmin rationalisable, then $\forall \gamma' \in I$, γ' is not maxmin rationalisable.*

Proof. The proof of the above corollaries is similar to the proof of Theorem 1. Observe that for the partition set P_{ns}, for any agent i, the aggregation of the utilities of i is the same for any pair of $\gamma, \gamma' \in I$. This follows from the equality condition $\mathcal{S}(U_i(a_i, a_{-i}), \gamma_i) = \mathcal{S}(U_i(a_i, a_{-i}), \gamma'_i)$, which holds for the partition P_{ns} in the same way as was established for P_{eq} earlier in Theorem 1. This means that the pairwise comparison between the utilities of actions of i is invariant to the value of $\gamma \in I$, thus establishing the conditions of definitions 3 and 4. □

4.6 Experiments and evaluation

4.6.1 Dataset

The dataset to evaluate the utility aggregation estimation methods presented in this chapter is based on three datasets, intersection, crosswalk and roundabout dataset from the Waterloo Multi-Agent Traffic dataset. Refer to Appendix B for a detailed description of the datasets. This set of datasets includes the additional datasets for the roundabout and crosswalk scenarios that were not available when the work on Chapter 3 was undertaken. Furthermore, I select a subset of scenarios from the intersection dataset with the following rationale. In order to select situations where there is a higher chance of strategic interaction between road users, in Chapter 3, I selected the left-turn and right-turn tasks from the intersection dataset. However, even under those tasks,

there are segments where there is less scope for strategic interactions; such as when a vehicle approaches the intersection and is on the traffic segments prior to the point of turn. Similarly, when the vehicle is on traffic segments in the exit lane after the turn has been executed. In such situations, the action of the vehicle mainly involves following the lead vehicle (if any), and therefore less scope for strategic reasoning compared to the action of having to decide whether to start executing the turn or not. Therefore, in this chapter, I refine the scenarios by excluding such situations. Each game is instantiated from the perspective of a *principal agent*, and any road user who is in conflict with the principal agent (a *relevant agent*) is included as a player in the game. The process of selecting situations from the three datasets is as follows.

- **Intersection.** 2-player games, in which a vehicle about to enter the intersection to make a left or right turn (on the `prep-left-turn` or `prep-right-turn` segments) is a principal agent. A straight through vehicle about to enter an intersection at the same time is the relevant agent. Situations are selected so that there are no other vehicles in conflict with the principal agent other than the relevant agent. These situations are a subset of the situations included in Chapter 3.
- **Crosswalk.** N-player games, in which a vehicle about to navigate the crosswalk (on the `west-entry` or `east-entry` segments) is a principal agent. Pedestrians who are about to enter the crosswalk or on the crosswalk are included as relevant agents.
- **Roundbaout.** N-player games, in which a vehicle about to enter the roundabout (on the `*_feeder2` segments) is a principal agent. Vehicles that are already in the roundabout (three closest inner and outer circle segments, refer to Fig. B.5) or about to enter the roundabout at the same time (two closest feeder segments) are included as relevant agents.

For crosswalk and roundabout scenarios, since the number of relevant road users can be high, a group of road users on the same traffic segment at the moment of game initiation is represented with a single player randomly selected from that group. For these two scenarios, the models are evaluated only on the basis of the actions of the principal agents since the game construction does not correctly capture the behaviour of the representative relevant agents due to the missing agents in the game; the missing agents on account of being on the same segment are more 'relevant' to the chosen representative agent that the principal agent. Table 4.1 shows the scenarios, the number of games, and the traffic segments of the principal agents included in each game. A detailed description of the traffic segments, the methodology for the identification of the relevant agent, and the data set collection process are covered in the Appendix B.

To make the comparison easier across the three different scenarios, the action space of manoeuvre for all agents included only two manoeuvres, *wait* and *proceed*. Under each manoeuvre,

20 trajectories were generated, half of which were aggressive. A trajectory was labelled aggressive for vehicles if the maximum acceleration limit was greater than $2\,\mathrm{ms}^{-2}$), and for pedestrians it was labelled aggressive if the mean walking speed was greater than $1.55\,\mathrm{ms}^{-1}$. Relevant code for the experiments is available at https://git.uwaterloo.ca/a9sarkar/single-shot-hierarchical-games.

Dataset	No. of games	Scenario	Principal agent segments
Intersection	1667	Unprotected right-turn, Left-turn across path	`prep-left-turn`, `prep-right-turn`
Crosswalk	288	Crosswalk navigation	`*_entry`
Roundabout	2441	Roundabout entry	`*_feeder2`

Table 4.1: Details of the datasets and scenarios covered in the evaluation.

4.6.2 Analysis of agent preference parameters

		Intersection		Roundabout		Crosswalk	
		Weighted	Satisficing	Weighted	Satisficing	Weighted	Satisficing
Strategic	Nash	100%	68%	100%	96.9%	100%	100%
Non strategic	maxmax	100%	72.2%	100%	56.8%	100%	66.9%
	maxmin	100%	72.28%	100%	56.87%	100%	64.3%

Table 4.2: Pass rate of estimated preferences for each model, aggregation method, and dataset.

Pass rate

The first point of analysis is the pass rate for each model, that is, the percentage of games in which a rationalisable parameter was found for each model and aggregation method combination (Table 4.2). Based on the estimation procedure, I observe that the chosen action of drivers

can be rationalised by a weighted aggregation parameter in all cases for all models. For satisficing aggregation, the pass rate is sensitive to the specific traffic situation and the choice of the reasoning model. In the roundabout and crosswalk scenarios, rationalisable parameters for strategic models could be estimated for almost all games (96.9% and 100%, respectively), whereas for non-strategic models it could only be found for 56.8% to 66.9% of the games depending on the specific solution concept. There are two possible reasons why weighted aggregation parameters show higher rationalisability. First, at least for non-strategic models, the optimisation method involves an approximate procedure (in the form of the use of trust region based method of [56]), which ends up finding a solution, albeit approximate, more easily than the corresponding exact estimation procedure for satisficing based methods. Second, it might be possible that contrary to models in traffic psychology, drivers indeed use a weighted aggregation methodology when evaluating different conflicting objectives as opposed to a satisficing based procedure. The actual parameter values of the weights shed more light on this aspect, which is discussed next.

Parameter values

Next, I study the values of the rationalisable parameters that were estimated under each model. Figures 4.3 and 4.4 show the violin plots of the safety weight (safety weight of $\mathbf{w}$) and safety aspiration level (γ) for weighted and satisficing aggregation methods, respectively. To study the association between the state factors, the figures are stratified based on the velocity of the agent as well as the scenario. The first observation about the weighted aggregation method is that irrespective of the reasoning model, the distributions of the weight parameters are multimodal with the modes being concentrated towards lower and higher values in most cases. This means that in most cases, with safety and progress as the two objectives, drivers tend to weigh heavily one or the other rather than weighing both together in some mixed proportion at the same time. Intuitively, this makes sense because what the revealed preference estimation *rationalises* are the weights of the two objectives that would make the chosen action of the driver optimal. For example, in a given game, if the driver chose to proceed, then what we find is that evaluating that action only with respect to its, say, progress utility, makes it more optimal than if the driver had evaluated that action based on both safety and progress utility. On the other hand, for the satisficing method of aggregation, the distribution of γ is not multimodal in most cases. Rather, the mean values (shown in red) are concentrated near 0, thereby indicating that the population of drivers is homogeneous with respect of satisficing aggregation regardless of the reasoning model or the game situation.

Subgroup analysis

Next, we study the association between the vehicle speed and the estimated parameters values. The mean values of the parameters are shown in red within the violin plots. Within each dataset and scenario, we perform subgroup analysis based on discretised velocity levels, and significance between groups is noted according to Wilcoxon *t-test* at significance levels $p \leqslant 0.05(*), p \leqslant 0.01(**), p \leqslant 0.001(***), p \leqslant 0.0001(****)$. In general, significant differences in parameter values with respect to velocity levels were found for 72% and 42% of pairwise group comparisons for weighted and satisficing aggregation, respectively, where a group is a combination of scenario and model. This points to the fact that the safety aspiration levels of drivers show more stability at different velocity levels compared to the weight parameters. Additionally, within each scenario and reasoning model, for the cases where there is a significant difference, higher velocities are associated with lower safety weights and higher safety aspiration level for weighted and satisficing aggregation, respectively.

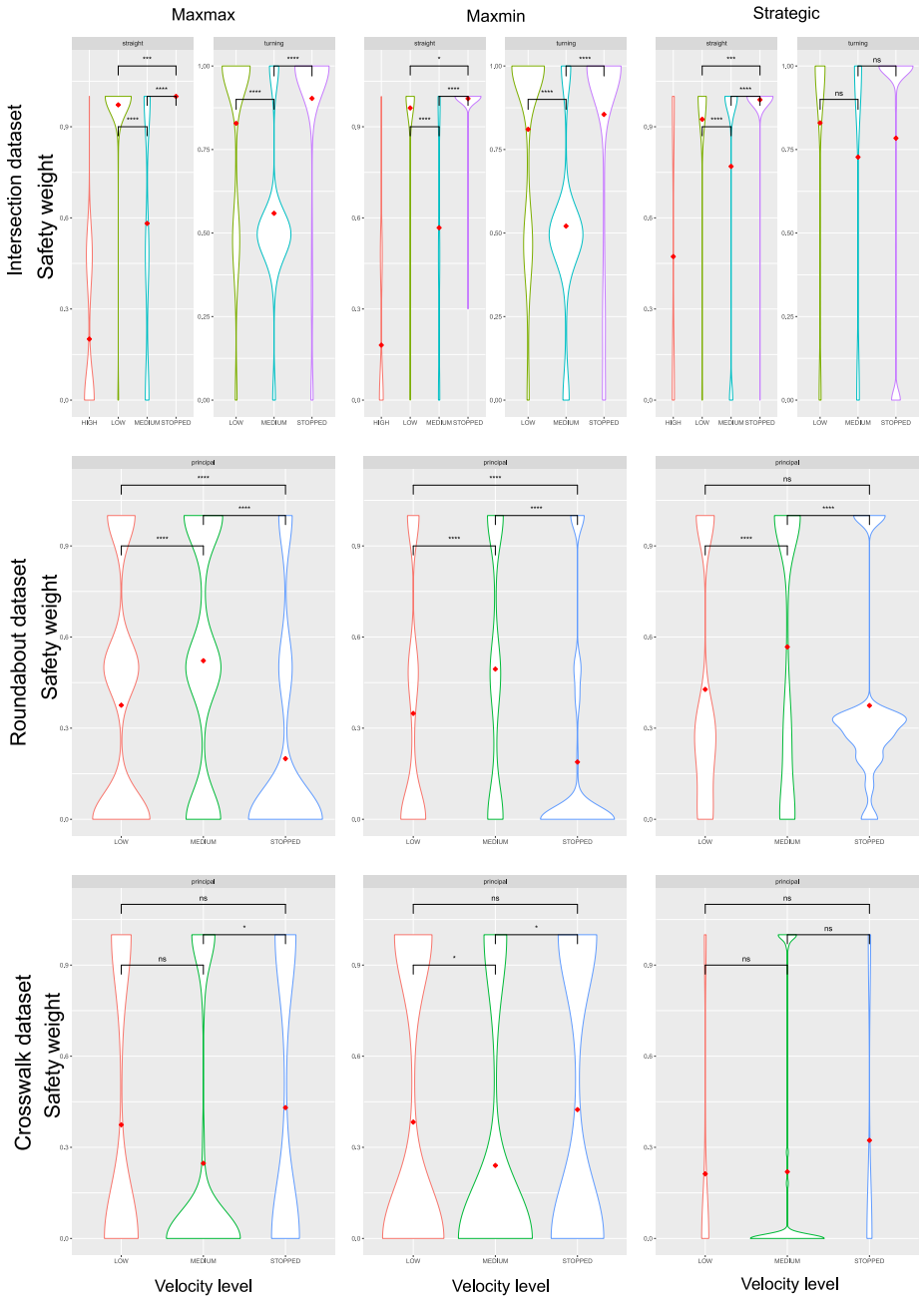


Figure 4.3: Weighted preference disaggregation parameter distribution stratified by vehicle speed, scenario, and task. Significance levels are noted as $p \leqslant 0.05(*), p \leqslant 0.01(**), p \leqslant 0.001(***), p \leqslant 0.0001(****)$, and *ns*.

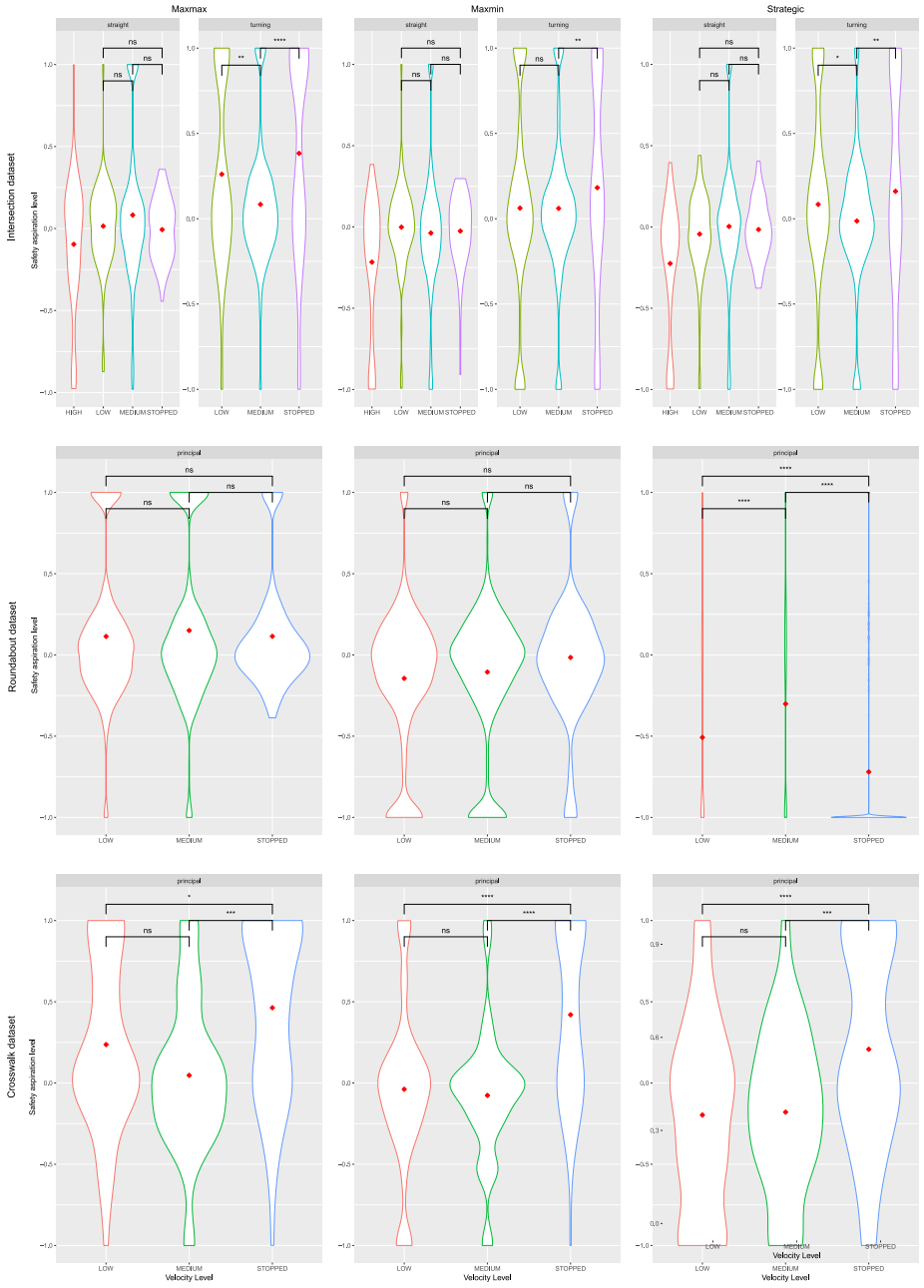


Figure 4.4: Satisficing preference disaggregation parameter distribution stratified by vehicle speed, scenario, and task. Significance levels are noted as $p \leqslant 0.05(*), p \leqslant 0.01(**), p \leqslant 0.001(***), p \leqslant 0.0001(****)$, and *ns*.

4.6.3 Predictive accuracy of rationalisable parameters

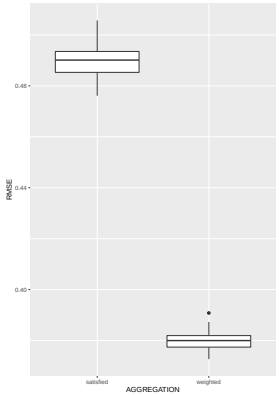


Figure 4.5: Root mean squared error (RMSE) of CART predictive model of aggregation parameters based on state factors.

Based on the subgroup analysis, in many cases, an association is observed between factors such as speed and task with the values of the aggregation parameter. In this section, I evaluate whether a statistical learning method, such as CART, can be used to predict parameters in unseen situations. Since the use of behaviour models as a predictive model in unseen situations involves constructing the utility U of all agents, being able to accurately predict the aggregation parameter for U is the first step in this process. To this end, I use a regression tree (CART) with a feature vector consisting of the driving task, reasoning model, scenario, and velocity, with the parameter value as a dependent variable. I construct two separate regression trees, one for weighted aggregation (which predicts the safety weight in the vector $\mathbf{w}$) and another for satisficing aggregation (which predicts the safety aspiration level, γ, of the agent). The root mean squared error (RMSE) of the predicted parameter values based on 30 runs of random subsampling validation with 80-20 training and testing split is shown in Fig. 4.5. We observe that the model for predicting parameters of weighted aggregation has higher predictive accuracy than the one for satisficing. This is expected from the previous discussion on parameter values and as reflected in Figs. 4.4 and 4.3 that the parameter values for satisficing do not seem to be strongly associated with the velocity level of the vehicle or the chosen feature vectors. This provides evidence that if satisficing is the method of aggregating multiple objectives in the utility, using a predictive model to estimate the parameter might not provide greater benefit, at least in relation to the state attributes (independent variables) considered in this work. The mean RMSE values for the prediction of the weighted aggregation parameter is 0.38. Compare this to the distribution observed in Fig. 4.3. Since the regression tree generates a prediction of single parameter value instead of a distribution, the RMSE

value is in good agreement with the variance observed within a specific mode of Fig. 4.3. This calls for an interesting direction for further work on aggregation methods in which the weight vector is a distribution (as this evidence suggests) rather than a single value.

4.6.4 Evaluation of model accuracy

In this section, I close the loop on the evaluation of behaviour models by using the predictive model of the previous section to first predict the aggregation parameter of the utilities, and use that to reevaluate how the accuracy of the behaviour models from Chapter 3 changes with this more sophisticated methodology of utility construction. When evaluating the models, I consider pure strategy solutions of the models with the same definition of mean accuracy, i.e., the proportion of games in which the model predicted the observed manoeuvre chosen by the driver. I include models from Chapter 3 with S(1+B) method of trajectory sampling. I also include a Stackelberg model for the 2-player games (intersection scenario) for comparison. In the Stackelberg model, the principal agent, on account of not holding the right of way, is modelled as the follower. To match the observed (ground truth) manoeuvre with one of the two manoeuvres in our games, I first select the trajectory generated in the game that is closest to the observed trajectory based on the trajectory length. The manoeuvre corresponding to that closest trajectory is selected to be the ground-truth manoeuvre of the vehicle. To give an estimate of how far off the closest trajectories were compared to the real ones, Fig. 4.6 shows the plot of the absolute difference between the trajectory length of the real trajectory and the closest trajectory sample in the game. We observe that this difference is comparable to the state-of-the-art machine learning based trajectory prediction methods in the literature [142].

Fig. 4.7 shows the mean accuracy of the behaviour models with the model nomenclature similar to the ones in Chapter 3. The new models include a level-2 model (Ql2), which is a model of best-response to level-1 (Ql1) strategies, and a Stackelberg (Stack.) model. Compared to a baseline where I use a weighted aggregation with $\mathbf{w} = [0.5, 0.5]$ (shown with the dashed line in figure Fig. 4.7), a prediction-based aggregation consistently shows better accuracy regardless of whether the choice of the $\mathcal{G}_2$ (trajectory game) solution concept is maxmax (red shapes) or maxmin (blue shapes). The performance of the models with respect to weighted and satisficing aggregation show some dependency on the specific scenario. For intersection and roundabout scenarios, weighted aggregation shows higher accuracy, whereas for crosswalk satisficing based aggregation shows higher accuracy. The crosswalk scenario, which is vehicle-pedestrian interaction type, is quite different compared to the intersection or roundabout where there are only vehicle-vehicle interaction games. Drivers are also much more cautious when navigating a crosswalk, since there are pedestrians involved. Combining the insights from the accuracy result along

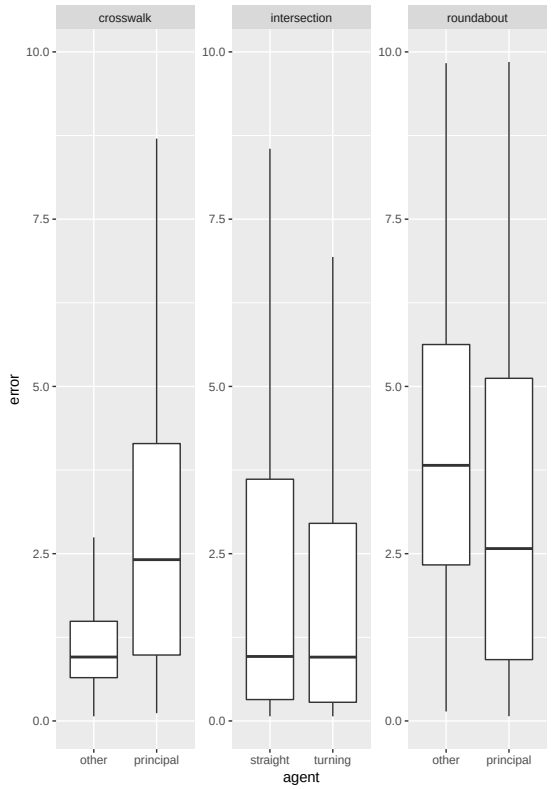


Figure 4.6: Trajectory errors, i.e, the absolute difference of trajectory length in meters between the real trajectory and the generated trajectories in the games.

with the pass rate of Table 4.2, where crosswalk was the only scenario where satisficing aggregation reached the 100% pass rate, the data suggest that satisficing is much more effective as an aggregation method in scenarios where drivers exhibit higher levels of caution, such as crosswalk navigation. The final observation is the worse performance of pure rule following in these scenarios compared to the results in Chapter 3, indicating that a model of pure rule following might not be best suited for the selected situations of high strategic interactions.

Additional analysis of model performance in terms of its false positive rate (model predicts *proceed* when the driver chose to *wait*), and the true positive rate (both predicted and chosen were *proceed*) is shown as receiver operating characteristic (ROC) plots in Fig. 4.8. The dashed line shows the *line of no-discrimination*, i.e., the points in the plot where a random classifier would lie. Each point in the figure represents one of the behaviour models included in the analysis, and different choices of the level-2 game solution concept and aggregation methods are shown with different colours and shapes, respectively. Overall, most models generate predictions

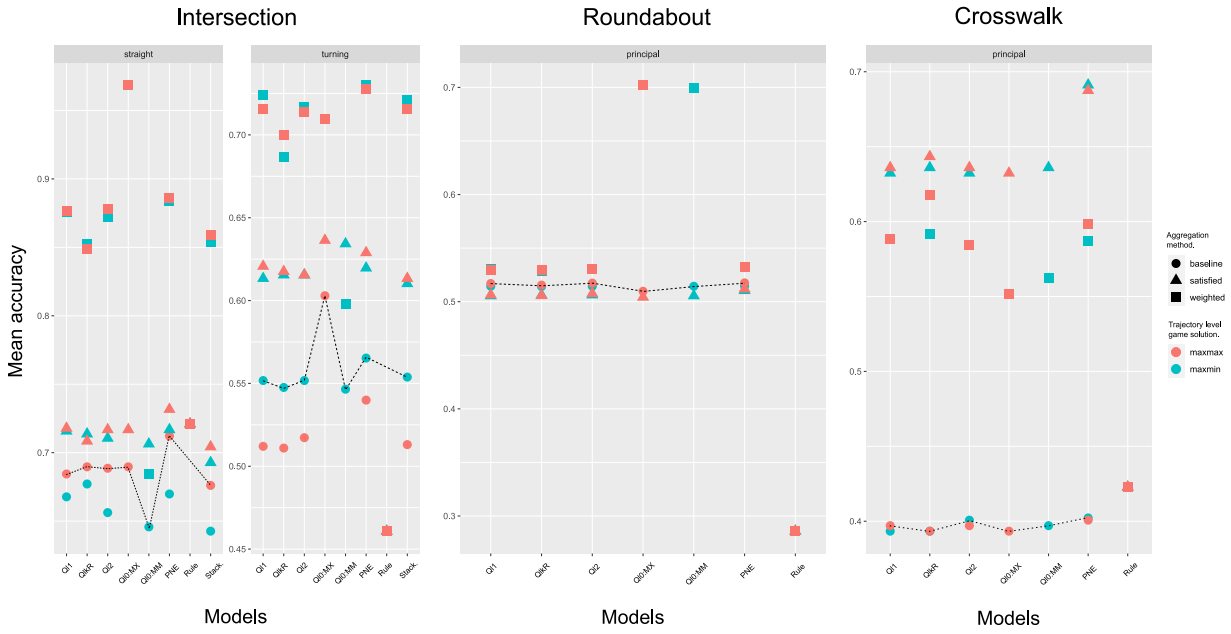


Figure 4.7: Mean accuracy of all the models. The dashed line highlights the accuracy of the models constructed with baseline weights.

that are better than random, with scenario specific effects. For intersection scenario, weighted aggregation performs especially well with the models showing close to perfect classification for the straight-through task (square boxes). This performance is primarily driven by the predictive aggregation model since the baseline model (i.e., non-predictive with $\mathbf{w} = [\mathbf{0.5}, \mathbf{0.5}]$) although being a weighted aggregation based method shows performance that is close to or worse than random (blue and red rounds). This hints at the fact that estimating the aggregation parameters based on state factors, such as velocity, is easier in some scenarios than in others. Similarly, for the crosswalk scenario, we see that the non-predictive aggregation based models do worse than random and the use of a predictive model improves the models' accuracy.

Effect of sampling method

The analysis of the model accuracy presented until now was performed based on S(1+B) or bounds sampling of the trajectories for the level-2 games. This was based on the observation from the previous chapter that bounds sampling shows better predictive accuracy of the models. Given that in this chapter, I presented new methods for calculating the utilities as well as additional datasets, I revisit *RQ. 3* of Chapter 3, i.e., whether sampling more trajectories results in better predictive accuracy compared to S(1) sampling. Specifically, I compare S(1+B), which was

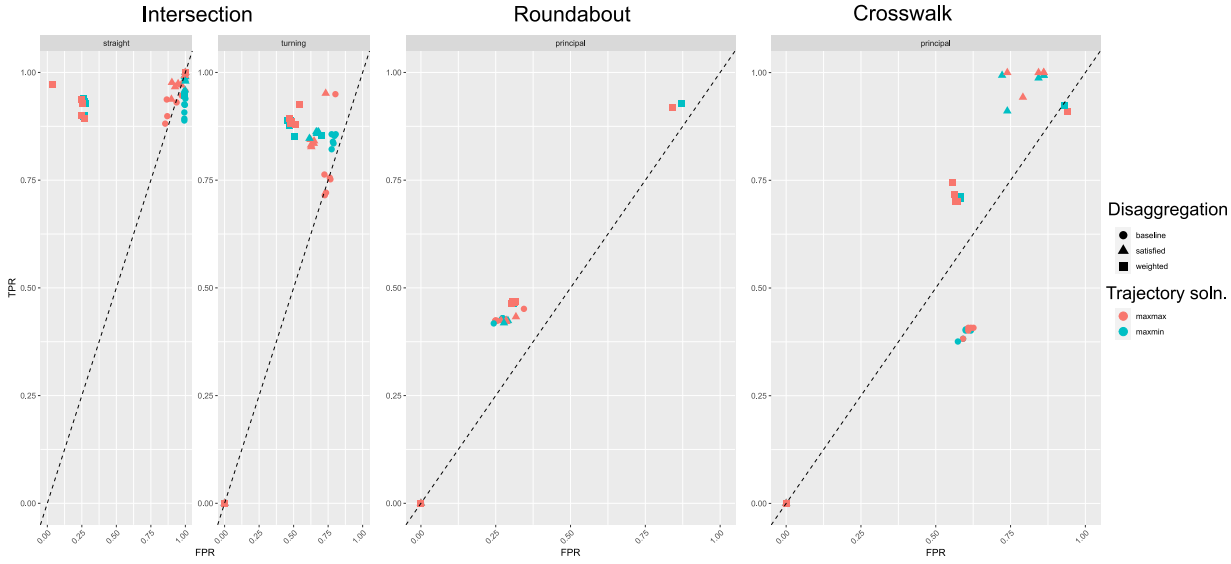


Figure 4.8: Receiver operating characteristics (ROC) of all the models. Dashed line represent the line of no-discrimination.

shown to have the highest predictive accuracy, with S(1). Fig. 4.9 shows the comparison of the accuracy of the models for each scenario with respect to S(1+B) and S(1) sampling. For S(1+B) sampling, the level-2 games are solved using the maxmax solution concept. The level-1 solution concepts are indexed in the figure with different shapes. We see that for straight through vehicles at intersection, there is a consistent drop in accuracy when using the S(1) sampling method, which suggests that regardless of the aggregation process, bounds sampling provides better accuracy than just using a prototype trajectory in the games. This is not observed consistently for other scenarios, where sampling those extra trajectories may not provide as much benefit, and in fact, in some cases such as the crosswalk with satisficing aggregation, we see increased accuracy with only the S(1) sampling. One reason for this difference can be that strategic models generate solutions that include extreme behaviour, thereby reducing the predictive accuracy of the model. This analysis shows that the choice of the sampling method is not independent of the solution concept, and therefore in practise, the game construction (*vis-a-vis* the trajectories sampled) must be carried out keeping in mind the solution concept that would be used to solve the game.

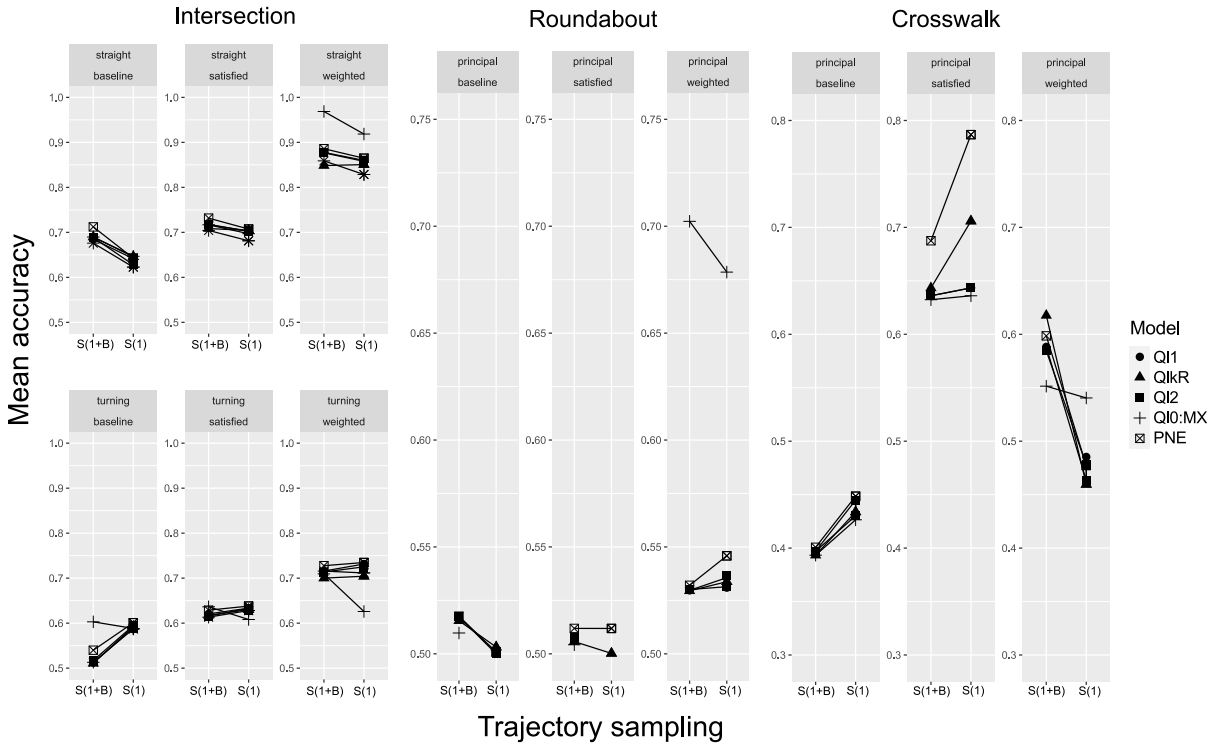


Figure 4.9: Comparison of model accuracy between S(1) and S(1+B) method of action construction with learned weights of utilities. The trajectory level games were solved with maxmax.

4.7 Additional impact on game structures

Different ways of aggregating multi-objective utility mean that for the same traffic situation, depending on the choice of aggregation, the game may have a different utility structure. Consequently, it may be the case that with some methods of aggregation, the drivers have more opportunity to select coordinated actions, i.e., selection of a strategy profile that is mutually beneficial to all the players in the game. In this section, I focus on the aspect of coordination and answer the following question — *are certain methods of multi-objective utility aggregation more likely to lead to the games being one of pure coordination*?

The purpose of this analysis is two-fold. First, it provides insight into the nature of the games that are constructed as a consequence of modelling decisions. Second, since in coordination games, different solution concepts often agree on the same set of strategy profiles as the solution of the game, a designer can focus their efforts elsewhere rather than spending resources on identifying the right model for every situation. The second point is illustrated with the help of an example situation in 4.10(a). As usual, the actions T, U, W, D refer to the manoeuvres *turn,*

speed up, wait, slow down, respectively. In this example, for the left turning vehicle, the action T is strictly dominated by the action W, i.e., no matter what the straight through vehicle does, the action W fetches higher utility for the left turning vehicle. This is reflected in the game's solution based on any of the different behaviour models discussed (Nash, Stackelberg, Level k), most of which agree on (W,U) being the solution. The only exception being the maxmin model which generates the action D for the straight through vehicle – a model that is useful to deal with the worst-case scenario.

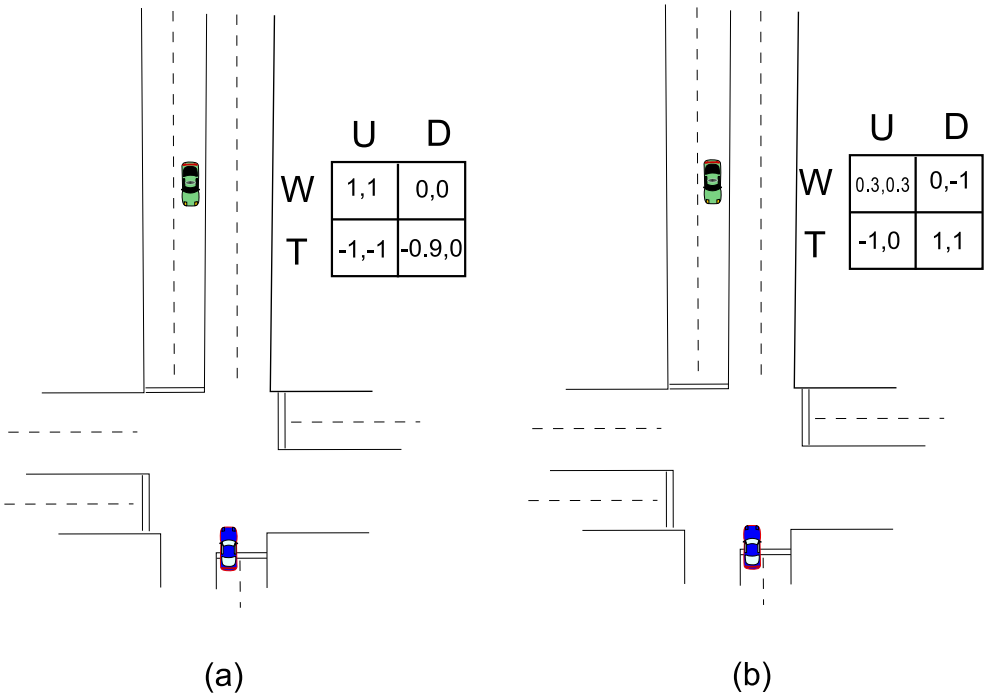


Figure 4.10: An example left turn across path situation illustrating a) pure coordination game with one Nash equilibrium {(W,U)} and b) pure coordination game with multiple Nash equilibria {(W,U), (T,D)}.

4.7.1 Coordination

In contrast to games of conflict where each player has different strategy profiles that they prefer, in coordination games, there is a strong incentive for all players to accept a particular set of outcomes; all of which are often a Nash equilibrium of the game [36]. I focus on games of *pure*

coordination, where there are common strategy profiles that are strictly better than others for all agents.

Definition 5. *A game is one of pure coordination iff* $\forall i$, $\arg\max_{a_i,a_{-i}} u_i(a_i,a_{-i}) = \arg\max_{a_i,a_{-i}} u_{-i}(a_i,a_{-i})$.

In games of pure coordination, other than behaviour models using a maxmin solution concept, all behaviour models converge to the strategy profile that is strictly better than the rest. The two games in Fig. 4.10 are examples of pure coordination, since both straight-through and left-turning vehicles would prefer (W,U) in the first game and (T,D) in the second game than any other outcome. In a two-player game, it is easy to check whether a game is of pure coordination. This can be done simply by checking that the condition of Definition 5 is satisfied for each agent in the game. However, in the case of N-player game, this is nontrivial since the game can be one of pure coordination for some subset of players and not for others. To solve this problem, I use an indicator function that evaluates whether a player in an N-player game can potentially coordinate with another player in the game. This coordination indicator function $I^{\text{CI}}(i)$ for a player i in the game is as follows.

$$I^{\text{CI}}(i) = \begin{cases} 1, & \text{if } \exists -i : \arg\max_{a_i,a_{-i}} u_i(a_i,a_{-i}) = \arg\max_{a_i,a_{-i}} u_{-i}(a_i,a_{-i}) \\ 0, & \text{otherwise} \end{cases} \tag{4.3}$$

The above function evaluates to 1 for a player i, if there exists another player $-i$ such that the utility maximising strategy for both players coincides. A value of 0 indicates that there is no pure coordination potential from the perspective of the player i. Fig. 4.11 shows the mean values of I^{CI} for all the games in each dataset grouped by the aggregation method and the task. Each point in the figure represents a specific method of solving the trajectory level games (shown in different shades) and reasoning model based on which the utilities are aggregated (shown in different shapes). The two methods of aggregation along with the baseline, i.e., weighted aggregation with $\mathbf{w} = [0.5, 0.5]$ is shown on the x-axis. For the two methods of aggregation, the aggregation parameters were predicted based on the CART model developed in the earlier section. If the mean I^{CI} value is 1 for a model, this means that all games in the dataset for that specific method of game construction are of pure coordination, whereas 0 would imply none. Based on the figure, we observe that the satisficing method of aggregation consistently leads to a higher proportion of pure coordination games across all datasets. In fact, for roundabout and crosswalk scenarios, almost all games, when constructed using a satisficing method of aggregation, leads to pure coordination. Second, we observe that the games in the roundabout scenarios have a much higher

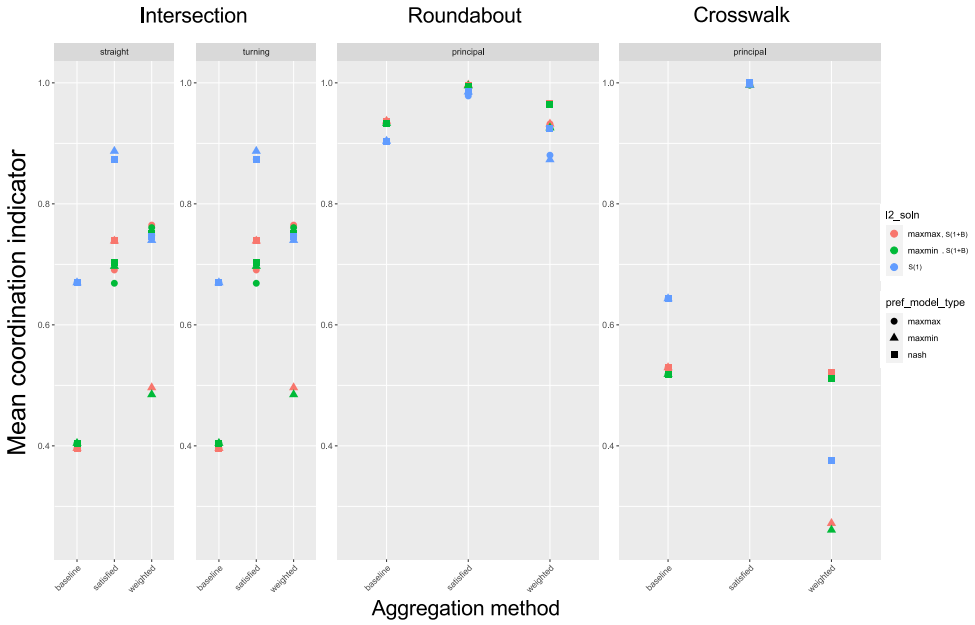


Figure 4.11: Mean coordination indicator value across all the games for each dataset grouped by aggregation method.

proportion of pure coordination games compared to the other scenarios. Other than the intersection scenario, where S(1) sampling of trajectories leads to higher proportion of pure coordination compared to S(1+B), the other two scenarios do not show much impact of model choices in trajectory-level game construction. Finally, we can also connect the above results with the model accuracy results in Fig. 4.7. Since there is a higher proportion of pure coordination games in the roundabout and crosswalk scenarios, there is less distinction between the models in terms of model accuracy in Fig. 4.7.

4.7.2 Risk and payoff dominance

When games are of pure coordination, agents would be better off choosing the utility maximizing strategy profile, and unless there are multiple such profiles, there should be minimal scope of uncertainty in the action selection. Drivers can further reduce uncertainty by communicating their intention through common practises such as hand signalling and other such communication channels. However, if there is no scope of such pre-play communication, as is often the case when interacting with autonomous vehicles, players may have reasons to select alternate strategy profiles. This is especially relevant in games with multiple Nash equilibria, such as the one in the example in Fig. 4.10b. In this example, there are two Nash equilibria, namely, (W,U) and (T,D).

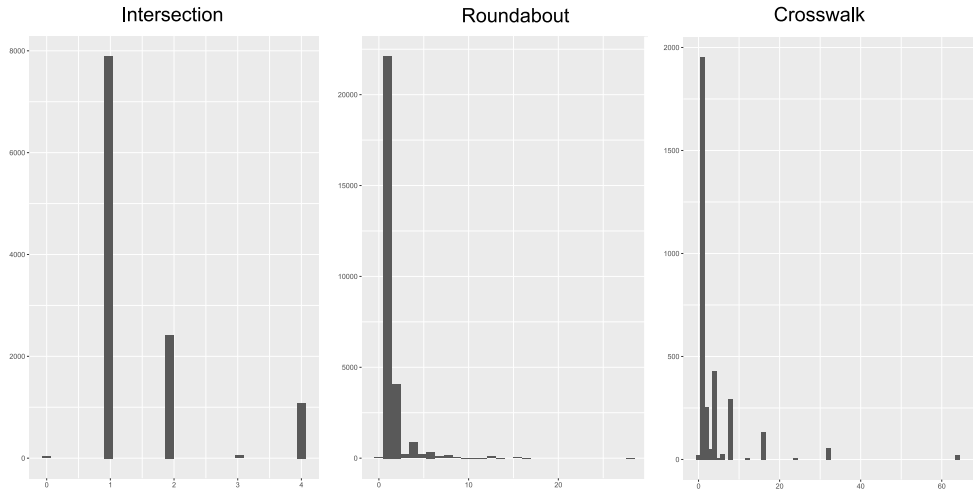


Figure 4.12: Distribution of the number of Nash equilibria across all models for each dataset.

Maxmax and NE solutions agree on (T,D) as the most optimal solution, and the coordination indicator, I^{CI}, is also 1 for both players. However, there is also a certain degree of risk involved from the driver's perspective in selecting (T,D), since if, for some odd reason, the other agent decides to deviate from (T,D), then there is a chance that the agent may end up with the worst possible utility of -1. In other words, although (T,D) is optimal and most models agree upon that, it is also a high stake situation that is contingent upon the other agent doing the right thing. Alternatively, consider the other Nash equilibrium (W,U). Although this strategy profile fetches less utility to both players (0.3,0.3), the stakes are lower than the more optimal (T,D), since in the worst case, even if the other player errs, the player loses 0.7 points instead of 2 as was the case for (T,D). Therefore, for a player who may be worried about things going wrong, (W,U) may be preferable to (T,D). It also happens that (W,U) is the maxmin solution from a nonstrategic perspective. In a general sense, when there are multiple Nash equilibria in a game, players can use different processes to select their preferred NE. The above example demonstrates two common processes, namely payoff dominant (PD) and risk dominant (RD) NE selection. The solution (T,D) is the *payoff dominant* refinement of the Nash equilibrium since this selection is based upon maximising the utility, and (W,U), which is based upon minimising the potential risk from uncertainty about the other player's action choice, is the *risk dominant* refinement. Next, I study whether drivers tend to select risk dominant solutions more often than payoff dominant or vice versa. Additionally, I also answer whether the PD or RD selection changes based on how the games are constructed with respect to the aggregation method and the trajectory level game solution concept.

Since the focus is on games with multiple Nash equilibria, I first filter those games where

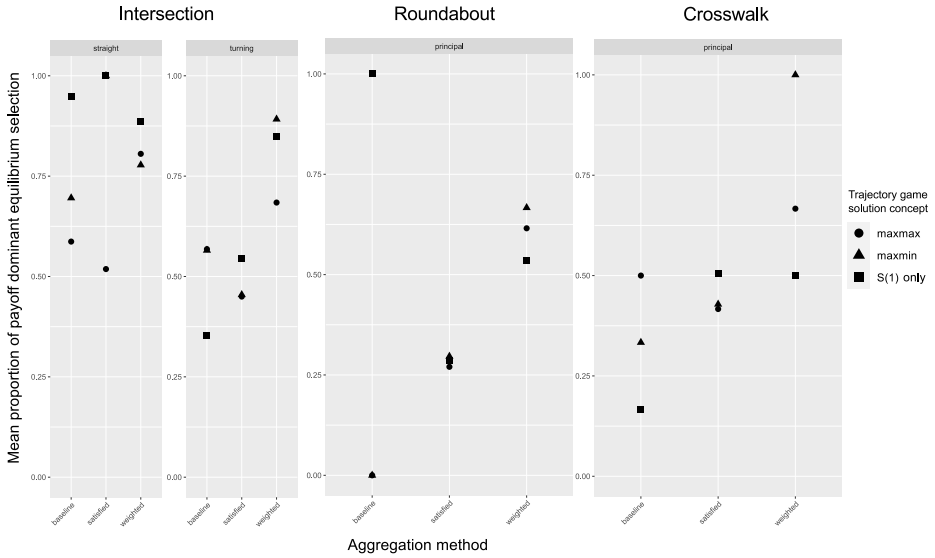


Figure 4.13: Mean proportion of payoff dominant equilibrium selection across all models stratified by scenario, aggregation method, and trajectory level game's solution concept.

there is more than one NE. Fig. 4.12 shows the distribution of the number of Nash equilibria in all games. In most cases, the games had a single Nash equilibrium; however, a significant number of games also had multiple ones. In the subsequent analysis of payoff and risk dominant selection, I include the games with multiple Nash equilibria. Additionally, since the goal is to compare PD and RD selection, I also exclude games in which the PD and RD equilibrium are the same.

Fig. 4.13 shows the proportion of all games in which the selected action was a payoff dominant equilibrium with the games stratified by scenarios, aggregation method (shown on the x-axis) and the solution concept used to solve trajectory level games (shown with different shapes). A value 1 on the y-axis implies that when a selected action by a driver was a NE equilibria of a game, the NE was a payoff dominant one in every game constructed with the model characteristics in question. On the contrary, a value 0 implies that the selected NE was RD in all cases. Based on Fig. 4.13, drivers are more likely to select a PD equilibrium (22 models) compared to a RD equilibrium (12 models). Additionally, when the games are constructed using weighted aggregation, regardless of the scenario or the solution concept chosen for the trajectory games, the drivers always select a payoff dominant equilibrium.

4.8 Conclusion

In this chapter, I address the problem of estimation of multi-objective aggregation parameters of agents based on observed behaviour. The methods developed in the chapter are based on the idea of *rationalisability*, i.e., a value of the aggregation parameter that makes the observed decision of a player optimal conditioned upon a reasoning model. The chapter covers two processes of aggregation, namely weighted and satisficing aggregation, and the reasoning models cover strategic as well as non-strategic models. I show that the process of estimating aggregation parameters for weighted aggregation can be formulated as a linear and non-linear program for strategic and non-strategic models, respectively. Furthermore, I develop a novel algorithm for estimation of aggregation parameters for satisficing aggregation that is linear time for strategic models and polynomial time in the size of actions for non-strategic models. Based on a naturalistic dataset of three different traffic scenarios, rationalisable parameters for weighted aggregation were found for all games in the dataset, and for majority of the games for satisficing aggregation.

The chapter also includes an extensive evaluation of game theoretic models for three different traffic scenarios, intersection, crosswalk and roundabout. The first part of the analysis evaluates the improvement in predictive accuracy of the model when a learning-based method (CART) is used to predict the aggregation parameters of the utilities in the game. Results show that the predictive accuracy improves significantly compared to a weighted aggregation with fixed set of weights. The second part of the analysis studied the process of aggregation with respect to coordination and dominant equilibrium selection. Results from this analysis showed that satisficing method of aggregation is more likely to lead to games being of pure coordination, and players are more likely to select a payoff dominant equilibrium regardless of how the games are constructed.

Chapter 5

Behaviour models for dynamic games

5.1 Introduction

In this chapter, I return to the study of reasoning models. In Chapter 3, the behaviour models developed were in the context of a one-shot hierarchical game. In that construction, the sequence of activity for the players are a repeated sequence of observe (perceiving the state S), solve (solving the game based on the model $\mathcal{B}$), and act (based on the solution $\mathcal{O}$)[1]. Dynamic games adds a fourth activity in this sequence — players updating belief about other players' behaviour. This allows for more sophisticated reasoning in two mutually related ways, communication and elicitation. First, since the belief updating activity is part of the common knowledge, i.e., every player knows that other players update beliefs about each other based on observations, a player can use that to communicate their own intention. Second, based on the response of other agents, a player can also elicit information about other players' behaviour, i.e., other players choosing (or not choosing) the expected action in response to the action of any player may say something about the other player's behaviour. In everyday driving, this is a common process of reasoning as exemplified by the following example. Nudging forward is a commonly observed behaviour in unprotected left and right turns. It communicates intention of a vehicle that it wishes to proceed, and at the same time, the response of other vehicles to being nudged elicits information about the other vehicles' driving attitude; a slowing down to being nudged communicates that the other vehicle may be more considerate than if they didn't yield.

The space of strategies in dynamic games are over the domain of beliefs along with the domain of player actions [213]. As long as the common knowledge [82] assumptions are established, that is, players are aware that everyone is aware of each other's model of behaviour,

[1] Section 1.3 covers details of the notation

solution concepts exist for dynamic games that take into account strategies as well as consistent belief assignment. However, in a naturalistic scenario where players are free to follow heterogeneous models of behaviour, assumptions about the common knowledge end up being too strict for the purpose of modelling naturalistic traffic behaviour. Applying standard models of behavioural game theory is also a challenge, since the addition of beliefs in the strategic solution space presents an additional scope of deviations from the optimal solutions in the game.

The first challenge is of *model uncertainty*, in which the players are uncertain about the reasoning model used by the other players. Whether in the case of Nash equilibiria based models (rational agents assuming everyone else is a rational agent) [201, 85], in Stackelberg equilibrium based models (common understanding of the leader-follower relationship) [70], or level-k model (the level of reasoning of AVs and humans), there has to be a consensus between an AV planner and other road users on the type of reasoning everyone is engaging in. But when there is the possibility of multiple models being followed, the same observation can be explained by different behaviour models, and it is not clear how a player can respond in such a scenario.

The second challenge is the question of *model instability*. Whereas model uncertainty arises from different players using different model of reasoning, model instability deals with the problem of a single player not adhering to a single model of reasoning over time. This problem is especially relevant for level-k models in dynamic games, since the model relies on some players following an elementary non-strategic reasoning (level-0). Therefore, the choice of level-0 model is important, since the behaviour of every agent in the hierarchy depends on the assumption about the behaviour of level-0 agents. The main models proposed for level-0 behaviour include simple obstacle avoidance [216], maxmax, and maxmin models [237, 196]. Although such elementary models may be acceptable when games are *one-shot*, in a dynamic game setting, it is not clear why human drivers, would cognitively bind themselves to such elementary models throughout the play of the game.

In addition to the above challenges, this chapter also demonstrates different ways in which *satisficing* can be used as a concept of game theoretic modeling of traffic behaviour. First, following from the work in Chapter 4, satisficing is used as a method of multi objective aggregation. This application is motivated by models in traffic psychology literature [132]. The second use is the more traditional use of satisficing in economic literature, and it is related to the process of optimization involved in individual decision making [34, 204]. A typical use of satisficing in the latter context is a billionaire businessman being indifferent between a profile of a $1 million and $1.001 million. The same concept can also model a suboptimal choice of actions in traffic where a driver is indifferent between two trajectories that have minimum distance gaps of 10m and 10.01m. Surprisingly, there has been less focus on satisficing compared to Quantal Best Response as a model of suboptimal response in game theoretic models of traffic behaviour. Both uses of satisficing are explored and evaluated in this chapter. Considering that the thesis links

traffic psychology literature and economic literature in several ways, evaluation of the use of satisficing in such models is a natural line of inquiry.

The primary contribution of this chapter is a framework that addresses the aforementioned challenges by unifying modeling of heterogeneous human driving behaviour with strategic planning for AV. In this framework, behaviour models are mapped into three layers of increasing capacity to reason about other agents' behaviour – *non-strategic*, *strategic*, and *robust*. Within each layer, the possibility of different types of behaviour models lends support for a population of heterogeneous behaviour, with a robust layer on top addressing the problem of behaviour planning with relaxed common knowledge assumptions. Standard level-k type and equilibrium models are nested within this framework, and in the context of those models, secondary contributions of the work are a) the use of automata strategies as a model of level-0 behaviour in dynamic games, resulting in behaviour that is rich enough to capture naturalistic human driving (dLk($\mathcal{A}$) model), and b) an interpretable support for bounded rationality based on different modalities of satisficing — *safety* and *manoeuvre*. Finally, the efficacy of the approach is demonstrated with evaluation on two large naturalistic driving datasets as well as simulation of critical traffic scenarios.

5.2 Related work

The models developed in the chapter consider players to have heterogeneous beliefs and behaviour. Heterogeneous agent models (HAMs) have been a mainstay of economic and financial literature where bounded rational behaviour of agents needs to be modeled. Hommes [95] presents an overview of the literature on HAMs in the field of economic and finance and identifies eight reasons behind the popularity of heterogeneous models. Some of the reasons, such as experimental evidence of bounded rational behaviour in humans go beyond just economic behaviour [106]. A common aspect among HAMs is identification of different prototype behaviours of agents. Traditional HAMs, in finance for example, have often used *fundametalists* and *chartists* as the prototype trading behaviours [245]. The former refer to traders who base their decisions on macroeconomic metrics, and the later refer to traders who focus on observed market performance. These prototype models in heterogeneous models are often application specific, and this chapter therefore includes prototype models (accomodating and non-accomodating) that are relevant to driving behaviour.

Driver behaviour heterogeneity has been well established through observational studies, and these heterogeneities are often interpreted as different driving styles [191]. For the task of car following behaviour, Ossen and Hoogendoorn [172] show that the heterogeneity in car following

behaviour can be better explained through heterogeneity in the models itself compared to heterogeneity in the parameters of a single models. This is similar to the construction of the models in this chapter, where the player types within a model represent heterogeneity within each model and different models provide another layer of heterogeneity. Ellison et al. [66] identifies four different factors relevant in the study of driving behaviour heterogeneity — variation in driving behaviour across different road structures, locations, driving tasks, etc. (spatial heterogeneity), variation in different times of the day (temporal heterogeneity), variation in behaviour within a driver (longitudinal heterogeneity), and variation across a population of drivers (cross sectional heterogeneity). After controlling for each of the factors, they show that based on an observational study, there is much more cross-sectional variation in driving behaviour compared to longitudinal variation. Also, there is significant impact of spatial and temporal factors in driving behaviour even within a single driver.

Heterogeneity in behaviour can be interpreted in multiple ways depending on the specific metric used to study the heterogeneity. In general, heterogeneity in driving behaviour can be measured at three different levels. First, as was studied in [66], variation in drivers' choice of speed and acceleration values are an indicator of heterogeneity. These attributes are referred to as driving style measures in [191]. Second, heterogeneity can be measured in terms of the underlying endogenous factors that play a role in speed and acceleration choices. Risk tolerance of the driver has found wide acceptance as a key endogenous factor relevant for modeling [132, 100]. In [159], based on user questionnaires the heterogeneity with respect to risk was captured through four different risk profiles. These include drivers who are unaware of their risky behaviour, drivers who undertake risks in a calculated manner, drivers who take risks only in response to certain situations such as being in hurry, and those who engage in risky behaviour in a compulsive manner. Simulation studies have also identified the heterogeneity in drivers' perception of risk as a function of speed and time gap [134, 133]. The third level at which heterogeneity can be measured is at the level of the behaviour model. For example, in [172] different models of car following behaviour are considered based on how many leading vehicles a follower vehicle takes into account in their decision making. The follow up observational study suggests that the heterogeneity can be explained by different drivers engaging in different reasoning models, whereas some respond only to the immediate leading vehicle, others consider two or three leaders when choosing their actions. In relation to the literature on heterogeneity, the models developed in this chapter can capture heterogeneity in all three levels. At the highest level, different models of behaviour (level-k, MSPE, SSPE, and robust) capture the heterogeneity of models, the agent types as expressed through safety aspiration level capture the heterogeneity of the endogenous attributes such as risk tolerance, and finally, the output of the models show the final heterogeneity in the specific indicators such as speed and acceleration choices.

Another line of research deals with the model instability problem of lower level agents in

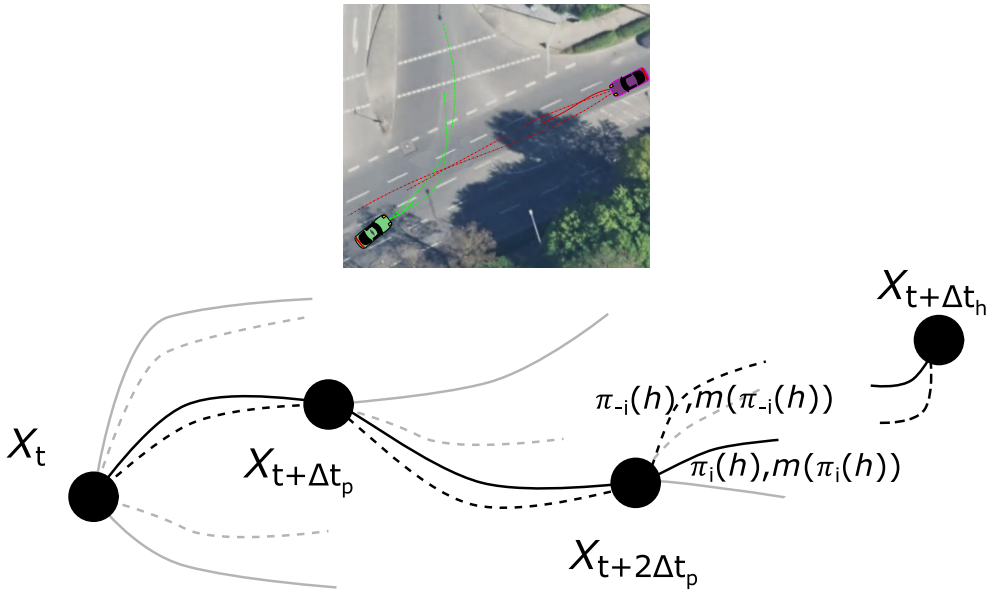


Figure 5.1: Schematic representation of the dynamic game. Each node is embedded in a spatio-temporal lattice and nodes are connected with a cubic spline trajectory.

dynamic games by developing a model of learning based on observations of game play. Ho and Su [94] presents a model of level-k behaviour in dynamic games where agents form beliefs about the levels of other players and update them based on observations of game play, thereby taking care of the learning aspect. Additionally, the agents are also free to choose from any level in the hierarchy that maximizes their payoff. This dynamic assignment of the levels allows agents to not be adversely affected by being restricted to a single level. In this work, since the focus is less on modelling the mechanism of how humans learn to drive, I do not incorporate an explicit learning aspect into the model. Rather, compared to [94], I address model instability by allowing level-0 agents to have a more sophisticated model of level-0 behavior that allows them to switch between the two finite state transducer models.

5.3 Game Tree, Utilities, and Agent Types

The dynamic game is constructed as a sequence of simultaneous move games played starting at time $t = 0$ at a period of Δt_p secs. over a fixed horizon of Δt_h secs. Each vehicle $i \in \{1, 2, .., N\}$'s state at time t is a vector $X_{i,t} = [x, y, v_x, v_y, \dot{v}_x, \dot{v}_y, \theta]$ representing positional co-ordinates (x, y) on R^2, lateral and longitudinal velocity (v_x, v_y) in the body frame, acceleration $(\dot{v}_x, \dot{v}_y)$, and yaw (θ). The nodes of the game tree $X_t = X_{i,t}^N$ are the joint system

states embedded in a spatio-temporal lattice [251], and the actions are cubic spline trajectories [109, 6] generated based on kinematic limits of vehicles with respect to bounds on lateral and longitudinal velocity, acceleration, and jerk [14]. A history h_t of the game consists of a sequence of nodes $X_0..X_t$ traversed by both agents along the game tree until time t. I also use a hierarchical approach in the trajectory generation process [156, 70, 196], where at each node, the trajectories are generated with respect to high-level manoeuvres, namely, *wait* and *proceed* manoeuvres. For wait manoeuvre trajectories, a moving vehicle decelerates (or remains stopped if it is already stopped), and for proceed manoeuvres, a vehicle maintains its moving velocity or accelerates to a range of target speeds. Strategies are presented in the behaviour strategy form, where $\pi_i(h_t) \in \mathrm{T_i(X_t)}$ is a pure strategy response (a trajectory) of an agent i that maps a history h_t to a trajectory in the set $\mathrm{T_i(X_t)}$, which is the set of valid trajectories that can be generated at the node X_t corresponding to both manoeuvres. The associated manoeuvre for a trajectory is represented as $m(\pi_i(h)) \in \{wait, proceed\}$. Depending on the context where the response depends on only the current node instead of the entire history, I use the notation $\pi_i(X_t)$; I also drop the time subscript t on the history when a formulation holds true for all t. Since each history ends in an unique node of the game tree, the overall strategy of the dynamic game is the cross product of behaviour strategies along all histories of the game $\sigma : \pi_1(h) \times \pi_2(h) \times ..\pi_N(h); \forall h$. I use the standard game-theoretic notation of i and $-i$ to refer to an agent and other agents respectively in a game.

The fixed horizon nature of the game construction is motivated by the need to study behaviour at specific interactive situations under static conflict, such as left and right turns at intersections. This is a departure from the usual moving horizon planning technique used in Model Predictive Control (MPC) type planners [37]. However, the framework developed in this chapter can also be extended as a general purpose planner (or for modelling behaviours of dynamic conflict such as lane changes) by changing the construction to a moving horizon setting.

The utilities in the game are formulated as multi-objective utilities consisting of two components — **safety** $u_{s,i}(\pi_i(h), \pi_{-i}(h)) \in [-1, 1]$ (modelled as a sigmoidal function that maps the minimum distance gap between trajectories to a utility interval [-1,1]) and **progress** $u_{p,i}(\pi_i(h), \pi_{-i}(h)) \in [0, 1]; \forall i, -i$ (a function that maps the trajectory length in meters to a utility interval [0,1]). In general, these two are the main utilities (often referred to as inhibitory and excitatory utilities, respectively) upon which different driving styles are built [191]. Agent types $\gamma_i \in \Gamma$ are numeric in the range [-1,1] representing each agent's *safety aspiration level* — a level of safety that an agent is comfortable operating. This construction is motivated by traffic behaviour models such as Risk Monitoring Model [220] and Task Difficulty Homeostasis theory [76], where, based on a traffic situation, drivers continually compare each possible action to their own risk tolerance threshold and make decisions accordingly. To avoid dealing with complexities that arise out of agent types being continuous, for the scope of this chapter, I discretize the types

by increments of 0.5 for the experiments. Based on their type (i.e. their safety aspiration level) how each agent selects specific actions at each node depends on the particular behaviour model, and is elaborated in Sec. 5.4 when I discuss the specifics of each behaviour model.

Unless mentioned otherwise, the utilities (both safety and progress) at a node with associated history h_t are calculated as discounted sum of utilities over the horizon of the game conditioned on the strategy σ, type γ_i and discount factor δ as follows

$$\sum_{k=1}^{L} \delta^k u_i(\pi_i(h_{(k-1)\Delta t_p}), \pi_{-i}(h_{(k-1)\Delta t_p}); \sigma, \gamma_i) + \mathcal{N} u_{i,C}$$

where $L = \lfloor \frac{\Delta t_h}{\Delta t_p} \rfloor$ is the maximum number of planning steps; $u_{i,C}$ is the continuation utility beyond the horizon of the game and is estimated based on agents continuing with the same chosen trajectory as undertaken in the last decision node of the game tree for another Δt_h seconds, and $\mathcal{N}$ is a normalisation constant to keep the sum of utilities in the same range as the step utilities.

5.4 Generalised Dynamic Cognitive Hierarchy Model

In order to support heterogeneous behaviour models, the generalised dynamic cognitive hierarchy model consists of three layers of increasing sophistication of strategic reasoning, each of which can hold multiple behaviour models. The three layers include: a) *non-strategic*, where agents do not reason about the other agents' strategies, b) *strategic*, where agents can reason about the strategies of other agents, and c) *robust*, where agents not only reason about the strategies of other agents but also the behaviour model [237] that other agents may be following (Fig. 5.2a). All of these layers operate in a setting of a dynamic game and I present the models and the solution concepts used in each layer in order.

5.4.1 Non-strategic Layer

Similar to level-0 models in the standard level-k reasoning [40], non-strategic agents form the base of the cognitive hierarchy in the model. However, in this case, I extend the behaviour for the dynamic game. The main challenge of constructing the right level-0 model for a dynamic game is that it has to adhere to the formal constraints of non-strategic behaviour, i.e. not reason over other agents' utilities [237], while at the same time it cannot be too elementary for the purpose of modelling human driving.

I propose that automata strategies, which were introduced as a way to address bounded rational behaviour that results from agents having limited memory capacity to reason over the entire strategy space in a dynamic game [187, 149], address the above problem by striking a balance between adequate sophistication and non-strategic behaviour. To this end, I extend the standard level-k model for a dynamic game setting with level-0 behaviour mediated by a finite state automata (referred to as dLk($\mathcal{A}$) henceforth in the chapter). However, modeling strategies with the help of a finite state automata (FSA) is just a modeling paradigm, and does not say much about the specific strategies of players in a game. Even with the level-0 constraint of non-strategic behaviour, one can construct several different strategies (modelled through an FSA) that can act as a model of level-0 behaviour. Evaluating the effectiveness of all such FSA based strategies in the context of traffic behaviour is beyond the scope of this chapter. Instead, I develop two models of manoeuvre selection using FSAs that enable level-0 agents demonstrate sophisticiated non-strategic behaviour.

I use Finite State Transducer (FST), which is a refinement of FSA, as a model of level-0 behaviour. FST is a type of FSA that supports both input and output symbols, therefore being well suited to modelling input (state) to output (action) based control processes [185]. In the proposed model, a level-0 agent has two reactive modes of operation, each reflecting a specific driving style modeled through a FST; *accomodating* FST ($\mathcal{A}^{\text{AC}}$), which always waits whenever safe, and *non-accomodating* FST ($\mathcal{A}^{\text{NAC}}$) (Fig. 5.2b), which always proceeds whenever safe based on risk tolerance. Next, I present the formal definition of FST along with the specific connections to the game constructs.

Definition 6. *A Finite State Transducer is a 6-tuple* $(M, \Sigma_1, \Sigma_2, \bigcirc, \delta)$ *such that*

- M *is the set of states. In this case, each state in* M *represents the high-level manoeuvre currently executed by the road user.*
- Σ_1 *is a finite set representing the input alphabet. In both* $\mathcal{A}^{AC}$ *and* $\mathcal{A}^{NAC}$*, this set is constructed based on Boolean properties over the game state* S*.*[2]
- Σ_2 *is a finite set representing the output alphabet. In both* $\mathcal{A}^{AC}$ *and* $\mathcal{A}^{NAC}$*, this set is constructed based on the output action of the player following the FST, i.e.,* $\Sigma_2 = \{wait, proceed\}$*.*
- $\bigcirc$ *is the initial state. In this state, the player has yet to choose a manoeuvre.*
- $\delta : \{M \bigcup \bigcirc\} \times \Sigma_1 \rightarrow M \times \Sigma_2$ *is the transition relation that models the progress of the FST.*

[2] The state of the FST being an element of M, tracks operation of the FST process executed by a level-0 player, whereas the game state X is a common state of all players in the game as discussed in Sec 5.3

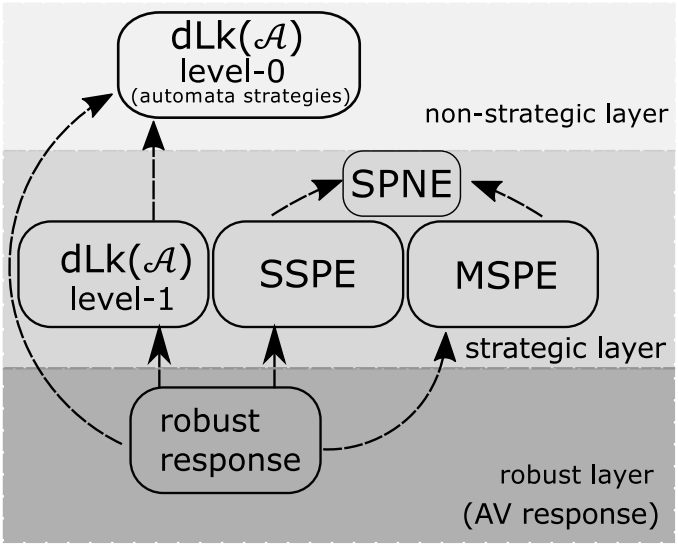
The process of selecting a trajectory is as follows. An agent i playing $\mathcal{A}^{\text{AC}}$ as a type $\gamma_i^{\text{AC}} \in \Gamma$ in state W_{AC} (Fig. 5.2b) randomize uniformly between trajectories belonging to *wait* manoeuvres that have step safety utility at least γ_i^{AC}. If no such *wait* trajectories are available to them, they move to state P_{AC} and randomizes uniformly between trajectories belonging to *proceed* manoeuvre. $\mathcal{A}^{\text{NAC}}$ is similar, but the states are reversed thereby resulting in a predisposition that prefers the proceed state. The switching between the states of the FST is mediated by the set $\Sigma_1 = \{\phi_i^{\text{AC}}, \neg\phi_i^{\text{AC}}\}$ in $\mathcal{A}^{\text{AC}}$ and $\Sigma_1 = \{\phi_i^{\text{NAC}}, \neg\phi_i^{\text{NAC}}\}$ in $\mathcal{A}^{\text{NAC}}$. ϕ_i^{AC} and ϕ_i^{NAC} are boolean properties over the current game state X_t in the game as follows:

$$\phi_i^{\text{AC}} := \max_{\substack{\pi_i(X_t) \in \text{T}_\text{i}(X_t)|_\text{W} \\ \pi_{-i}(X_t)}} u_{s,i}^{\Delta t_p}(\pi_i(X_t), \pi_{-i}(X_t)) > \gamma_i^{\text{AC}} \tag{5.1}$$

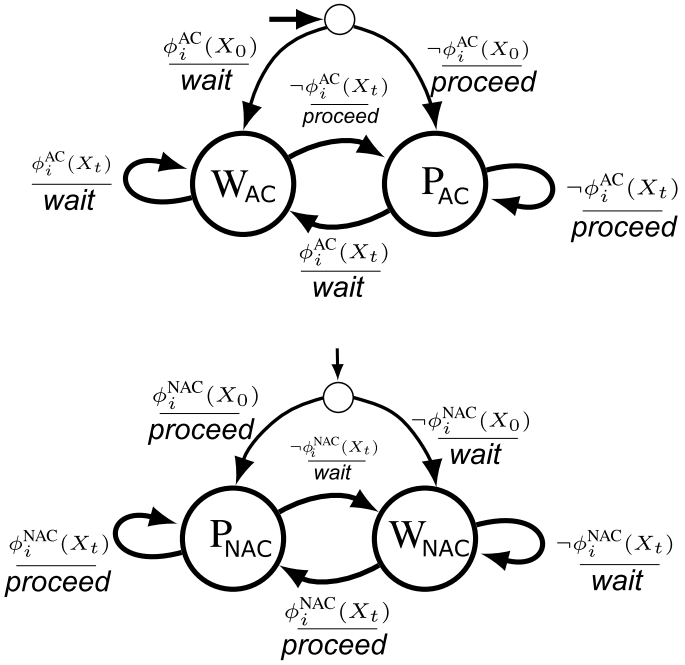
$$\phi_i^{\text{NAC}} := \max_{\substack{\pi_i(X_t) \in \text{T}_\text{i}(X_t)|_\text{P} \\ \pi_{-i}(X_t)}} u_{s,i}^{\Delta t_p}(\pi_i(X_t), \pi_{-i}(X_t)) > \gamma_i^{\text{NAC}} \tag{5.2}$$

ϕ_i^{AC} is true when there is at least one wait trajectory (and conversely, at least one proceed trajectory in the case of ϕ_i^{NAC}) available whose step safety utility ($u_{s,i}^{\Delta t_p}$) at the current node X_t is above γ_i^{AC} (and γ_i^{NAC} for ϕ_i^{NAC}). Recall that agent types are in the range [-1,1] and are reflective of safety aspiration.

One can see that in the above construction, the agent type is conditioned on the specific FST rather than being a fixed value attached to an agent. This allows for a player to have different risk tolerance depending on the FST they use in the game. Along with switching between states, agents are also free to switch between the two FSTs; however, I leave open the question of modeling the underlying process and the causal factors. I envision them to be non-deterministic and a function of the agent's affective states, such as impatience, attitude, etc., which are an indispensable component of modeling driving behaviour [191]. For example, one can imagine a left-turning vehicle approaching the intersection starting with an accommodating strategy, but along the game play, changing its strategy to non-accommodating on account of impatience or other endogenous factors; and in each FST, they may have different safety aspiration levels. Although leaving open the choice of mode switching leads to the level-0 model in this paradigm to be partly descriptive rather than predictive, as we will see in the next section, such a choice does not compromise the ability of higher level agents' to form consistent beliefs about the level-0 agent and respond accordingly to their strategies. This richer model of level-0 behaviour not only imparts more realism in the context of human driving, but also allows for level-0 agent to adapt their behaviour based on the game situation in a dynamic game.



(a)



(b)

Figure 5.2: (a) Organization of models in the generalised dynamic cognitive hierarchy framework. Dashed arrows indicate agents' belief about the population. (b) Automata $\mathcal{A}^{\mathrm{AC}}$ (*accomodating*) and $\mathcal{A}^{\mathrm{NAC}}$ (*non-accomodating*)

5.4.2 Strategic Layer

The difference between non-strategic and strategic models is that while both adhere to the property of *dominance responsiveness*, i.e. optimising over their own utilities, the latter adhere to an additional property of *other responsiveness*, i.e., reasoning over the utilities of the other agents [237]. Agents in the strategic layer in this framework adhere to the above two properties and I include three models of behaviour, namely, level-k ($k \geqslant 1$) behaviour based on the dLk($\mathcal{A}$) model and two types of bounded rational equilibrium behaviour, i.e. SSPE (Safety satisfied perfect equilibria) and MSPE (manoeuvre satisfied perfect equilibria). I note that the choice of models in the strategic layer is not exhaustive; however, I select popular game-theoretic models (level-k and equilibrium based) that have been used in the context of autonomous driving and address some of the gaps within the use of those models as secondary contributions.

For the strategic models, I use satisficing as a method of utility aggregation (Chapter 4), and the aggregation parameter is the same as the agent's type γ_i. Specifically, the combined utility $u_i(\pi_i(h), \pi_{-i}(h))$ is equal to $u_{s,i}(\pi_i(h), \pi_{-i}(h))$, i.e., the safety utility when $u_{s,i}(\pi_i(h), \pi_{-i}(h)) \leqslant \gamma_i$, and otherwise $u_i(\pi_i(h), \pi_{-i}(h)) = u_{p,i}(\pi_i(h), \pi_{-i}(h))$, i.e., the progress utility. Based on the aggregation method, an agent would consider the progress utility of an action only when it's safety utility is greater than γ_i. This way of aggregation provides a natural connection between an agent's type and their risk tolerance. In the following sections, when I use the agent types in different behaviour models, I also index the agent types with the specific models for clarity.

dLk($\mathcal{A}$) Model

A level-1 agent[3] believes that the population consists solely of level-0 agents, and generates a best response to those (level-0) strategies. In a dynamic setting, however, a level-1 agent has to update it's belief about level-0 agent based on observation of the game play. This means that in order to best respond to level-0 strategy, a level-1 agent should form a *consistent* belief based on the observed history h_t of the game play about the type $(\gamma_{-i}^{\text{AC}}, \gamma_{-i}^{\text{NAC}})$ a level-0 agent plays in each automaton. The tuple of types follows from the construction that since each level-0 agent has two modes of operation as modelled by the automata, the types in each of these two automata can be different, therefore the belief of level-1 agent also needs to reflect the same construction.

Definition 7. *A belief* $\mathcal{B}_{l1} = \{\hat{\Upsilon}_{l0}^{AC}, \hat{\Upsilon}_{l0}^{NAC}\}$ *of a level-1 agent at history* h_t *is* consistent *iff* $\forall \gamma_{-i}^{AC} \in \hat{\Upsilon}_{l0}^{AC}$*, the history of manoeuvres* $m(\pi_{-i}(X_0))...m(\pi_{-i}(X_t))$ *is in the output language of FST* $\mathcal{A}^{AC}(\gamma_{-i}^{AC})$ *and* $\forall \gamma_{-i}^{NAC} \in \hat{\Upsilon}_{l0}^{NAC}$*,* $m(\pi_{-i}(X_0))...m(\pi_{-i}(X_t))$ *is in the output language of FST* $\mathcal{A}^{NAC}(\gamma_{-i}^{NAC})$.

[3] I focus on $k = 1$ behaviour in this section as well as later in the experiments, but the best response behaviour can be extended to $k > 1$ similar to a standard level-k model in normal form games.

Before the level-1 agent has observed any action by the level-0 agent, their estimates for $\hat{\Upsilon}_{l0}^{\text{AC}}$ and $\hat{\Upsilon}_{l0}^{\text{NAC}}$ is the default set of types, i.e. $\hat{\Upsilon}_{l0}^{\text{AC}} = \hat{\Upsilon}_{l0}^{\text{NAC}} = \Gamma$ with range 2 (recall $\Gamma = [-1, 1]$). However, over time with more observations of level-0 actions, the level-1 agent forms a tighter estimate, i.e. $|\max(\hat{\Upsilon}_{l0}^{\text{AC}}) - \min(\hat{\Upsilon}_{l0}^{\text{AC}})| \leqslant 2$, of level-0's type. If the observed history is inconsistent with that of an automaton, the belief for the corresponding automaton will be an empty set. The following theorem formulates the set of consistent inference of beliefs $\mathcal{B}_{l1}$ based on observed level-0 actions with history h more formally.

Theorem 2. *Given a history h of a game, a consistent belief $\gamma_{l0}^{AC} \in \hat{\Upsilon}_{l0}^{AC}$ and $\gamma_{l0}^{NAC} \in \hat{\Upsilon}_{l0}^{AC}$ for a level-1 agent is given by,*

$$\begin{aligned}
\gamma_{l0}^{AC} \quad & < \min_{X \in h[W]} \max_{\pi_{l0}(X) \in \mathrm{T}_{l0}(X)|_W} u_{s,l0}^{\Delta_{t_p}}(\pi_{l0}(X), \pi_{l1}(X)) \\
& \geqslant \max_{X \in h[P]} \max_{\pi_{l0}(X) \in \mathrm{T}_{l0}(X)|_W} u_{s,l0}^{\Delta_{t_p}}(\pi_{l0}(X), \pi_{l1}(X)) \\
\gamma_{l0}^{NAC} \quad & \geqslant \max_{X \in h[W]} \max_{\pi_{l0}(X) \in \mathrm{T}_{l0}(X)|_P} u_{s,l0}^{\Delta_{t_p}}(\pi_{l0}(X), \pi_{l1}(X)) \\
& < \min_{X \in h[P]} \max_{\pi_{l0}(X) \in \mathrm{T}_{l0}(X)|_P} u_{s,l0}^{\Delta_{t_p}}(\pi_{l0}(X), \pi_{l1}(X))
\end{aligned}$$

where $h[P]$ and $h[W]$ are a set of nodes in the game where the level-0 agent chose a proceed and wait manoeuvre, respectively, and $T(X)|_P$,$T(X)|_W$ represent the available trajectories at node X belonging to the two manoeuvres, proceed and wait, respectively.

The intuition behind the proof is that $h[W]$ contain the set of nodes X in which the level-0 agent chose to wait, and $h[P]$ are the nodes where the observed action was proceed. In each of these nodes, if the level-0 agent was playing $\mathcal{A}^{\text{AC}}$ (or $\mathcal{A}^{\text{NAC}}$), then the condition of Eqn. 5.1 (or Eqn. 5.2) has to be met. The conjunction of all such conditions over the history of nodes provides the bounds of the parameters that would be consistent with all the observations.

Proof. There are two parts to the equation, one for γ_{l0}^{AC} and another γ_{l0}^{NAC}. I prove the bounds of γ_{l0}^{AC} corresponding to automata $\mathcal{A}^{\text{AC}}$, and the proof for γ_{l0}^{NAC} corresponding to automata $\mathcal{A}^{\text{NAC}}$ follows in an identical manner.

By construction of automata $\mathcal{A}^{\text{AC}}$, proceed trajectories are only generated in state P_{AC}, which follows one of the three transitions $\{\bigcirc|\text{P}_{\text{AC}}|\text{W}_{\text{AC}}\} \xrightarrow{\neg\phi_{l0}^{\text{AC}}(X)} \text{P}_{\text{AC}}$. Therefore, $\forall X \in h[\text{P}]$, $\phi_{l0}^{\text{AC}} \coloneqq \bot$ for the transition to happen. Based on eqn. 1, this means that $\forall X \in h[\text{P}]$,

$$\gamma_{l0}^{\text{AC}} \geqslant \max_{\pi_{l0} \in \mathrm{T}_{l0}(X)|_{\text{W}}} u_{s,i}^{\Delta_{t_p}}(\pi_{l0}(X), \pi_{l1}(X)) \tag{A.1}$$

Since γ_{l0}^{AC} stays constant throughout the play of the game, for γ_{l0}^{AC} to be consistent for the set of all nodes $X \in h[\text{P}]$, the lower bound (i.e., at least as high to be true for all nodes $X \in h[\text{P}]$) of γ_{l0}^{AC} based on eqn A.1 is

$$\gamma_{l0}^{\text{AC}} \geqslant \max_{X \in h[\text{P}]} \max_{\pi_{l0} \in \text{T}_{\text{l0}}(X)|_W} u_{s,l0}(\pi_{l0}(X), \pi_{l1}(X)) \tag{A.2}$$

Similarly, by construction of automata, wait trajectories are only generated in state W_{AC}, which follows one of the three transitions $\{\bigcirc|\text{P}_{\text{AC}}|\text{W}_{\text{AC}}\} \xrightarrow{\phi_{l0}^{\text{AC}}(X)} \text{W}_{\text{AC}}$. Therefore, $\forall X \in h[\text{W}]$, $\phi_{l0}^{\text{AC}} \coloneqq \top$ for the transition to happen. Based on eqn. 1, this means that $\forall X \in h[\text{W}]$,

$$\gamma_{l0}^{\text{AC}} < \max_{\pi_{l0} \in \text{T}_{\text{l0}}(X)|_{\text{P}}} u_{s,i}^{\Delta t_p}(\pi_{l0}(X), \pi_{l1}(X)) \tag{A.3}$$

and the upper bound (i.e., at least as low to be true for all nodes $X \in h[\text{W}]$) of γ_{l0}^{AC} based on eqn A.3 $\forall X \in h[\text{W}]$

$$\gamma_{l0}^{\text{AC}} < \min_{X \in h[\text{W}]} \max_{\pi_{l0} \in \text{T}_{\text{l0}}(X)|_W} u_{s,l0}(\pi_{l0}(X), \pi_{l1}(X)) \tag{A.4}$$

Since $h[\text{P}] \bigcap h[\text{W}] = \varnothing$ and $h = h[\text{P}] \bigcup h[\text{W}]$, equations A.2 and A.4 in conjunction proves the case for γ_{l0}^{AC} bounds.

The proof for γ_{l0}^{NAC} follows in the identical manner as γ_{l0}^{AC}, but with the condition reversed based on the P and W states. □

The above theorem formalizes the idea that looking at manoeuvre choices made by a level-0 agent at each node in the history, as well as the range of step safety utility at that node for both manoeuvres (recall that preference conditions of the automata are based on step safety utility), a level-1 agent can calculate ranges for $\gamma_{-i}^{\text{AC/NAC}}$ from Eqns 5.1 and 5.2, for which the observed actions were consistent with each automata. With respect to the set of consistent belief $\mathcal{B}_{l1}$ about level-0 agent's strategy, level-1 agent generates a best response that is consistent with $\mathcal{B}_{l1}$. Dropping the AC/NAC superscripts Let $\pi(X_t; \mathcal{A}, \hat{\Upsilon}_{l0}) = \{\pi(X_t; \mathcal{A}, \gamma_{-i}); \forall \gamma_{-i} \in \hat{\Upsilon}_{l0}\}$ be the union of all actions when the automata $\mathcal{A}$ is played by the types in $\hat{\Upsilon}_{l0}$, then $\Pi_{l0}(X_t, \mathcal{B}_{l1}) = \pi(X_t; \mathcal{A}^{\text{AC}}, \hat{\Upsilon}_{l0}^{\text{AC}}) \bigcup \pi(X_t; \mathcal{A}^{\text{NAC}}, \hat{\Upsilon}_{l0}^{\text{NAC}})$ is the set of all actions that level-0 agent can play based on level-1's consistent belief $\mathcal{B}_{l1}$. The response to those actions by level-1 agent (indexed as i) is as follows.

$$\pi_i(h_t; \mathcal{B}_{l1}) = \underset{\substack{\text{T}_i(X_t) \\ \Pi_{l0}(X_t, \mathcal{B}_{l1})}}{\arg\max} \; u_i(\pi_i(h), \pi_{-i}(X_t)|\gamma_i^{l1}) \tag{5.3}$$

where $\gamma_i^{l1} \in \Gamma$ is the type of the level-1 agent, $\pi(h) \in \mathrm{T}_i(X_t)$ is a valid trajectory that can be generated at the node with history h, and $\pi_{-i}(X_t) \in \Pi_{l0}(X_t, \mathcal{B}_{l1})$ is the other agent's trajectory. Note that the strategy of the level-1 agent, unlike level-0 agent depends on the history h_t instead of just the state of the node X_t; since the history influences the belief $\mathcal{B}_{l1}$ which in turn influences the response.

The above model of best response can be extended similarly for $k > 1$ just like in standard level-k models. However, I restrict the focus for k up to 1 mainly due to following reasons: i) aspects of higher level thinking are already incorporated in the equilibrium based models; ii) in other domains, I often see diminishing returns with higher k values with respect to empirical data [234].

5.4.3 Equilibrium Models

Along with the level-k (k$\geqslant$ 1) behaviour, another notion of strategic behaviour that have been proposed as a model of behaviour planning in AV [179, 201] are based on an equilibrium. However, when it comes to modelling human driving behaviour, an equilibrium model needs to accommodate bounded rational agents in a principled manner, and ideally should provide a reasonable explanation of the origin of the bounded rationality. Based on the idea that drivers are indifferent as long as an action achieves their own subjective safety aspiration threshold [132], I use the idea of *satisficing* as the main framework for bounded rationality in the equilibirum models [207]. Specifically, I develop two notions of satisficing; one based on safety satisficing (SSPE), where agents choose actions close to the Nash equilibria as long as the actions are above their own safety aspiration threshold, and another based on manoeuvre satisficing (MSPE), where agents chose actions close to the Nash equilibria as long as the actions are of the same high-level manoeuvre as the optimal action.

Safety-satisfied Perfect Equilibrium (SSPE)

The main idea behind satisficing is that a bounded rational agent, instead of always selecting the best response action, selects a response that is *good enough*, where *good enough* is defined as an aspiration level where the agent is indifferent between the optimal response and the response in question. In the case of Safety-satisfied perfect equilibrium (SSPE), I define a response good enough for agent i if the response is above their own safety aspiration threshold as determined by their type. The goal of this model is to let agents chose their action based on the subgame perfect Nash equilibrium of the game, however, also allow for selection of actions close to the equilibrium that are safe enough based on their own safety aspiration level. Therefore, in response

to other agents' equilibrium action, any action that is safer than either their own equilibrium action or their own safety aspiration level γ_i is part of the solution. A more formal definition is as follows

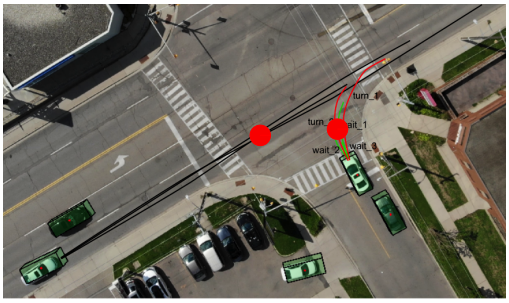
Definition 8. *A strategy* $\sigma; (\gamma_i^{SS}, \gamma_{-i}^{SS}) : \prod_{\forall i \in N} \pi_i(h)$*, is in safety satisfied perfect equilibria for a combination of agent types* $(\gamma_i^{SS}, \gamma_{-i}^{SS}) \in \Gamma^N$ *if for every history* h *of the game and* $\forall i \in N$

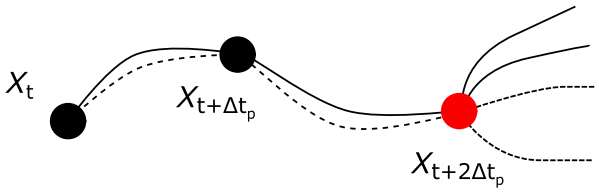
$$u_{s,i}(\pi_i(h), \pi^*_{-i}(h)) \geqslant \min\{u^*_{s,i}(h), \gamma_i^{SS}\}$$

where $\sigma^*; (\gamma_i^{SS}, \gamma_{-i}^{SS}) : \prod_{\forall i \in N} \pi^*_i(h)$*, is a subgame perfect Nash equilibrium in pure strategies of the game for agents with type* $(\gamma_i^{SS}, \gamma_{-i}^{SS})$ *and* $u^*_{s,i}(h) = u_{s,i}(\pi^*_i(h), \pi^*_{-i}(h))$. [4]

Based on the above definition, if the safety utility of the best response of agent i to agent $-i$'s subgame perfect Nash equilibrium (SPNE) strategies at history h is less than agent i's own safety threshold as expressed by their type γ_i^{SS}, then the SSPE response is any trajectory that matches the safety utility of the SPNE response. However, if the SPNE response is higher than their safety threshold, then any suboptimal response that has safety utility higher than γ_i^{SS} is a *satisfied* response, and thus in SSPE.

[4] I calculate the SPNE using backward induction with the combined utilities (lexicographic thresholding) in a complete information setting where agent types are known to each other.





	τ_{-i}^1(proceed)	τ_{-i}^2 (proceed)	τ_{-i}^3 (wait)
τ_i^1 (wait)	0.1p 0.5s 0.9p 0.5s	0.1p 0.5s 0.8p 0.5s	0.1p 0.9s 0.2p 0.9s
τ_i^2 (wait)	0.05p 0.5s 0.9p 0.5s		
τ_i^3 (proceed)	0.5p -0.5s 0.9p -0.5s		
τ_i^4 (proceed)	0.7p -0.1s 0.9p -0.1s		
τ_i^5 (wait)	0.4p -0.75s 0.9p -0.75s		

SPNE (safety satisfied) Trajectory

SPNE Trajectory

Figure 5.3: Illustrative example of Safety Satisfied Perfect Equilibrium by expanding the solution set of Subgame perfect Nash equilibrium (SPNE). Full calculation of the Nash equilibrium is omitted for brevity.

Fig. 5.3 shows an illustrative example of SSPE at a typical unprotected right turn scenario. The normal form matrix for the subgame starting at node marked in red that is constructed in the process of calculating the Subgame Perfect Nash Equilibrium through backward induction is shown in the table. The safety and progress utilities are marked with suffixes $\mathtt{s}$ and $\mathtt{p}$, respectively. The top and bottom numbers in each box in the utility matrix show the two utilities of right turning (i) and straight through vehicle $(-i)$ respectively. The type of both vehicles, γ_i^{SS} and γ_{-i}^{SS} is 0.1. Since this example is for the purpose of illustration, only the utilities that are relevant for that purpose are shown in the figure. Let us say that the trajectory pair (τ_i^1, τ_{-i}^1) is the Nash equilibrium of the subgame. The SSPE set is calculated by expanding the best response of both players to each other's equilibrium action. Specifically, any action that fetches a safety utility greater than 0.1 for each player would be in the SSPE set. These set of actions are shown with a lighter shade in the figure; the trajectory pairs (τ_i^2, τ_{-i}^2). Therefore, SSPE for this subgame includes the two pairs of actions (τ_i^1, τ_{-i}^1) and (τ_i^2, τ_{-i}^2). Since SSPE generates multiple equilibria at a game node, the process of calculating the SSPE for the rest of the game tree proceeds by repeating the backward induction process one subgame SSPE solution at a time.

Manoeuvre-satisfied Perfect Equilibrium (MSPE)

This model of satisficing is based on the idea that agents may be indifferent between actions that belong to the manoeuvre corresponding to the equilibrium trajectory. For example, at any node, if the equilibrium trajectory belongs to the *wait* manoeuvre, then all the trajectories belonging to the *wait* manoeuvre will be in MSPE. However, additional constraints need to be imposed on this manner of action selection. In order to avoid selection of trajectories that belong to equilibrium manoeuvre but have lower utility than a non-equilibrium manoeuvre, I add the constraint that the utility of a selected trajectory has to be strictly higher than that of any non-equilibrium manoeuvre's trajectory. A typical example for the need of this constraint includes situations where the equilibrium trajectory corresponds to *wait* manoeuvre, but selecting a trajectory that is akin to a rolling stop, which although falls under wait manoeuvre, is worse than executing a proceed trajectory, i.e., a non-equilibrium manoeuvre. Therefore, such trajectories should be excluded from the MSPE solution set. A formal definition follows.

Definition 9. *A strategy* $\sigma; (\gamma_i^{MS}, \gamma_{-i}^{MS}) : \prod_{\forall i \in N} \pi_i(h)$*, is in manoeuvre satisfied perfect equilibria for a combination of agent types* $(\gamma_i^{MS}, \gamma_{-i}^{MS}) \in \Gamma^N$ *if for every history* h *of the game and* $\forall i \in N$*,* $m(\pi_i(h)) = m(\pi_i^*(h))$ *and*

$$u_{s,i}(\pi_i(h), \pi^*_{-i}(h)) > \max_{\pi_i^{'}(h) \in \mathrm{T}_i(X) \setminus m^*} u_i(\pi_i^{'}(h), \pi^*_{-i}(h))$$

where $\sigma^*; (\gamma_i^{MS}, \gamma_{-i}^{MS}) : \prod_{\forall i \in N} \pi_i^*(h)$*, is a subgame perfect equilibrium in pure strategies of the game with agent types* $(\gamma_i^{MS}, \gamma_{-i}^{MS})$*, and* $\mathrm{T}_i(X) \setminus m^* = \{\pi_i(h) : m(\pi_i(h)) \neq m(\pi_i^*(X))\}$ *or in other words, the set of available trajectories at node* S *that do not belong to the manoeuvre corresponding to the equilibrium trajectory* $m(\pi_i^*(h))$ *and* S *is the last node in the history* h*.*

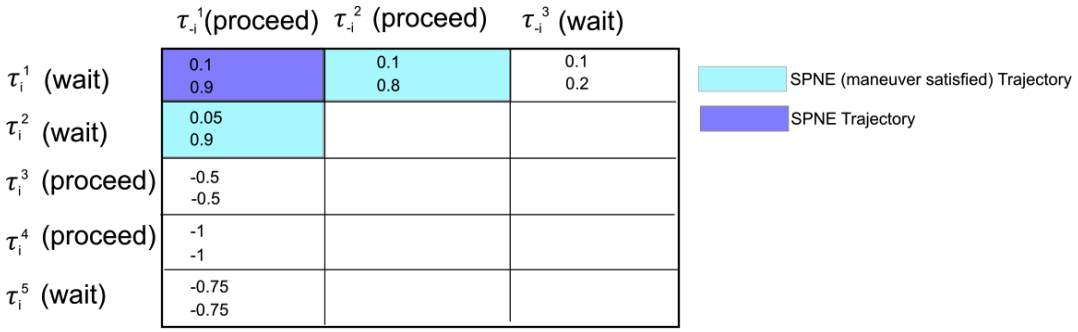


Figure 5.4: Illustrative example of Manoeuvre Satisfied Perfect Equilibrium with $(\gamma_i^{\text{MS}}, \gamma_{-i}^{\text{MS}}) = (0.1, 0.1)$.

Fig. 5.4 shows the utility matrix for the same example in Fig. 5.3 with the multiobjective utilities aggregated into a scalar value based on each agent's type, i.e., 0.1. The only difference between the SSPE calculation and MSPE is the process of expansion of the action sets for each player in relation to the Nash equilibrium solution. For the right-turning vehicle (i), the two *wait* trajectories with utilities 0.1 and 0.05 are in MSPE, since they belong to the same manoeuvre as the equilibrium trajectory. The third *wait* trajectory, however, is not in MSPE, since the utility (-0.75) of that trajectory is lower than the maximum utility the nonequilibrium manoeuvre (-0.5), thereby violating the condition in Definition 9.

5.4.4 Robust Layer

While the presence of multiple models in the strategic layers allows for a population of heterogeneous reasoners, an agent following one of those models still has specific assumptions about the reasoning process of other agents, e.g., level-1 agents believing that the population consists of level-0 agents and equilibrium responders believing that other agents adhere to a common knowledge of rationality. However, it follows from the results in Chapters 3 and [210] that there is no single model of reasoning that can uniquely capture human traffic behaviour. Rather, the performance of models is context-dependent on the specific traffic situation. For behaviour planning, this raises two problems. First, the problem of heterogeneity, meaning that an AV at any point in time can encounter multiple road users, each following a different model of reasoning. Second is the problem of model underspecification, i.e., based on the observed action of other road users an AV can rationalise the same actions of a single road user based on multiple models of reasoning and player-type combination.

In order to address the aforementioned problems, I develop the robust layer. What differ-

entiates the robust layer from the strategic layer is that, along with the two properties of other responsiveness and dominance responsiveness for the strategic layer, agents in the robust layer also adhere to the property of *model responsiveness*, i.e., the ability to reason over the behaviour models of other agents. This gives them the ability to reason about (forming beliefs about and responding to) a population of different types of reasoners including strategic, non-strategic, as well as agents following different models within each layer. The overall behaviour of a robust agent can be broken down into three sequential steps as follows.

i. Type expansion: Since the robust agent has to reason over the types of other agents, but also the possible behaviour models, I augment the initial set of agent types Γ that were based on agents' safety aspiration with the corresponding agent models. Reasoning over this augmented type can then proceed similarly to non-augemented types. Let $\Gamma^+ : \mathcal{M} \times \Gamma$ be the augmented type of an agent, where $\mathcal{M}$ is the set of models presented earlier, i.e., {accomodating, non-accomodating, level-1, SSPE, MSPE} and Γ is the (non-augmented) agent type ($\gamma_i^{\text{AC}}, \gamma_i^{\text{NAC}}, \gamma_i^{\text{l1}}, \gamma_i^{\text{SS}}, \gamma_i^{\text{MS}}$) within each model. For example, an element of Γ^+ can be $(\text{level-1}, 0.5)$, which means that the augmented type of the other player is a dLk($\mathcal{A}$) level-1 reasoner with safety aspiration level (non-augmented type) 0.5.

ii. Consistent beliefs: Similar to strategic agents, based on the observed history h of the game, a robust agent forms a belief $\beta_h \subseteq \Gamma^+$ such that the observed actions of the other agent in the history h are consistent with the augmented types (i.e., model and agent type) in β_h. The process of checking whether a history is consistent with a combination of a model and agent type was already developed earlier for two non-strategic models (Def. 7). For level-1 models, the history is consistent if at each node in history, the response of the other agent adheres to equation 5.3. For the equilibrium models, a history h is consistent if the observed actions follow the equilibrium path of the game tree according to definitions 8 and 9 for SSPE and MSPE respectively. Assuming that in driving situations agents behave truly according to their types, β_h is then constructed as a union of all consistent beliefs for each model. This set represents the possible reasoning models, as well as the types of the other road users in question that can be rationalised based on the observed actions.

iii. Robust response: In the final step, the agent in the robust layer responds to the possible actions of other agents based on consistent beliefs. The belief set β_h represents the uncertainty over the possible models the other agents may be using along with the corresponding agent types within those models. The idea of a robust response to heterogeneous models is along the lines of the robust game theory approach of [5]. An alternate method of responding to heterogeneous types is by estimating the expected utilities of each action according to a distribution of the types of other agents. However, I favour the distribution-free approach of [5] since in our case, we only have constructed the support of the possible augmented types (β_h) rather than a distribution

over β_h[5]. A robust response to that is the optimisation over the worst possible outcomes that can happen with respect to that set. Eqn. 5.4 formulates this response of a robust agent playing as agent i.

$$\pi_i(h;\beta_h) = \arg\max_{\mathrm{T}_i(X)} \min_{\forall\beta\in\beta_h} \max_{\forall\pi_{-i}(h;\beta)} u_i(\pi(h), \pi_{-i}(h;\beta); \gamma_i^{\mathrm{R}}) \tag{5.4}$$

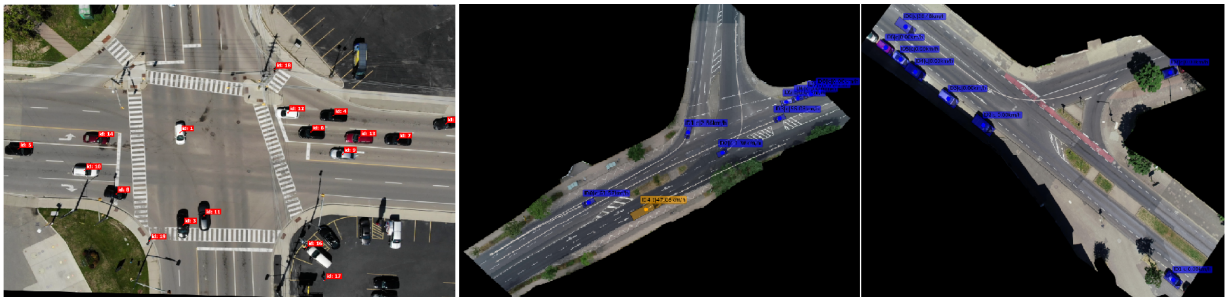
where $\pi_{-i}(h;\beta)$ are the possible actions of the other agent based on the augmented type $\beta \in \beta_h$ and $\gamma_i^{\mathrm{R}} \in \Gamma$ is the robust agent's own type. In this response, the minimisation happens over the agent types (inner $\min$ operator), rather than over all actions $\pi_{-i}(h)$ as is common in a maxmin response. Since driving situations are not, in most cases, purely adversarial, this is a less conservative, yet robust, response compared to a maxmin response.

I illustrate the calculation of the robust response with the help of the same game matrix as in Fig. 5.3 from the perspective of the column player with $\gamma^{\mathrm{R}} = 0.1$. Let the example belief set β_h of the column player (based on the row player's history of action) be $\beta_h = \{(\text{level-0}, \gamma^{\mathrm{NAC}} = -0.6), (\text{SSPE},0.1), (\text{MSPE},0.1)\}$. From the perspective of the column player, the innermost $\max$ operator in Eqn. 5.4 calculates the maximum utility they can achieve by playing against each type of row player in the belief set β_h. The row player's actions corresponding to the three types in β_h are (τ_i^3, τ_i^4), (τ_i^1, τ_i^2), and (τ_i^1, τ_i^2) respectively. One can check that $\mathcal{A}^{\mathrm{NAC}}$ with $\gamma^{\mathrm{NAC}} = -0.6$ would randomise between the two proceed trajectories since the maximum safety utility is higher than -0.6. Consequently, for the column player's action τ_{-i}^1, the innermost $\max$ operator evaluates to -0.5, 0.9, and 0.9 for those three sets of row player's actions. Since -0.5 is the result of the $\min$ operator over this set, the utility that the column player can expect to get playing τ_{-i}^1 is -0.5 based on the robust response formulation in Eq. 5.4. Note that this is higher than a maxmin based calculation for τ_{-i}^1, which fetches -1, thereby demonstrating a less conservative behaviour of a robust response compared to maxmin. Once this calculation is repeated for τ_{-i}^2 and τ_{-i}^3, the robust response of the column player would be the action that fetches the maximum utility among τ_{-i}^1, τ_{-i}^2, and τ_{-i}^3.

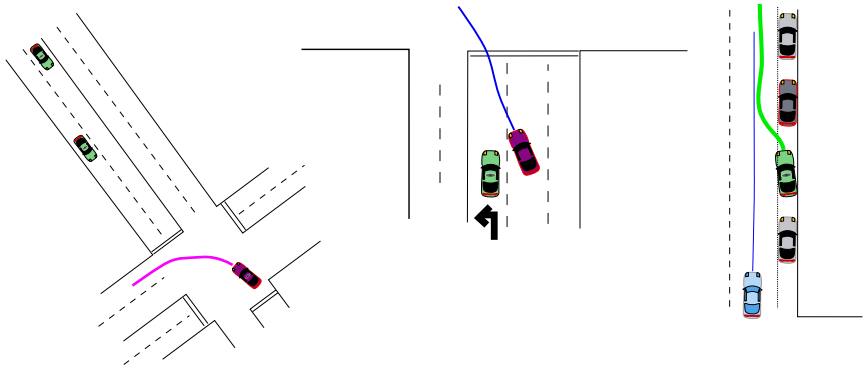
5.5 Experiments and Evaluation

In this section, I present the evaluation of the models under two different experiment setups. First, I compare the models with respect to large naturalistic observational driving data using a) the Intersection dataset from the Waterloo multi-agent traffic dataset (WMA) recorded at a busy

[5] The main property of a robust layer is *model responsiveness* rather than a specific method of response to the augmented types. The method of response chosen in this chapter just happens to bear the title of 'robust' following from [5]. In future one can also construct a response according to the Bayesian Nash equilibrium as an alternative without violating the basic model responsiveness property of the robust layer.



(a) Snapshot of naturalistic datasets (WMA and inD)



(b) Simulation of critical scenarios: intersection clearance, merge before intersection, parking pullout.

Figure 5.5: Evaluation setup based on naturalistic datasets and simulation scenarios.

Canadian intersection (Appendix B), and b) the inD dataset recorded at intersections in Germany [27] (Fig. 5.5a). From both datasets, which include around 10k vehicles in total, I extract the long duration unprotected left turn (LT) and right turn (RT) scenarios, and instantiate games between left (and right) turning vehicles and oncoming vehicles with $\Delta t_h = 6$s and $\Delta t_p = 2$s, resulting in a total of 1678 games. The second part of the evaluation is based on simulation of three critical traffic scenarios derived from the NHTSA pre-crash database [160], where I instantiate agents with a range of safety aspirations as well as initial game states, and evaluate the outcome of the game based on each model. All the games in the experiments are 2 agent games with the exception of one of the simulation of critical scenario (intersection clearance), which is a 3 agent game. Code and supplementary videos are available at https://git.uwaterloo.ca/a9sarkar/repeated_driving_games.

Baselines. I select multiple baselines depending on whether a model is strategic or non-strategic. For non-strategic models, I compare the automata based model with a maxmax model, shown to be most promising from a set of alternate elementary models with respect to naturalistic data

[196]. For the strategic models (level-1 in dLk($\mathcal{A}$), SSPE, MSPE), I select a Qlk model used in multiple works within the context of autonomous driving [136, 215, 216, 140]. I use the same parameters used in [216] for the precision parameters in the Qlk model, i.e., $\lambda = 0.5, 1$.

Naturalistic data

I evaluate the performance of the models based on the match rate, i.e., the number of games where the observed strategy of the human driver matched the solution of a model divided by the total number of games in the data set. More formally, let D be the set of games in the dataset, an indicator function $I : D \rightarrow \{0, 1\}$ be 1 if in the game g, there exists a combination of agent types (γ_i, γ_{-i}), such that the observed strategy is in the set of strategies predicted by the model or 0 otherwise. The overall match rate of a model is given by $\sum_{\forall g \in D} I/|D|$. Qlk (baseline) models being mixed strategy models, I count a match if the observed strategy is assigned a probability of $\geqslant 0.5$. Table 5.1 shows the match rate of each model for each dataset and scenario. It also shows in parenthesis the mean γ_i, i.e. the agent type value for each model when the strategy matched the observation. Since the agent types are a free parameter within each model, this gives the models flexibility to match the observed action to a driver's safety aspiration level. The overall numbers in the table show that there is variation both with respect to match rate as well as the matched safety aspiration level (agent type) for a given model. The match rate, especially for the WMA dataset, is better for right-turning scenarios than for left-turning ones. The models sometimes find it harder to match a consistent safety aspiration level at left turns in WMA when vehicles exhibit impatient behaviour by creeping forward. Notice that the difference is less stark in the inD dataset because in inD (Fig. 3a), the LT vehicle is still in the turn lane at point of initiation (i.e., before the stopline), and therefore has less incentive to take a risk and creep.

A major takeaway is that for non-strategic models, automata models show much higher match rate thereby reflecting high alignment with human driving behaviour compared to the maxmax model. In fact, as I can see from the table that the entries for AC and NAC sum up to 1. The combination of AC and NAC, although being non-strategic, can capture all observed driving decisions in the dataset, which indicates that automata models are very well suited for modelling level-0 behaviour in a dynamic game setting for human driving. The performance of an accommodating model is not very surprising since for left and right turning scenarios, most drivers generally give way to oncoming vehicles. For the strategic models, dLK($\mathcal{A}$) and SSPE models show better performance than Qlk and MSPE models. However, the mean agent type values for the case when SSPE strategies match the observation are very low. Under the reasonable assumption that the population of drivers on average have moderate safety aspiration level, say in the range [-0.5,0.5], dLk($\mathcal{A}$) is a more reasonable model of strategic behaviour compared to SSPE. I include the robust model comparison for the sake of completeness (and it shows perfor-

	WMA		inD	
	LT (1103)	RT (311)	LT (187)	RT (77)
maxmax	0.33438	0.43023	0.37975	0.43506
	(0.02)	(0.27)	(-0.2)	(0.1)
AC	0.82053	0.90698	0.92089	0.81818
	(0.82)	(0.84)	(0.75)	(0.79)
NAC	0.17947	0.09302	0.07911	0.18182
	(0.87)	(0.82)	(0.88)	(0.85)
Qlk(λ=1)	0.18262	0.43265	0.37658	0.43506
	(-0.07)	(0.33)	(0.37)	(-0.1)
Qlk(λ=0.5)	0.34131	0.43023	0.37658	0.43506
	(-0.03)	(0.26)	(0.37)	(-0.1)
dLk($\mathcal{A}$)	0.5529	0.65449	0.51266	0.53247
	(0.19)	(0.3)	(0.51)	(-0.4)
SSPE	0.69144	0.90033	0.6962	0.53247
	(-0.84)	(-0.94)	(-0.86)	(-0.85)
MSPE	0.30479	0.44518	0.21519	0.27273
	(-0.1)	(-0.13)	(0.21)	(-0.7)
Robust	0.56045	0.66944	0.51582	0.53247
	(0.20)	(0.34)	(0.51)	(-0.4)

Table 5.1: Overall match rate of the models for each dataset and scenario. Mean agent type (γ) noted in parenthesis. LT: Left turn, RT: Right turn. Number of games noted in the header.

mance comparable to dLk($\mathcal{A}$) model), but as mentioned earlier, the robust model is a model of behaviour planning for an AV, and therefore ideally needs to be evaluated on criteria beyond just comparison to naturalistic human driving, which I discuss in the next section.

Critical scenarios

While evaluation based on a naturalistic driving datasets helps in the understanding of how well a model matches human driving behaviour, in order to evaluate the suitability of a model for behaviour planning of an AV, the models need to be evaluated on specific scenarios that encompass the operational design domain (ODD) of the AV [97]. Since the models developed in this chapter are not specific to an ODD, I select three critical scenarios from the ten most frequent crash scenarios in the NHTSA pre-crash database [160].

Intersection Clearance (IC): Left turn across path (LTAP) scenario where the traffic signal

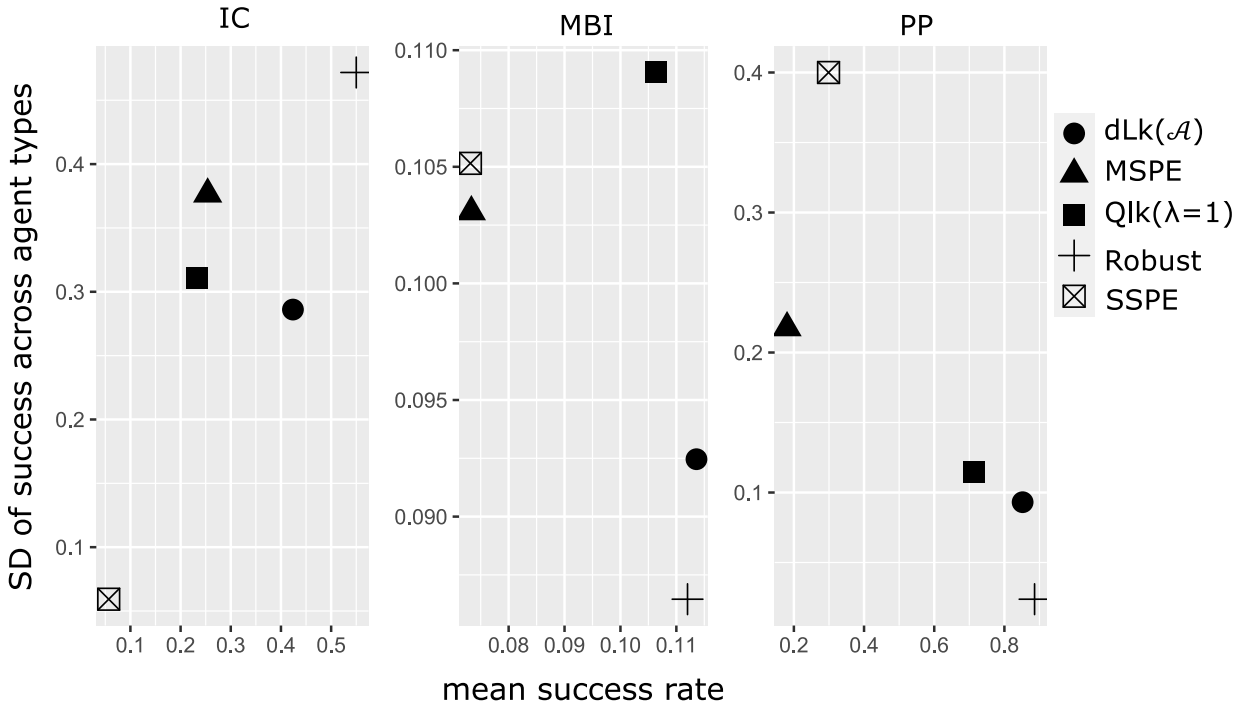


Figure 5.6: Mean and SD of success for each model in each scenario across all agent types.

has just turned from green to yellow at the moment of the game initiation. There is a left turning vehicle in the intersection and two oncoming vehicles from the opposite direction close to the intersection who may chose to speed and cross or wait for the next cycle. The expectation is that the left turning vehicle should be able to clear the intersection by the end of the game horizon without crashing into either oncoming vehicles, and no vehicles should be stuck in the middle of the intersection.

Merge Before Intersection (MBI): Merge scenario where a left-turning vehicle (designated as the merging vehicle) finds itself in the wrong lane just prior to entering the intersection, and wants to merge into the turn lane in front of another left-turning vehicle (designated as the on-lane vehicle). The expectation is that the on-lane vehicle should allow the other vehicle to merge.

Parking Pullout (PP): Merge scenario where a parked vehicle is pulling out of a parking spot and merges into traffic while there is a vehicle coming along the same direction from behind. The expectation is that the parked vehicle should wait for the coming vehicle to pass before merging into traffic.

For each scenario, simulations are run with a range of approach speeds, as well as all combination of agent types from the set of agent types $\Gamma = \{-1, -0.5, 0, 0.5, 1\}$.

One way to compare the models is to evaluate them based on the mean success rate across all initiating states and agent types. Fig. 5.6 shows the mean success rate (success defined as the desired outcome based on expectation defined in the description for each scenario) for all the strategic and robust models. The mean success rate of the robust and dLk(A) model is higher compared to the equilibrium models or the Qlk model. However, this is only part of the story. With varying initiation conditions, it may be harder or easier for a model to lead to a successful outcome. For example, in the parking pullout scenario a vehicle with low safety aspiration coming at a higher speed is almost likely to succeed in all models when facing a parked vehicle with high safety aspiration at zero speed. Therefore, to tease out the stability of models across different safety aspirations (i.e. agent type combinations), Fig. 5.6 also plots on y-axis, the standard deviation of the mean success rate across different agent types. Ideally, a model should have a high success rate with low SD across types indicating that with different combinations of agent type population (from extremely low safety aspiration to very high), the success rate stays stable. As seen in Fig. 5.6, robust and dLk(A) models are broadly in the ideal lower right quadrant (high mean success rate, low SD) for parking pullout and merge before intersection scenarios. However, for the IC scenario, the success rate comes at a price of high SD (for all models), as indicated by the linearly increasing relationship between the mean success rate and its SD across agent types. This means that the success outcomes are skewed towards a specific combination of agent types; specifically, the case where the left turning vehicle has low safety aspiration. It is intuitive to imagine that in a situation like IC, agents with high safety aspiration would be stuck in the intersection instead of being able to navigate out of the intersection quickly.

Finally, the failure of models to achieve the expected outcome can also be due to a crash (minimum distance gap between trajectories $\leqslant 0.1$m) instead of an alternate outcome (e.g. getting stuck in the intersection). In all simulations, for the parking pullout and intersection clearance, no crashes were observed for any of the models. However, for the merge before intersection, due to starting at a more risky situation than the other two in terms of the chance of a crash, the crash rate (ratio of crashes across all simulations) for the models across all initial states and agent types were as follows: (dlk($\mathcal{A}$): 0.052, MSPE: 0.022, SSPE: 0.007, Qlk(level-1): 0.026, Robust: 0.053). Qlk (level-1) demonstrates conservative behaviour primarily due to maxmax behaviour of level-0 agent, where best responding to maxmax behaviour always ends up being more conservative than best-responding to diverse models by belief updating (as in robust and dlk(A)). This is reflected in the lower crash rate for the Qlk model compared to Robust. Equilibrium models lead to a reduced collision rate compared to the dlk(A) and the Robust models, likely a result of working under a complete information setting where there is no scope for misinterpreting the other agents' type. In terms of success rates, MBI also shows a lower success rate for all models. This is mainly because the situation (low distance gap between vehicles during game initiation) is setup such that for most models the optimal strategy profile is for the merging vehicle to wait

for the on lane vehicle to pass. In fact, the WMA dataset had one instance of MBI scenario and that resulted in failure. Similarly, collisions are also observed only in the MBI scenario.

Overall, whether or not an AV planner can succeed in their desired outcome depends on a variety of factors, such as, the assumption the vehicle and the human drivers hold over each other, the safety aspiration of each agent, as well as the specific state of the traffic situation. The analysis presented above helps in quantifying the relation between the desired outcome and the criteria under which it is possible.

5.6 Conclusion

This chapter developed a unifying framework for modelling human driving behaviour and strategic behaviour planning of AVs that supports heterogeneous models of strategic and non-strategic reasoning. The model consists of three layers of increasing ability for strategic reasoning, where each layer can hold multiple behaviour models. For the non-strategic layer, I also developed a model of level-0 behaviour for level-k type models through the use of automata strategies (dLk($\mathcal{A}$)) that is suitable as a non-strategic model for the context of modeling driving behaviour. The evaluation on two large naturalistic datasets shows that a combination of a rich level-0 behaviour can capture most of the driving behaviour as observed in the dataset. On the other hand, for the problem of behaviour planning, with the awareness that there can be different types of reasoners in the population, an approach of robust response to heterogeneous behaviour models is not only effective, but also is stable across a population of drivers with different levels of risk tolerance.

Chapter 6

Application: Rare event sampling and estimation

6.1 Introduction

With autonomous vehicles (AV) poised to change the transportation landscape, the ability of AVs to handle a wide range of human traffic behaviours safely and reliably is of paramount importance. To guarantee this, it is not adequate to rely solely on field tests as the primary method of AV evaluation, since the number of kilometres that needs to be driven for any statistical safety guarantee is prohibitively high [107]. Thus, there is an increasing role of simulation in all major components of an autonomous driving system (ADS), including perception, planning, testing, and verification. Although it is possible to significantly speed up the verification process in simulation, it is also necessary for simulation environments to be realistic. For the behaviour planner (which is the component in ADS responsible for tactical and high-level decision making), this means that simulation environments should be able to qualitatively model the behaviour of other traffic users in a way that is reflective of the real-world behaviour. Popular approaches design this behaviour in several ways: *expert-driven*, where designers programme the motion and behaviour of users [243], *data-driven*, where a model of behaviour is learnt from observations and naturalistic driving datasets [45], or a *hybrid* model that uses a combination of both [246] and[197]. Although it is possible to design models that learn from real-world data, a major challenge in any approach is the generation of unusual or atypical behaviour that is not readily observed in the data, such as crashes or near-miss scenarios.

In dynamic systems, rare event (RE) sampling provides a mathematical framework to analyse events of very low probability [186]. RE sampling techniques can be used to both

estimate the probability of occurrence of rare events and to generate the conditions that lead to rare events. In recent years, RE sampling based techniques have been used for simulation based verification and testing of a wide range of motion and behaviour planners. O'Kelly et al. use RE sampling for testing of planners that work in an end-to-end manner based on deep learning [169], whereas other approaches apply similar techniques to evaluate performance in specific traffic situations, such as lane changes and cut-ins [250]. Most approaches that use rare event sampling for AV evaluation use cross-entropy based importance sampling, which is an adaptive sampling technique to search for a sampling distribution that maximises the odds of leading to crashes and near-miss scenarios. Part of the uncertainty in traffic environments arises from the inherent stochastic behaviour of road users, as reflected in different driving styles of human drivers. This is in contrast to the design of motion and behaviour planners of an ADS, which optimize a set of defined objectives, such as, progress, safety, observance of traffic rules, etc. Therefore, applying the same driving algorithm that drives the subject autonomous vehicle to simulate the behaviour of other road users does not lead to the diversity needed for a proper safety validation methodology. In this chapter, I bridge this disconnect by applying the models of bounded rational behaviour developed in the previous chapters to sample both rare and typical situations of interest for testing an AV planner. First, I construct different behaviour categories based on the Quantal Best Response model to model stochastic traffic behaviour, and next, I develop a rare event sampling and optimisation method that provides greater interpretability to the generation of rare event situations compared to standard rare event sampling approaches that are based on safety surrogate metrics. The results show that categorising different driving behaviours and optimising for an appropriate driving policy can act as an effective technique for rare event estimation. I compare the approach with a crude Monte Carlo based method, as well as a baseline cross-entropy based approach [250], which has been shown to be effective for accelerated evaluation of ADS. The proposed method in the chapter shows better performance in key metrics such as the variance of the rare-event probability estimates and achieves 39% speed-up over cross entropy sampling and a speed-up to the order of 10^4 compared to crude Monte Carlo sampling. Further, I fit the behaviour model to a naturalistic driving dataset, and evaluate its use for the generation of new situations. The evaluation is based on vehicle cut-in events from the University of Michigan SPMD naturalistic driving dataset, which contains several hours of real world driving.

6.2 Background

6.2.1 Rare event sampling

Rare event sampling provides a framework for studying events of very low probability in dynamic systems with the primary goal of estimating their probability. In a dynamical system, the generalised dynamics of the system can be expressed as $X_{t+1} = \Phi(X_t, G_t)$, where $X_t \in \mathbf{R}^n, G_t \in \mathbf{R}^m$ are the state of the system and the input at time t, and Φ is the dynamics of the system. A rare event ϵ is defined using a scalar performance function η, and the rare event is the occurrence of the condition in which the function is greater than or equal to a specified threshold, $\eta(X_t) \geq b$. For convenience, rare events can also be defined over the input space, i.e, the set of inputs that leads the system trajectory to a critical state, as defined by $\eta(\cdot)$ and b. Thus, the probability of rare events can be expressed by the following integral.

$$p_\epsilon = \int I_\epsilon(g) p(g) dg \tag{6.1}$$

where $I_\epsilon(g)$ is the indicator function such that $I_\epsilon(g) = 1$ if $g \in \epsilon$, and $I_\epsilon(g) = 0$ otherwise. We assume that the outcome for ϵ is deterministic given g. In crude Monte Carlo methods, the above integral can be approximated by generating several independent and identically distributed samples[1] of the system inputs $G^{(0)}, G^{(1)}, .., G^{(N)}$ drawn from the distribution $p(g)$ as

$$p_\epsilon \approx p_\epsilon^{MCS} = \frac{1}{N} \sum_{i=1}^{N} I_\epsilon(G^{(i)}) \tag{6.2}$$

By the Central Limit Theorem, as $N \to \infty$, p_ϵ^{MCS} is distributed asymptotically as a Gaussian distribution with mean p_ϵ and variance $\sigma^2 = \frac{p_\epsilon(1-p_\epsilon)}{N}$. Since the above is only an estimate of the true probability, it is important to calculate the margin of error based on the number of samples. The relation between the sample size and the relative margin of error (re) of estimation for a confidence interval of 0.95 is given by the relation[2] $N > \frac{1.96}{re^2 p_\epsilon}$. Therefore, even for a relative error of 0.01 and a high rare event probability of 0.001, we need 10^7 samples. A major problem of crude Monte Carlo sampling is that reducing error variance is difficult; we need an prohibitively high number of samples, since most $I_\epsilon(G^{(i)})$s are 0 and only very few are 1. Importance sampling is a variance reduction technique that helps improve the accuracy of the estimate p_ϵ with fewer samples. Instead of drawing samples from $p(g)$, importance sampling uses a proposal distribution $q(g)$, where samples drawn from $q(g)$ have a higher probability of leading

[1] I use the convention where small g is the random variable and capital G is a sample. [2] Derivation in [186] page 4.

to rare event situations. Using $q(g)$, the integral in (6.1) can be written as $p_\epsilon = \int_\epsilon \frac{I_\epsilon(g)p(g)}{q(g)} q(g)dg$, which is the expectation $\mathbf{E}_q[\frac{I_\epsilon p}{q}]$. Following a similar approach to the crude Monte Carlo, the above integral can be approximated as

$$p_\epsilon \approx p_\epsilon^{IS} = \frac{1}{N}\sum_{i=1}^{N} I_\epsilon(G^{(i)})w(G^{(i)}) \tag{6.3}$$

where $w(G^{(i)}) = \frac{p(G^{(i)})}{q(G^{(i)})}$ is the *importance* weight of the sample $G^{(i)}$ or the *likelihood ratio*. Estimator 6.3 is an unbiased estimator with $\mu = p_\epsilon$ and $\sigma^2 = \frac{1}{N}(\mathbf{E}_q[\frac{I_\epsilon p^2}{q^2}] - p_\epsilon^2)$ under the condition that when $I(g)p(g) > 0$, $q(g) > 0$ is true. Thus, for every sample drawn from $p(\cdot)$ that leads to a rare event, the same sample should also lead to a rare event if it were drawn from $q(\cdot)$. Given a proposal distribution $q(g)$, the complete algorithm for estimating p_ϵ based on the IS technique is shown in Algorithm 4.

The most optimal choice for the proposal distribution $q(\cdot)$ is the original distribution conditioned on the rare event ϵ

$$q^*(\cdot) = p(g|\epsilon) = \frac{p(\epsilon|g)p(g)}{p_\epsilon} = \frac{I_\epsilon(g)p(g)}{p_\epsilon} \tag{6.4}$$

We can see that with this choice the variance of the estimator (6.3) reduces to zero, since any sample drawn from the distribution would have the value I_ϵ of 1 and the weight $w = p_\epsilon$. Thus, just a single sample can estimate the exact value of p_ϵ. In most cases, it is practically impossible to find the exact distribution $p(g|\epsilon)$ as it requires knowing p_ϵ. However, this optimal distribution gives an indication that a distribution close to $p(g|\epsilon)$ is a good proposal distribution.

One approach to generate the proposal distribution is based on cross-entropy (CE) method. If q is chosen from a family of distributions $\psi(\cdot, \theta)$, then the distance between the distribution $\psi(\cdot, \theta)$ and $p(g|\epsilon)$, measured by the Kullbeck-Leibler divergence, gives an estimate of the goodness of the proposal distribution. Thus, the optimal distribution can be found by solving the following optimisation problem:

$$\theta^* = \arg\min_\theta D_{KL}(\psi(\cdot, \theta), p(g|\epsilon)) \tag{6.5}$$

where D_{KL} is the KL divergence between the two distributions, and θ^* is the parameter of the optimal distribution. The CE approach gives a fast iterative scheme to find the solution to the above minimisation problem. The optimisation procedure can be further simplified by choosing $\psi(\cdot, \theta)$ as the exponential change of measure (ECM) function of the original distribution $p(g)$. Most approaches for rare event sampling in the AV literature approximate $p(g)$ with a heavy-tailed distribution from the exponential family. This enables CE optimisation to find a closed

form solution to (6.5), thus making the process much faster. Although the CE approach provides a point estimate of the probability of rare events, it provides little insight into the type of behaviour that leads to rare event situations. This becomes especially relevant for safety evaluation, where it is important to know what conditions and behaviours cause rare event situations.

Algorithm 4: Importance sampling algorithm for estimating probability of rare events.

Result: IS estimate $p_\epsilon := \frac{1}{N} \cdot \sum_{i=1}^{N} w_i$

foreach $i \in [0, .., N]$ **do**
  Sample an initial condition from the proposal distribution $g_i \sim q(g)$;
  Compute system trajectory based on the system dynamics :
  $\mathbf{X_i} = X_{t=0}, X_{t=1}, ..., X_{t=T}$;
  if $\max_{0 \leq i \leq T} \eta(\mathbf{X_i}) \geq b$ **then**
    Calculate the importance weight of the i^{th} particle $w_i := \frac{p(g_i)}{q(g_i)}$;
  end
end

6.2.2 Bounded rationality and utility alignment

The Quantal Best Response (QBR) model described in Sec. 2.3 is a model of suboptimal decision making, in which a boundedly rational agent, instead of always selecting the utility maximising action, makes cost-proportional errors. Although QBR was presented as a model of *response* to other agents' strategies, it can also be reformulated as a method of action selection that is independent of other agents' actions. In this formulation, the probability $P(a|s, \lambda)$ of taking a discrete action $a \in A$ in the environment state $s \in S$ is based on the utility $u : S \times A \rightarrow [-1, 1]$ as

$$p(a|s, \lambda) = \frac{\exp[\lambda \cdot u(s, a)]}{\sum_{\forall a} \exp[\lambda \cdot u(s, a)]} \tag{6.6}$$

The above formulation is a single agent view of QBR, and the only difference between the above formulation and the one in Sec. 2.3 is that the agent's utilities are expressed as a function of the environment state (s), rather than as a function of the actions of the other agents a_{-i} (a more specific semantic meaning of the states and actions considered for the scenario under study is presented in Section 6.3.1). Recall that when $\lambda \rightarrow \infty$, the behaviour converges to pure utility maximisation, i.e. the agent always takes the optimal action. Whereas $\lambda \rightarrow 0$ leads to a behaviour

of action selection based on a uniform distribution. The utilities in the standard QBR model is single objective, however, in our case, we build a model by considering multi-objective utilities. In the case of multi-objective utilities, an agent may have different levels of precision for each utility objective. Therefore, the precision parameter is a vector $\Lambda = [(\lambda_1, u_1), (\lambda_2, u_2).., (\lambda_k, u_k)]$, where each λ_i acts as the precision parameter for the corresponding utility $u_i \in U$, and U is the set of multi-objective utilities. The probability of action selection $f : S \times A \rightarrow [0, 1]$ in this multi-objective model can be formulated as the following mixture model in which $p(a|s, (\lambda_i, u_i)$ refers to the action selection based on Eq. 6.6 conditioned on the individual utility function $u_i \in U$.

$$f(a|s, \Lambda) = \frac{1}{k} \sum_{i=1}^{k} p(a|s, (\lambda_i, u_i)) \tag{6.7}$$

The main idea behind the methods developed in this chapter is that individual values of the precision parameter $\lambda_i \in \Lambda$ result in different degrees and categories of suboptimal behaviour, and some values of the vector Λ may lead to adverse outcomes, such as collisions more often than others. To broaden the scope of the behaviours that result from the different values of Λ, I also expand the acceptable range of individual precision parameters. Note that the typically acceptable range of λ is [0,∞], and the worst an agent can do with respect to optimality is a random selection of action ($\lambda = 0$). For other values, the agent's action selection is still in alignment with the utility, which means that the probability of selecting an action that fetches a higher utility is strictly higher than the one that fetches a lower utility, $u(s, a_i) > u(s, a'_i) \leftrightarrow P(a_i|s, \lambda) > P(a'_i|s, \lambda)$. In a typical scenario, it is understood that a driver will not willingly wish to crash into a vehicle and, therefore, if the utility is modelling, for example, safety, the condition should hold. However, for the kind of situation we want to model, this condition may not always hold. For example, consider the case of a distracted driver accelerating and crashing into another vehicle during a lane change manoeuvre. From a purely observational point of view, this behaviour is indistinguishable from the case where a driver wants to willingly crash into a vehicle. To generate these types of scenario, the range of individual values of the precision parameters λ in this model is in the range $[-\infty, \infty]$, in which a negative value allows a convenient modelling of behaviour that is completely misaligned with the predefined utilities.

Figure 6.1 shows the relation between the probability of an action $p(a|s)$ and its utility $u(s, a)$ at various levels of λ. A high value of $\lambda = 100$, skews the distribution $p(a|s)$ such that the policy prefers actions with maximum utility ($u(s, a) = 1$) with close to probability 1, i.e., a pure utility maximisation model. Whereas, lower values of $\lambda = 2, -2, -10$ progressively lead to a more sub-optimal policy. $\lambda = 0$ being the special case where actions are chosen based on a random policy. As I discuss in the next section, the flexibility of the bounded rationality model provided by the precision vector Λ helps model a wide range of naturalistic driving behaviours.

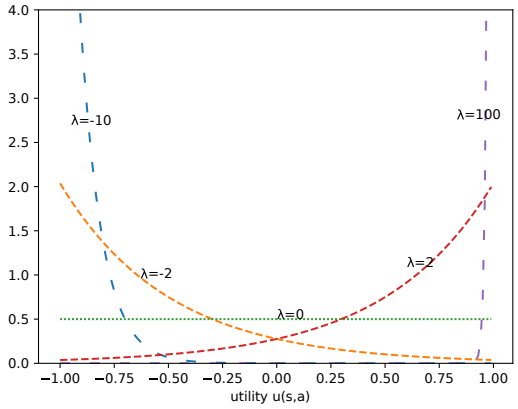


Figure 6.1: Probability distribution of an action a based on its utility $u(s, a)$. The plot shows the effect of the precision parameter λ on the probability (a higher λ leads to actions that have higher utility).

6.3 Rare event sampling and situation generation

6.3.1 Behavior categorisation

In this section, I use a typical vehicle cut-in scenario as a motivating example and show how the bounded rationality model developed in the previous section can be used to categorise a range of driving behaviours. A vehicle cut-in scenario (Figure 6.2) involves a vehicle (V_S) maintaining its lane of travel, while another vehicle (V_{LC}) performs a lane change manoeuvre into the lane of travel of V_S. I consider the case where V_S is driven in autonomous mode (subject vehicle) and V_{LC} is driven by a human (target vehicle). There are two conflict points, as marked by a cross in the figure — a side-to-side conflict that can result in sideways collision, and a sequential conflict that can result in rear-end collision.

As part of the cut-in manoeuvre, V_{LC} needs to decide on its target velocity $v^{LC}_{t+\Delta t}$, where $t + \Delta t$ is the time step when the front wheel of V_{LC} crosses the lane boundary of V_S's travel lane. V_{LC} also needs to decide the safety distance (distance gap) it has to keep from V_S, as measured by the difference between the vehicles' respective positions along the direction of travel $\delta = d(l^{LC}_{t+\Delta t}, l^{S}_{t+\Delta t})$. Based on these choices, V_{LC}'s cut-in behaviour can vary significantly. For example, V_{LC} can cut-in close to V_S with a low speed, or execute a high-speed manoeuvre maintaining a fair distance gap. Thus, the action space for V_{LC} for the manoeuvre consists of the tuple $(v^{LC}_{t+\Delta t}, \delta)$, and the state space consists of v^{S}_{t}. Once V_{LC} initiates the cut-in manoeuvre, the subject vehicle V_S needs to respond appropriately according to its behaviour decision logic,

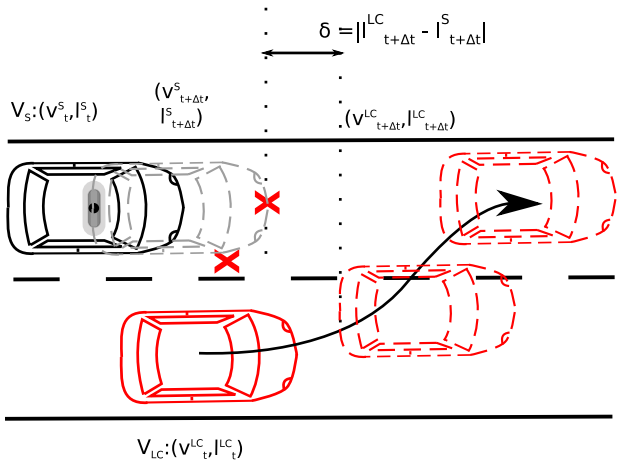


Figure 6.2: Vehicle cut-in scenario on a two-lane road. $v_t^{S,LC}$ are the velocities of the autonomous subject vehicle (V_S) and the target vehicle (V_{LC}) resp. at the start of the lane change manoeuvre. $v_{t+\Delta t}^{\{S,LC\}}$ and $l_{t+\Delta t}^{\{LC,S\}}$ are the velocities and locations when the front wheel of the target crosses the lane boundary. δ is the distance gap.

Table 6.1: Behavior categories based on the constraints on the values of the precision vector dimensions $\Lambda = [(\lambda_\delta, .), (\lambda_\tau, .), (\lambda_p, .)]$

Behavior id	$[\lambda_\delta, \lambda_\tau, \lambda_p]$	**Behavior description**
B1	$[-, -, +]$	cut-in with high speed at close distance with low ttc
B2	$[-, +, +]$	high speed at close distance with high ttc
B3	$[+, +, -]$	low speed at longer distance with high ttc
B4	$[+, -, -]$	low speed at longer distance with low ttc
B5	$[-, -, -]$	low speed at close distance with low ttc
B6	$[-, +, -]$	low speed at close distance with high ttc
B7	$[+, +, +]$	high speed at longer distance with high ttc
B8	$[+, -, +]$	high speed at longer distance with low ttc

which might include slowing down to maintain a safe distance gap or time-to-collision (ttc).

To model the behaviour of the target vehicle V_{LC}, I use three utility functions; two based on safety (u_δ,u_τ), measured by the distance gap and the time to collision (ttc), and one based on progress (u_p), measured by velocity ($v^{LC}_{t+\Delta t}$). $u_{\{\delta,\tau\}}(s,a) = 2S(2\alpha_{\{\delta,\tau\}}(s' - \{\delta^*,\tau^*\})) - 1$ and $u_p(s,a) = 2S(10(x-5)) - 1$, where S is the standard logistic sigmoid function, α is the slope parameter, δ, τ are the ttc and distance gap for the next state (s'), and δ^*, τ^*, v^* are the parameter values that are based on safe driving best practices. These functions belong to the general class of exponential utility functions, which is a popular class of utility functions used in decision theory [71]. The choice is also based on insights from [105], which shows that a driver's perception of risk level and their response have an exponential relation to critical vehicle and environmental states, such as getting close to an obstacle or curbside, and vehicle speed. Based on these utilities, the precision vector is $\Lambda = [(\lambda_\delta, u_\delta), (\lambda_\tau, u_\tau), (\lambda_p, u_p)]$. Following the equations in Section 6.2.2, every instance of the vector Λ generates a stochastic driving policy $f(v_S, (v^{LC}_{t+\Delta t}, \delta))$, and $\lambda_\delta, \lambda_\tau, \lambda_p$ control the level of adherence of the policy to each utility.

As shown in Table 6.1, based on the level of adherence, the values of λ in Λ can be grouped to form categories of driving behaviour. For example, the constraint $\lambda_\delta < 0, \lambda_\tau < 0, \lambda_p > 0$ leads to a driving policy that cares less about maintaining a safe distance-gap and time-to-collision, but more about making fast progress; shown in the table as the behaviour category B1. Thus, we get eight behaviour categories for a vehicle cut-in scenario, and even within a category, there is a wide range of individual driving policies sharing the common behaviour. We model the response of the subject vehicle V_S based on the Krauss car following model [120], which is activated at time step $t + \Delta T$.

6.3.2 Parameter optimisation

In this section, we develop an optimisation scheme and show how the behaviour model developed can be used for the purpose of rare event sampling. Revisiting (6.4), an optimal proposal distribution $q(\cdot)$ for importance sampling of rare events should be as close as possible to the distribution $p(g|\epsilon)$. In other words, the goal is to find a low variance estimator that has high probability in regions of the system input space g that lead to rare events. One way to achieve that is to find a driving policy that is more likely to lead to such events. To this end, we use the parameterized driving policy of (6.7) as the proposal distribution $q(\cdot)$. The system input space (g) is $\mathbf{R}^2_{>0}$ which consists of the velocity of the target vehicle and the distance gap. Thus, $q = f(g|\mathbf{\Lambda}^*)$, where $\mathbf{\Lambda}^*$ is the solution to the following optimisation problem $\mathbf{\Lambda}^* = \arg\max_{\mathbf{\Lambda}} I_\epsilon(g) f(g|\mathbf{\Lambda})$, where $I_\epsilon(g)$ is the indicator function for rare events. To solve the above optimisation problem, we use a Simulated Annealing (SA) based heuristic that first finds the category of behaviour (B1-8) that has a

high $I_\epsilon(g)$ and then subsequently finds a value of $\mathbf{\Lambda}$ within that behavior category that maximises the optimisation objective.
Algorithm 5 describes the optimisation procedure. The two main structures in the algorithm, $[p^{B1}_{max}, .., p^{B8}_{max}]$ and $[\Lambda^{B1}_{max}, .., \Lambda^{B8}_{max}]$ maintain the maximum probability of rare events for each behaviour category and the corresponding Λ of the driving policy that caused the rare events. There are two loops in the procedure, the outer loop iterates over all behaviours to find a behaviour with maximum rare event probability (line 5), and the inner loop iterates to find the Λ that maximises rare events within the behaviour category line (10). Following the standard technique in Simulated Annealing, the acceptance of a better solution (line 8 and 14) is controlled by the temperature parameters (T_{out}, T_{inn}), which are reduced by a constant factor in every iteration of the loop (line 18 and 21). The neighborhood generation of the outer loop (line 6) performs a weighted sampling of behaviour ids based on the current $[p^{B1}_{max}, .., p^{B8}_{max}]$ vector at each iteration. For the inner loop, the `sample()` method generates a value of Λ constrained by the behavior category based on a uniform distribution (line 11). `simulate_scene()` is the main entry point to simulate a set of situations with different initial conditions. In our implementation, we use the SUMO open source simulator to simulate the cut-in scenarios [118]. The method samples the initial state v^S_t from the distribution of the subject vehicle velocities observed in the naturalistic driving dataset, and runs N separate simulations where the behaviour of the target vehicle is sampled based on the driving policy $f(s, a|\Lambda)$. The algorithm outputs an estimate of Λ^*, which is subsequently used to construct the final driving policy to be used in the estimation of the rare event probability p^{IS}_ϵ based on the algorithm 4.

6.3.3 Situation generation

While the bounded rationality model can be used to provide a point estimate of the probability of rare events, the model can also be used to sample new situations of interest to evaluate the planner's performance in specific circumstances. One simple way to achieve this is by sampling the behaviours of other vehicles from the driving policy conditioned on a behaviour category. For example, to generate situations of high speed cut-ins at close distances (B1, B2), the behaviours can be sampled from the distribution $f(s, a|\Lambda_{\text{B1,B2}})$, where $\Lambda_{B1,B2}$ is the domain of Λ after applying the constraints of the respective behaviour categories (B1,B2) based on Table 6.1. Although this approach can sample a wide range of behaviours, a more effective technique can use a data-driven strategy consisting of the following steps: (i) acquisition of naturalistic driving data for the scenario under evaluation, (ii) fitting a behaviour model based on the data, and (iii) using the behaviour model to sample new situations that are not present in the dataset. In this section, we propose an approach to achieve the above objectives.

Compared to the problem of rare event sampling, where we optimise for a single value of Λ

Algorithm 5: Simulated Annealing (SA) based optimization procedure for Λ^*

Result: $\Lambda_{max}^{bid_{max}}$
$[\Lambda_{max}^{B1}, .., \Lambda_{max}^{B8}] \leftarrow$ init_$\Lambda()$
foreach $bid \in [B1, .., B8]$ **do**
  $p_{max}^{bid} \leftarrow$ simulate_scene(Λ_{max}^{bid}, N)
end
while $i < I_{max}$ **do**
  $bid \leftarrow$ weighted_sample$([p_{max}^{B1}, .., p_{max}^{B8}])$
  $p_{max} \leftarrow$ max$([p_{max}^{B1}, .., p_{max}^{B8}])$
  if $\exp\{((p_{max}^{bid} - p_{max})/T_{out})\}$ <*random(0,1)* **then**
    $j \leftarrow 0$
    while $j < J_{max}$ **do**
      $\Lambda \leftarrow$ sample(bid)
      $p\_\epsilon \leftarrow$ simulate_scene(Λ, N)
      if $p\_\epsilon > p_{max}^{bid}$ or $\exp\{((p\_\epsilon - p_{max}^{bid})/T_{inn})\}$ <*random(0,1)* **then**
        $p_{max}^{bid} \leftarrow p\_\epsilon$
        $\Lambda_{max}^{bid} \leftarrow \Lambda$
      end
      $T_{inn} \leftarrow$ temperature(j)
    end
  end
  $T_{out} \leftarrow$ temperature(i)
  $bid_{max} \leftarrow \arg\max_{bid} p_{max}^{bid}$
end

that maximises the probability of rare events, fitting the model to naturalistic data poses additional challenges. Naturalistic data are often multimodal in nature, i.e, they contain a mix of different driving behaviours, and thus a model fitted with a single value of Λ cannot adequately capture the variation. In order to resolve this problem, we apply insights from the behaviour categories developed earlier, and extend the bounded rationality based model for the more general setting of modelling mixed behaviours. We introduce three mixing parameters $A = \{\alpha_\delta, \alpha_\tau, \alpha_p\} \in \mathbf{R}^3_{[0,1]}$, one for each utility, and correspondingly extend the driving policy formulation of (6.7) to

$$f(s, a|\Lambda, A) = \frac{1}{k}\sum_{i=1}^{k} \alpha_i p(a|s, (\lambda_i^+, u_i)) + (1 - \alpha_i) p(a|s, (\lambda_i^-, u_i)) \tag{6.8}$$

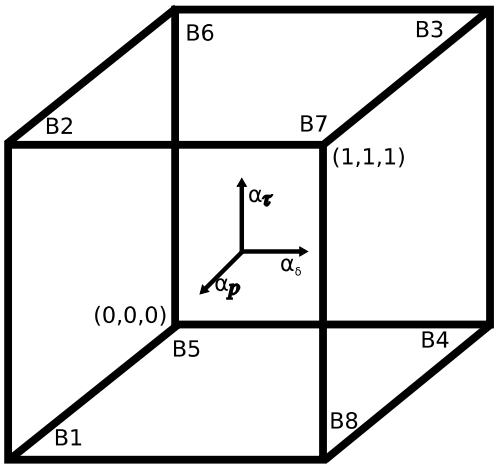


Figure 6.3: Behaviour model that mixes different categories of behaviours (B1-B8) based on the mixing parameter $A = \alpha_\delta, \alpha_\tau, \alpha_p$

where λ^+, λ^- are the positive and negative precision parameters, respectively. The distribution in the above model has multiple peaks and allows for mixing multiple behaviours [B1-B8]. A convenient way to imagine this is with a unit hypercube (Figure 6.3) where the corners are the behaviour categories from Table 6.1, and the parameter A controls the corresponding mix of behaviours. To fit the nine parameters $(\lambda_\delta^{+,-}, \lambda_\tau^{+,-}, \lambda_p^{+,-}, \alpha_\delta, \alpha_\tau, \alpha_p)$ to the observed data, we use least squares optimisation based on Trust Region Reflective algorithm since the parameters are bounded [141]. Finally, based on the fitted parameters, we use the behaviour model of (6.8) to sample new situations that are unseen in the original dataset.

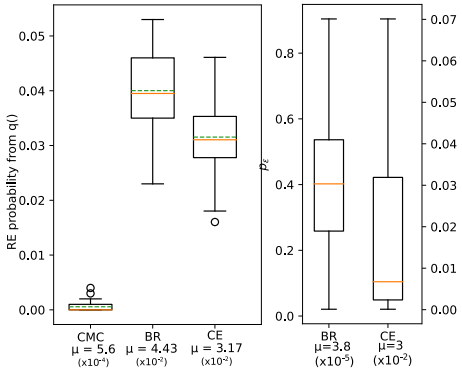
6.4 Experiments

In order to evaluate the bounded rationality based behaviour model for rare event sampling as well as situation generation, we use the University of Michigan SPMD (Safety Pilot Model Deployment) dataset. SPMD is one of the largest naturalistic driving datasets collected over two years, with 2842 equipped vehicles driving a total of 34.9 million miles. Part of the dataset with two months of driving data is publicly available in [1], and contains information recorded from the vehicle's data acquisition systems, such as MobilEye camera, CAN bus, and GPS. We follow the approach in [249] to extract 74,449 cut-in events recorded in the dataset, as well as the target and subject vehicle trajectory for 5 seconds immediately following the event. For our experiments, we define rare events as near-crash situations where the distance gap between the subject and the target vehicle is 0.01 metre or less and the subject vehicle is not stopped.

As part of the evaluation, we address two specific research questions based on the approaches developed earlier:[3]

- RQ1: How does behaviour-driven RE sampling compare to crude Monte Carlo and cross-entropy based methods?
- RQ2: How well does the bounded-rationality model fit the observed naturalistic driving data?

RQ1: Based on the theory presented in Section 6.2.1, we revisit that any system input sampled from the optimal proposal distribution $q^* = p(g|\epsilon)$ always leads the system to a rare event. A factor that helps judge the goodness of a proposal distribution is the probability of generation of rare events; the general intuition being that a higher probability is an indication of the distribution being closer to $p(g|\epsilon)$. Thus, we first compare the bounded rationality based model developed in the chapter (BR) with a proposal distribution based on surrogate metrics and Cross-Entropy optimisation (CE) from [250], as well as crude Monte Carlo sampling (CMC) on the basis of probability of generation of rare events.



(a) (b)

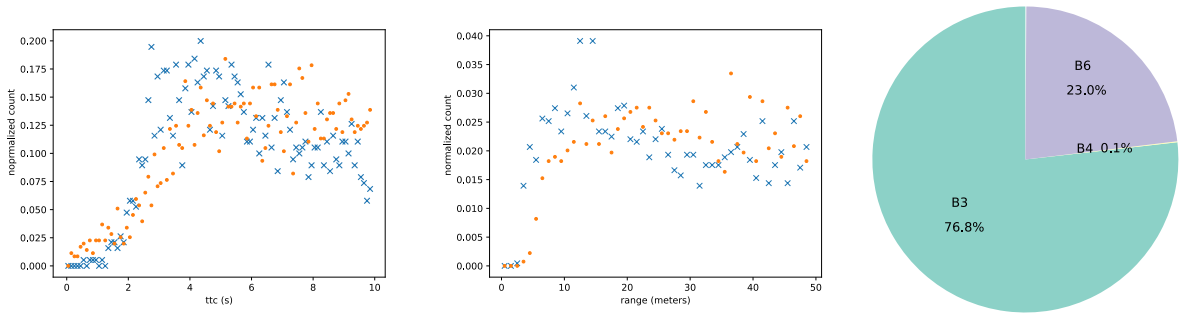
Figure 6.4: Comparison of probability of rare events based on proposal distribution $q()$ and estimates of p_ϵ after 100x1000 simulation runs. (a) Box plot of the probability of rare events based on $q()$ (b) Box plot of the p_ϵ estimates. Dotted and straight line shows the median and mean values respectively for box plots.

For both BR and CE, we run 100 iterations of the algorithm to optimise the parameters of the proposal distributions, and subsequently run another 100 iterations of 1000 simulations each with the final proposal distributions. For the BR approach, the final proposal distribution parameter

[3] Link to source code: https://bit.ly/2H83i1o

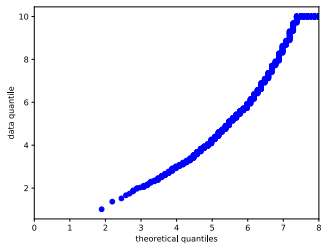
was $\Lambda = [\lambda_p = -6, \lambda_\tau = -71, \lambda_\delta = 6]$, with the distribution falling under the behaviour category B4. As shown in Figure 6.4a, the proposal distribution based on the bounded rationality model outperforms the cross-entropy-based model on the basis of the probability of RE when sampled from the respective proposal distributions. ($\mu = 4.43 \times 10^{-2}$ and $\mu = 3.17 \times 10^{-2}$ for BR and CE, respectively.) Since both BR and CE are biased distributions, they have a higher probability of generating rare events compared to crude Monte Carlo sampling ($\mu = 5.6 \times 10^{-4}$).

Along with the probability of occurrence of rare events, another important metric to evaluate the quality of the proposal distribution is the variance of the estimate p_ϵ. A variance closer to 0 is an indication of the proposal distribution being closer to the optimal distribution ($p(g|\epsilon)$). A lower variance also reduces the width of the confidence intervals of the p_ϵ^{IS} estimates, thus reducing the relative error between the estimate and the true probability p_ϵ. Figure 6.4b compares the variance of the estimates. As seen in the figure, the variance is significantly lower for BR compared to CE, suggesting that BR provides a more reliable estimate of p_ϵ compared to CE. Furthermore, one reason why the estimate for BR is lower compared to CE is because of a lower probability of the input samples in the original distribution $p(g)$. In other words, this means that the BR approach was better able to sample values that have a higher probability of leading to rare events, but a low probability of occurrence in the original distribution $p(g)$. This is an advantage of having a proposal distribution of different functional form, which means that input samples that have low probability can be sampled more easily. Thus, the results show that categorising based on the behaviours and optimising over them to find a crash-prone driving policy can act as an effective rare event sampling strategy. *RQ2:* As a part of the second research question, we evaluate how

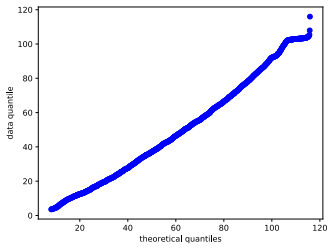


(a) (b) (c)

Figure 6.5: (a,b): Comparison of generated (blue cross) and naturalistic data (orange dots) for low speed cut-in situations (subject vehicle speed is less than 15 meters per second). x-axis: metric values (a: ttc (secs), b: range (meters)), y-axis: probability. (c) Distribution of behaviours for all situations in the dataset.



(a)



(b)

Figure 6.6: QQ plot for naturalistic data with respect to bounded rationality based behaviour model (a) ttc (b) distance-gap. x-axis: theoretical quantiles of the distribution for the behaviour model, y-axis: empirical quantiles from naturalistic data.

well the behavioural model based on bounded rationality can model naturalistic traffic data. To do this, we performed a random split (80-20) of the SPMD lane change dataset. We use 80% of the data to fit the parameters based on the approach discussed in Section 6.3.3, and we use the remaining data for evaluation. Since the distributions depend on the speed, we repeat our analysis for low (≤ 15 m/s), medium ($15 - 25$ m/s), and high speed (> 25 m/s) situations. Figures 6.5a,b show a visual comparison of the range and time-to-collision distribution of the generated and observed data for low speed situations, shown as blue crosses and orange dots, respectively. 6.5c also shows the distribution of the behaviour categories in the dataset. Since almost 70% of the events in the dataset were of low velocity, the distribution is tilted towards low speed behaviours. To evaluate the fit analytically, the QQ plots are shown in Figures 6.6a,6.6b. QQ plots are an effective tool to measure the fit of observed data to a theoretical distribution. The x-axis represents the quantiles of the distribution, the y-axis represents the observed data quantiles, and the blue dots map a quantile from the data to the distribution. As seen in the figure, we see a linear relation (Pearson correlation coefficient $\rho = 0.93, 0.98$ resp. for ttc and distance gap) between the two, which indicate that the observed data from the dataset is distributed according to the fitted bounded rationality based model.

6.5 Related work

Rare event sampling: There are a number of different approaches used for rare event sampling, including importance sampling, subset simulation, and splitting [16]. Subset simulation has been used in domains where the rare event probability p_ϵ can be expressed as a product of factors of higher probability, and the approach estimates the factor probabilities separately. Splitting is a related technique where the simulation makes iterative copies of the system state that leads to a state close to a rare event, and subsequently running simulations from that state [231]. Rare event sampling also has a rich history of application in various domains such as aerospace [25], systems biology [194], and telecommunication [24]. Blom et al. [25] apply splitting technique to the problem of safety verification for air traffic control in order to avoid rare events such as aircraft collision. Blanchet and Mandjes [24] apply an importance sampling based technique for queuing systems, and highlight the relevance of standard deviation of the estimate in a good IS proposal distribution. In the domain of autonomous vehicles, Zhao et al. use rare event sampling for accelerated evaluation of AV for lane change scenarios [250]. Kelly et al. use rare event sampling to test driving policies that are based on end-to-end learning [169]. Both approaches use cross-entropy based importance sampling as the simulation technique. To our knowledge, we present the first approach that highlights the importance of different driving behaviours for sampling rare events in the context of autonomous vehicles. *Behavior modelling:* Most previous approaches to modelling traffic behaviour are limited to the deterministic case, where vehicle behaviour was modelled as differential equations. Examples of such models include the Intelligent Driver Model [218], along with its extensions, such as the Newell car-following model [162]. When applied to the problem of ADS simulation, these approaches are limited in their ability to model the variation of human traffic behaviour, including positive, negative, and edge case behaviour. In the broader field of behaviour modelling, there is an extensive body of literature on modelling and simulation of pedestrian behaviour under varied situations. Popular approaches use variations of the Social Forces Model (SFM) [93], where the behaviour of agents is modelled as a dynamical system containing attractive and repulsive forces, and the final behaviour is the result of all such forces acting on the agent. Although SFM provides an intuitive modelling paradigm to model agent movement, it has been shown to be difficult to calibrate the models to real empirical data due to the forces not being linearly additive in nature. To address the shortcomings of the social forces model, potential-based methods follow an agent-free model, where the behaviour is not modelled individually for every agent, as in SFM. Instead, potential-based methods treat goals and obstructions as a continuous potential field, and the resulting behaviour is the solution to the energy minimisation problem in the field. Potential field based methods can be considered to be a special case of utility-based methods. However, like most utility-based methods, potential field models work under the assumption that the behaviour always follows the optimal path. We con-

sider this assumption to be restrictive and address this using bounded rationality in our approach. Schmidt uses the notion of bounded rationality through prospect balancing theory to study driver speed choices [199]. In our work, we use bounded rationality for lane change behaviours and demonstrate methods to apply the models for verification and testing of ADS. Yang and Peng [239] develop an errable driver model to model suboptimal driving behaviours, including distraction and perceptual errors. Compared to the errable model, our approach is based on a more general utility-driven framework, and thus can be applied to a wider variety of driving situations.

6.6 Conclusion

In this chapter, I develop a model of driver behaviour based on the Quantal Best Response model of suboptimal decision making. I apply the behaviour model to two cases: (i) generation of rare event situations and estimating the probability of rare events, and (ii) applying the behaviour model to generate new synthetic data for testing behaviour planners. I evaluate the proposed model based on a large naturalistic dataset and show that bounded rationality based behaviour model can improve crude Monte Carlo sampling by an order of 10^4. Compared to an approach of sampling surrogate safety metrics and cross-entropy optimisation, the performance of the new IS proposal distribution model provides a 39% speedup of estimation and a reduction in variance by an order of 10^2. I also show that the synthetic data sampled from the developed behaviour model has a strong correlation with the naturalistic driving data.

Chapter 7

Application: Dynamic occlusion safety validation

7.1 Introduction

Safety validation of autonomous vehicle (AV) planners is a critical component in the development of AVs. In recent years, as AVs face the challenge of sharing the roads with other human drivers with diverse behaviour, the problem of behaviour planning for AVs has taken a multi-agent view with the use of game-theoretic models for planning [136, 215, 216, 140, 85, 70]. Such planners, referred to in this chapter as *strategic* planners, view other road users in the vicinity as agents playing a game, and the AV chooses an action based on a game solution, for example a Nash equilibrium. Such models have been shown to be effective in simulation, and also have been evaluated against naturalistic human driving behaviour [196, 210]. On the other hand, evaluation of safety is also arguably a multi-agent problem, based on the idea that the outcome of a traffic situation depends on the assumptions traffic agents have of each other as well as the collective behaviour based on those assumptions. For example, it is clear that two vehicles deciding to cross an intersection at the same time result in a higher risk than one vehicle deciding to wait for the other. In order to perform a safety assessment of strategic planners, a safety validation framework needs to be aware that the AV planner places certain assumptions on the model of behaviour of other drivers, and therefore ideally focus the safety assessment on scenarios where such assumptions are likely to break down. Since most existing game theoretic planners work under the assumption that each vehicle in the game is aware of all other vehicles in traffic, occlusions are prime avenues of such high risk situations.

Occlusions or obstructed views, where a road user's view is obstructed due to static structures

(trees, buildings, etc.), other vehicles in traffic, and elevation and geometry of the road, are the leading cause of traffic accidents [2]. Most of the risk assessment due to occlusion has focused on *static* occlusion, i.e., occlusions that are caused by static structures, such as trees, buildings, parked cars, etc. [242, 150, 58]. On the other hand, situations of *dynamic* occlusion, i.e., occlusion caused by another vehicle in traffic, have unique challenges and can appear unexpectedly at any moment in traffic. For the problem of planning under dynamic occlusions, existing methods have been proposed with a single-agent view (i.e., without taking into account collective behaviour) [96, 143, 30]; however, to our knowledge, there are no safety validation frameworks for the evaluation of strategic planners for the problem. Given the criticality of dynamic occlusion scenarios [2, 47], it is necessary to address this gap.

A major challenge of dealing with dynamic occlusions is that they are transient in nature, i.e., they can occur in any traffic situation based on certain alignments of three vehicles. Therefore, unlike the case of static occlusions, it is not possible to leverage prior information about structures, such as buildings and trees, from high-definition (HD) maps and incorporate that information in the assessment of occlusion risk. The transient nature also means that the scope of the problem is much bigger, with infinite possibilities of dynamic occlusions that cause potentially risky situations that can arise in traffic. Therefore, existing repositories of observational data drawn from naturalistic driving are not adequate to achieve complete coverage of all dynamic occlusion situations an AV will face in its operational lifetime. To address the problem at scale, following the principle of a scenario-based accelerated evaluation [250, 182], an ideal approach should generate realistic and critical simulation test scenarios that can effectively assess the safety based on situations that go beyond the set of observed scenarios in existing data. This chapter addresses the aforementioned challenges by making the following contributions. i) A novel *planner-in-the-loop* (white box) safety validation framework for strategic planners using the theory of *hypergames* ii) A multi-agent dynamic occlusion risk (DOR) measure for assessing situational risk in dynamic occlusion scenarios, and iii) a search-based method to augment naturalistic data with realistic dynamic occlusion scenarios using vehicle injection.

While dynamic occlusion can occur anywhere while driving, occlusions tend to occur at multi-lane intersections where there is ample opportunity for vehicles to block each other's view, and vehicles are often moving along paths that intersect. In fact, the National Motor Vehicle Crash Causation Survey found that 7.8% of all intersection-related collisions were caused by the driver's incorrect decision to turn with an obstructed view of traffic [47]. Therefore, we demonstrate the efficacy of the approach with the help of experiments conducted on a large naturalistic dataset from a busy traffic intersection, and show that our proposed validation method achieves a 4000% gain in generating occlusion causing crashes compared to naturalistic data only, along with a more diverse coverage, and ability to generate commonly observed dynamic occlusion crashes in traffic that are beyond the naturalistic data used as input in an automated

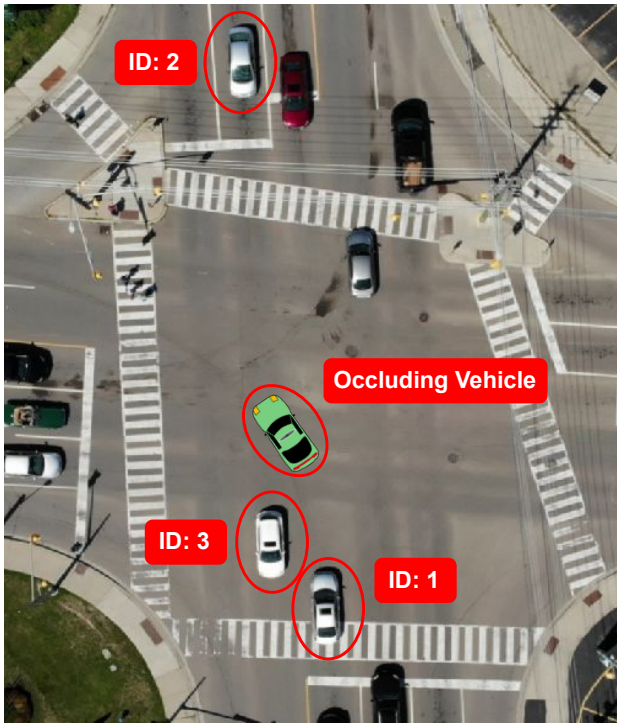
manner.

7.2 Background

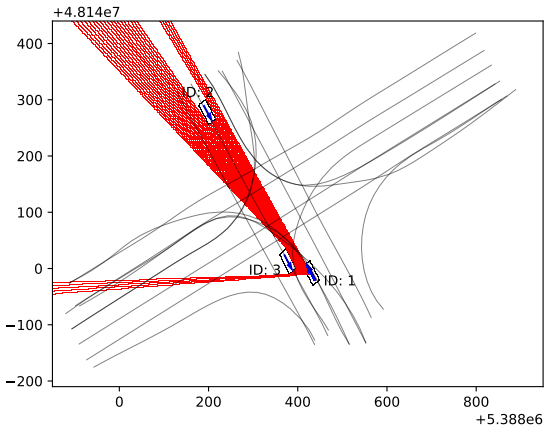
Strategic planners for AV planning use standard models of game theory with the assumption that all agents have a common view of the game, including the set of available actions to each player, the utility of each action for every agent, collectively are part of *common-knowledge* in game-theoretic terminology [82]. However, due to factors such as occlusion, distraction, and inattention, a driver may not be aware of the presence of another conflicting vehicle and thereby have a different view of the game than other vehicles in the vicinity. The hypergame framework provides a formal model of interaction for such scenarios where the strict assumption of a common view of the game breaks down [19, 116, 226].

In a standard formulation of a game, all agents play a common game $G = (N, A, U)$, where N is the number of agents in the game indexed by i, $A = \prod_{\forall i \in N} A_i$ is the set of actions available to all agents and $U : A \rightarrow R^N$ are the utilities that map a set of actions of every agent (a strategy) to a real vector R^N, and U_i is the i^{th} component of the vector representing the utility of the strategy to player i. In the hypergame framework, instead of agents playing a common game G, they play *hypergames* (H). The hypergames $H = \{H^0, H^1, .., H^L\}$ are organised in levels of game hierarchy, where at higher levels, agents have greater awareness of the view of other agents of the game that may not match their own. The level-0 hypergame is the common singular game G, i.e., $H^0 = G$, where all agents share the common view and play the common game. At level 1, players have different views of the game, i.e., $H^1 = \{G_1^1, G_2^1, .., G_N^1\}$, where $G_i^1 = (N^1, A^1, U^1)_i$ is the i^{th} agent's view of the game, where they may have a completely different view of the number of agents, the actions and utilities of the game relative to other agents' view G_{-i}^1 (where $-i$ represents any other agent). At level 2, agents not only have their own individual view of the game, but also awareness that other agents may have their own different views; therefore, the level-2 hypergame $H^2 = \{H_1^1, H_2^1, .., H_N^1\}$, where each agent i's view of the level-1 hypergame, H_i^1, together forms H^2. Continuing up the hierarchy, the hypergame model can be extended to a finite level L; however, we focus up to level-2 in our analysis, as that level is sufficient to model risk arising from conflicts due to dynamic occlusion scenarios in traffic.

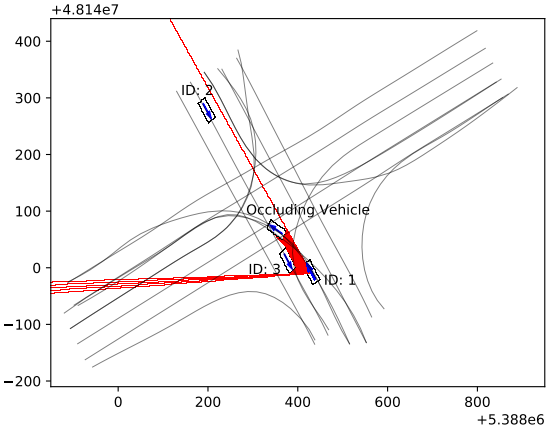
In this chapter, we use the above theory to construct three distinct perspectives of a traffic situation with respect to dynamic occlusion: level-0 is *occlusion resolved*, where all agents have a common omniscient view of the traffic situation (i.e., as if vehicles were transparent objects), level-1 is *occlusion naive*, i.e., drivers ignore occluded spaces, and level-2 is *occlusion aware*, i.e., drivers are aware of the occlusions in the situation and can have their own subjective process



(a)



(b)



(c)

Figure 7.1: A left turn across path (LTAP) scenario from the WMA database. (a) Real traffic footage along with the synthetic OV injected. (b) occlusion check without the OV; 2 is not occluded from 1. (c) occlusion checking with the OV; 2 is occluded from 1. The road lines in (b) and (c) represent the centerlines of each lane.

of resolving these occlusions and incorporating that awareness in their planning. We also assume a setting where only one vehicle in the scene is occlusion-aware, such as an AV under test, and the rest are occlusion-naive human drivers. Therefore, $H^2 = \{H^1\}$ is a singleton set where the level-1 hypergame H^1 is constructed from the sole occlusion-aware vehicle's perspective.

A solution concept provides a solution to a game, which results in a strategy profile, i.e., a set of trajectories that every vehicle executes. Some of the solution concepts proposed in the literature for strategic planning in AVs include Nash equilibrium [179, 201, 85, 196], Stackelberg equilibrium [70], the Qlk model [215, 216, 136, 196], and Pareto optimality [210].

The games are instantiated as simultaneous move games based on the joint system state $X_t = \prod_{\forall i \in N} X_{i,t}$, of N vehicles in traffic at time t. $X_{i,t} = [x, y, v_x, v_y, \dot{v_x}, \dot{v_y}, \theta]$ are location coordinates (x, y) on R^2, lateral and longitudinal velocity (v_x, v_y) in the body frame, acceleration $(\dot{v_x}, \dot{v_y})$, and yaw (θ) of a vehicle i at time t. We use the notation $\sigma(X_t, G)$ for a strategy profile for all players in the game G played in state X_t, with an additional subscript σ_i to refer to vehicle i's strategy in the strategy profile. The actions in the game are cubic spline trajectories [109, 6], generated over a planning horizon of 6 secs. The utilities U_i in the game are multi-objective, with two components: safety (a sigmoidal function that maps the minimum distance gap between vehicle trajectories into the interval [-1,1]), and progress (a linear function that maps the length of the trajectory in metres into the interval [0,1]). The two objectives are combined using a lexicographic thresholding parameter (γ) [135]. Following [70, 196], we also use a hierarchical decomposition of the actions of the game, where trajectories are generated based on high-level manoeuvres. The construction of the game is the same as that in Chapter 3. Since our validation method is white-box, we need to choose a solution concept a strategic planner uses. We choose the Nash equilibrium (for the level of manoeuvres) due to its ubiquity [179, 201, 85, 196] along with maxmax (for the level of trajectories) for its promise of being able to model naturalistic human driving behaviour better than others [196]. However, we note that the method is planner agnostic and can be extended to any strategic planner with just the knowledge of the solution concept.

7.3 Dynamic occlusion

Occlusions are caused by specific spatial alignments between at least three vehicles such that one vehicle is obstructed from the view of another vehicle. Let $O(i, j, k) \in \{0, 1\}$ be an indicator function that represents an occlusion. It has value 1 when vehicle k is occluded from vehicle i's view by an *occluding* vehicle (OV), j. For a traffic scenario with N vehicles, if $O(i, j, k) = 0; \forall i, j, k \in N$, then the maximum level of the hypergame played by N agents is 0. This follows

trivially from the observation that there are no occlusions, and therefore each agent is aware of every other agent in the scene and plays the common game $H^0 = G$. However, if there is a set of vehicles for which $O(i, j, k) = 1$, there is an occlusion in the scenario, and therefore, on account of i having a different view of the game, the minimum level of hypergame that the agents are playing is at least 1. Fig. 7.1a illustrates a scenario where $O(1, \text{OV}, 2) = 1$ and $O(2, \text{OV}, 1) = 1$, i.e., vehicle 1 and 2 are occluded from each other by vehicle OV.[1]

Purely the presence of an occlusion does not necessarily lead to a collision; for example, in the same illustrative example, regardless of the view 1 has of the game, if the solution of that game is such that 1 waits for the OV to cross, then that is a much safer outcome compared to a solution where 1 decides to follow the OV across 2's path and 2 decides to accelerate assuming the OV will cross the intersection in time (recall that 1 and 2 are not aware of each other). Therefore, we develop our risk estimation with a *planner-in-the-loop* approach that takes into account, along with the traffic situation, the dynamic behaviour of all the involved vehicles given by the strategic model used by the AV planner. Next, we present this notion more formally.

An occlusion-aware perspective is the level-1 hypergame $H^1 = \{G_1^1, G_2^1, .., G_N^1\}$ that represents the individual games that occlusion-naive vehicles play. A Dynamic Occlusion Risk (DOR) is a measure of relative risk based on that hypergame and traffic state X_t as follows.

$$\begin{aligned} DOR(X_t, H^1, \mathcal{S}) = &\mathcal{S}(\underbrace{\sigma^*(X_t, H^0)}_{\text{occlusion-resolved strategy}}) \\ &- \mathcal{S}(\underbrace{\prod_{\forall i \in N} \sigma_i(X_{i,t} \cdot \hat{X}_t^i, G_i^1)}_{\text{occlusion naive strategies}}) \end{aligned} \tag{7.1}$$

where $\hat{X}_t^i = \{X_{y,t} : O(i, x, y) = 0; \forall x \in N, \forall y \in N \setminus x\}$, or the game state constructed from the vehicles that an occlusion naive vehicle can see. The first component of the equation is the safety, with respect to a surrogate metric $\mathcal{S}$ (we use minimum distance gap), of the occlusion-resolved game. This reflects the situation in which all vehicles see each other and follow the equilibrium strategy. However, an occlusion-naive vehicle i will instead solve the game G_i^1 and construct occlusion-naive strategies. Comparing that strategy with the solution of the occlusion-resolved game (H^0) estimates the relative risk arising from dynamic occlusion. The above measure is calculated based on one occlusion-aware AV's level-1 perspective (H^1). However, if there are multiple AVs in the scenario with their own level-1 perspectives, the measure can be calculated from each of those perspectives. The perspectives could be coordinated via V2X and the estimated risk could be incorporated into planning.

[1] The function O is asymmetric since occlusion depends on the position and orientation of the sensors on each vehicle.

7.4 Method

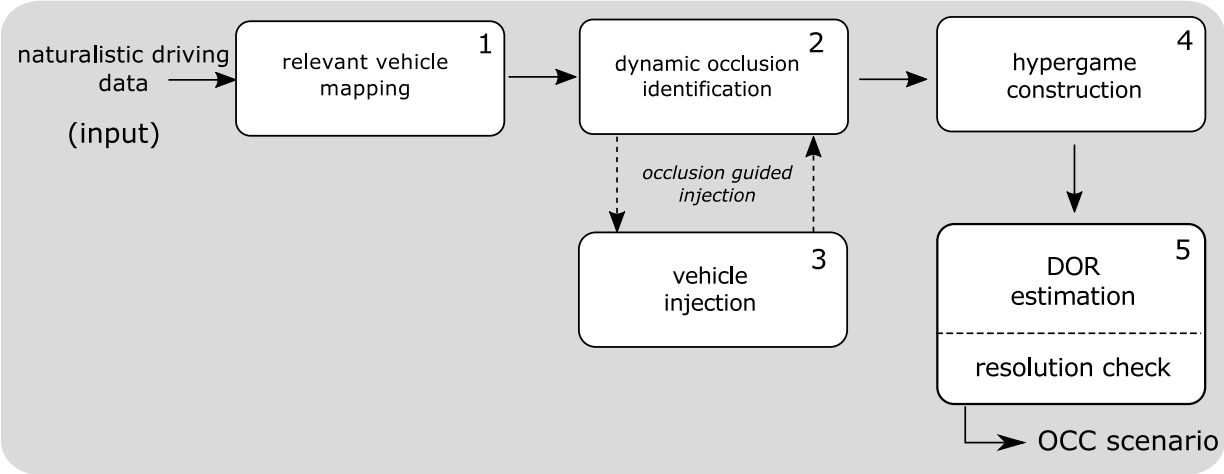


Figure 7.2: Schematic representation of accelerated scenario-based dynamic occlusion safety validation method

Fig. 7.2 shows a schematic representation of our proposed method. A scenario-based accelerated evaluation method [250, 182] generally has three key components, all of which are covered in our approach — a) being able to draw scenarios from naturalistic driving data (input, step 1), b) guided generation or sampling of novel test scenarios based on a target condition (steps 2-3), and c) identification of risk based on a quantifiable measure and assessment of the component under test through the measure (steps 4-5). Next, we describe each of the steps in more detail.

7.4.1 Naturalistic data and Relevant vehicle mapping.

From all situations in naturalistic driving data (we use the intersection dataset from the WMA database [196]), we extract scenarios corresponding to the two main intersection navigation tasks, namely, left turn across the path (LTAP) (Fig. 7.1a) and unprotected right turn (RT). For each of these scenarios, we select the vehicle that executes the scenario at a given time and call it the *subject* vehicle. With respect to the subject vehicle, we construct the set of *relevant* vehicles by including i) any vehicle that is in cross-path conflict with the subject vehicle, ii) the leading vehicle of the subject vehicle, and iii) the leading vehicle of a cross-path conflicting vehicle. The set of relevant vehicles along with the subject vehicle forms the set of vehicles N, the system

Algorithm 6: Occlusion-guided random search for a naturalistic scene with N vehicles.

$D^{\mathcal{O}} \leftarrow \{\emptyset\}$
for $s \in D$ **do**
  $V^{\mathcal{I}} \leftarrow$ occluding_vehicle_sampling(s)
  for $o \in V^{\mathcal{I}}$ **do**
    for $i \in N$ **do**
      if $\exists x \in N \setminus i; O(i, o, x) = 1$ **then**
        $s \leftarrow$ add_to_scene(o)
        $D^{\mathcal{O}} \leftarrow D^{\mathcal{O}} + \{s\}$
      end
    end
  end
end
return $D^{\mathcal{O}}$

state X_t at time step t, and represents a situation. The set D is the set of all such situations in the dataset and represents important interactions based on traffic conflicts, and we use the set to create the games/hypergames in the subsequent steps. The notion of *subject* and *relevant* vehicle is used here as a way of isolating the LTAP and RT scenarios from the input data set and does not have any special meaning outside of this context. As a minimal example of this construction, in the snapshot shown in Fig. 7.1a, 1 is a subject vehicle corresponding to an LTAP scenario; the relevant vehicles is the set {2,3}, since 2 is in cross-path conflict with 1, and 3 is the leading vehicle of 2.

7.4.2 Dynamic occlusion identification and vehicle injection

The goal of this stage is twofold; first, to implement an occlusion check process that identifies whether there is a dynamic occlusion in a given traffic scene (step 2), or in other words, implement the occlusion indicator function O, and second, augment naturalistic data with such scenes by injecting vehicles in realistic configurations (step 3).

To achieve the first goal, we use a voxel-based raycasting approach [8]. We plot the raycasts and vehicle bounding boxes (size l by w) using an occupancy grid map with a grid size of c. A vehicle is considered occluded if fewer than ϵ rays collide with its bounding box. A particular challenge of modelling driver vision is modelling driver attention, since *where* a driver is looking directly influences what they can and cannot see. We use a distance-based approach as a model of driver attention, whereby closer vehicles receive more attention than vehicles farther away.

Algorithm 7: DOR identification algorithm.

```
D^C ← {∅} ;                                              // Initialize OCC set
for s ∈ D^O do
    H^1 ← {∅} ;                                          // initialize hypergame set
    X_t ← states(s) ;                                    // construct vehicle states
    for i ∈ N do
        N_i ← N
        for j ∈ N \ i do
            if ∃x : O(i, x, j) = 1 then
                N_i ← N_i − {j} ;                        // Remove occluded vehicle
            end
        end
        G_i^1 ← (N_i^1, A_{N_i}^1, U_{N_i}^1) ;          // occlusion-naive game
        H^1 ← H^1 + {G_i^1} ;                            // Add it to the hypergame
    end
    if DOR(X_t, H^1, S) ⩾ θ ;                            // Collision check
     then
        D^C ← D^C + {s} ;                                // Add to OCC
    end
end
return D^C
```

The intuition behind this is that human drivers pay more attention to vehicles that are nearby, since these vehicles are more relevant to the driver's decision-making process. This translates to closer vehicles receiving a larger number of raycasts than vehicles farther away. Figs. 1b,c are examples of our occlusion-checking approach applied to a LTAP scenario with and without a synthetic occluding vehicle.

Algo. 6 achieves the second goal (step 3). Since dynamic occlusion can occur anywhere in the intersection, we sample configurations of potential occluding vehicles that can be realistically injected into the scene (ln.,3, *occluding_vehicle_sampling*). We use a grid-based sampling with a resolution of d metres along the lane centerline, and place vehicles ensuring minimum distance gap with existing vehicles. For velocities, we sample from a distribution based on the naturalistic dataset. Step 3 iterates over the samples of injected vehicles, $V^{\mathcal{I}}$, and if the injected vehicle causes an occlusion, the vehicle's configuration is added to the scene s (ln. 7, *add_to_scene*), and the scene is added to the list of occlusion scenes $D^{\mathcal{O}}$ (ln. 8).

7.4.3 Hypergame construction and DOR estimation

Step 5 outputs the set of occlusion-caused collision (OCC) scenarios, represented by their initial scenes $D^{\mathcal{C}} \subseteq D^{\mathcal{O}}$. It does so by selecting the scenes from $D^{\mathcal{O}}$ that lead to collisions based on the DOR measure. For each scene in $D^{\mathcal{O}}$, Algo. 7 constructs the level-1 hypergame based on what each occlusion-naive vehicle sees and plays (G_i^1). If the DOR measure from the hypergame results in a collision, the corresponding scene is added to the OCC set $D^{\mathcal{C}}$.

In order to make dynamic occlusion situations more realistic, we finish Step 5 with a resolution check (Fig. 7.2). We run the game starting from state X_t in simulation. If, right after the game initialisation, the occlusion is resolved and an emergency manoeuvre by any of the vehicles avoids a crash, we exclude that situation from $D^{\mathcal{C}}$.

7.5 Experiments and evaluation

To evaluate and demonstrate the efficacy of our approach, we use the Waterloo Multi-Agent (WMA) database recorded at a busy Canadian intersection with over 3.5k vehicles. We evaluate our approach by comparing its output with the dynamic occlusion and risk identified directly from naturalistic data, that is, without injecting additional occluders. We also evaluate the realism of the occlusions and crashes identified by our method relative to ones commonly seen in traffic and provide a severity analysis of the identified crashes. For the experiments, we used the following parameter values: $l = 4.1m$, $w = 1.8m$, $c = 0.1m$, and $\epsilon = 3$.

Comparison with naturalistic data. We first compare the number and variety of occlusion-caused collisions (OCC) that were generated based on our approach with occlusion injection and the OCCs generated purely from the naturalistic dataset. As seen in Table 7.1, using a non-augmented approach purely based on the naturalistic dataset, we can generate only 2 OCCs, whereas using the proposed accelerated approach allows us to increase the total number of OCCs

	Naturalistic data	Our approach
No. of OCC	2	80
No. of dynamic occlusion scenarios	1534	105,914

Table 7.1: Comparison with validation from only naturalistic data.

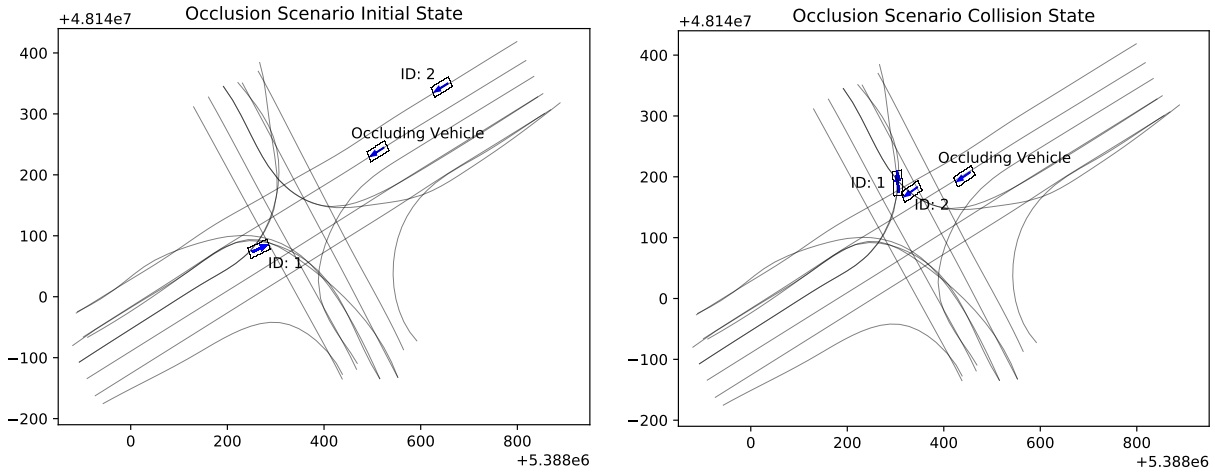
by 4000%, to a total of 80 OCCs. Similar gains in manifold (7×10^4) are also seen in the number of dynamic occlusion scenarios generated by our method.

In addition to faster generation of OCCs, our method also shows a greater coverage of OCC configurations. Figures 7.5a and 7.5d show the positions of the occluding vehicles in the 80 OCCs generated using the augmented dataset and the 2 OCCs generated from the nonaugmented dataset, respectively. Similarly, figures 7.5c and 7.5d show the positions of each vehicle involved in an OCC at the moment of impact for the augmented and nonaugmented dataset, respectively. The figures show that our approach achieved a much more even and wider coverage of the dynamic occlusion situations compared to the one observed from naturalistic data.

Diversity of generated scenarios. We were also able to generate a diverse range of OCCs. Among the 80 OCCs, we found 53 *front-to-front* collisions, 21 *angle* collisions, 4 *sideswipe* collisions, and 2 *front-to-rear* collisions (refer to [3] for details of these types). Many of these collisions can be attributed to "tagging on" behaviour, which is characterised by a vehicle following behind a left-turning occluding vehicle and not noticing that there is an oncoming vehicle. This behaviour is typically caused by the follower vehicle making the impatient decision to proceed with the left-turn even though they do not know if there are oncoming vehicles. In fact, 69 out of the 80 OCCs possess this characteristic. This "tagging on" is an occlusion-naive behaviour, and is observed to be a common cause of crashes due to dynamic occlusion in the real world (https://youtu.be/jDEZ-igoDgw?t=629). The same behaviour was also illustrated earlier in Fig. 7.1a, where vehicle 1 performs this "tagging on" behaviour which results in a collision with vehicle 2.

11 out of the 80 OCCs do not possess this "tagging on" behaviour. However, in the case of LTAP scenarios, the collision is also caused by a similar occlusion naive behaviour of the left-turning vehicle, proceeding without having vision of potential oncoming vehicles. Figure 7.3a is an example of this situation. Here, vehicle 1 executes a left-turn without having vision of vehicle 2. The result is an "angle" collision between both vehicles. This type of occlusion scenario configuration is also commonly seen in real world situations (https://youtu.be/tDN-mwNSJc8?t=43, https://youtu.be/Qk7ejm8MlEc?t=198).

Analyzing the locations of the occlusions, Figure 7.5a shows that occluding vehicles tend to be positioned close to the center of the intersection (and along their respective straight-through path for LTAP scenarios). However, for RT scenarios, our results suggest that the occluding vehicle must be positioned at a particular location—at the end of the SW left-turn lane for (WS,NS) OCCs or at the end of the ES left-turn lane for (SE,WE) OCCs. The large clusters in Figure 7.5c show that OCCs tend to occur as one of the colliding vehicles is in the process of completing its left-turn and the other colliding vehicle is crossing straight through the intersection. This demonstrates that our proposed method not only has high diversity in the type of risky situations



(a) (b)

Figure 7.4: (a) 1 begins making a left-turn while 2 proceeds through the intersection. Both vehicles are occluded from each other. (b) 1 collides with 2.

it can identify, but can also generate common situations observed in daily traffic in an automated manner.

Severity analysis. Our results also allow for an analysis on how *severe* OCCs tend to be. Figure 7.7a shows the distribution of the 4 severity classes across the 80 synthetic OCCs. We calculate severity of an OCC by extracting the relative velocity between the colliding vehicles and mapping the value based on the ranges from injury models in [119, 79]. The severity class mapping is as follows: S0:[0,5.3], S1:(5.3,7.7], S2:(7.8,10.3], S3:$\geqslant$10.3, all in ms^{-1}. We use relative velocity as a worst-case assumption instead of Δv (where Δv for a light vehicle approaches the relative velocity between colliding vehicles when the other vehicle in the collision is much heavier, such as a truck). That the severity distribution is skewed towards S3 is not surprising. Out of the 80 OCCs, 78 were LTAP scenarios, which most often lead to front-to-front collisions, resulting in the highest relative impact speeds. In addition to the high likelihood of a severe collision, OCCs are dangerous because they allow for little time for either driver to respond to the situation. Figure 7.7b shows the distribution of durations from the moment both colliding vehicles are no longer occluded to the moment of impact, across the 80 synthetic OCCs. The distribution ranges from 0.5s to 3.0s with a mean value of 1.65s. The range of driver response time is typically between 0.8s to 2.5s with a mean response time between 1.3s to 1.5s [63, 31, 130]. Based on the average response times then, drivers only have between 0.15s and 0.35s to decelerate or perform an evasive maneuver before a collision. Therefore, proactively identifying potential occlusion situations *before* they occur is critical for ensuring driver safety.

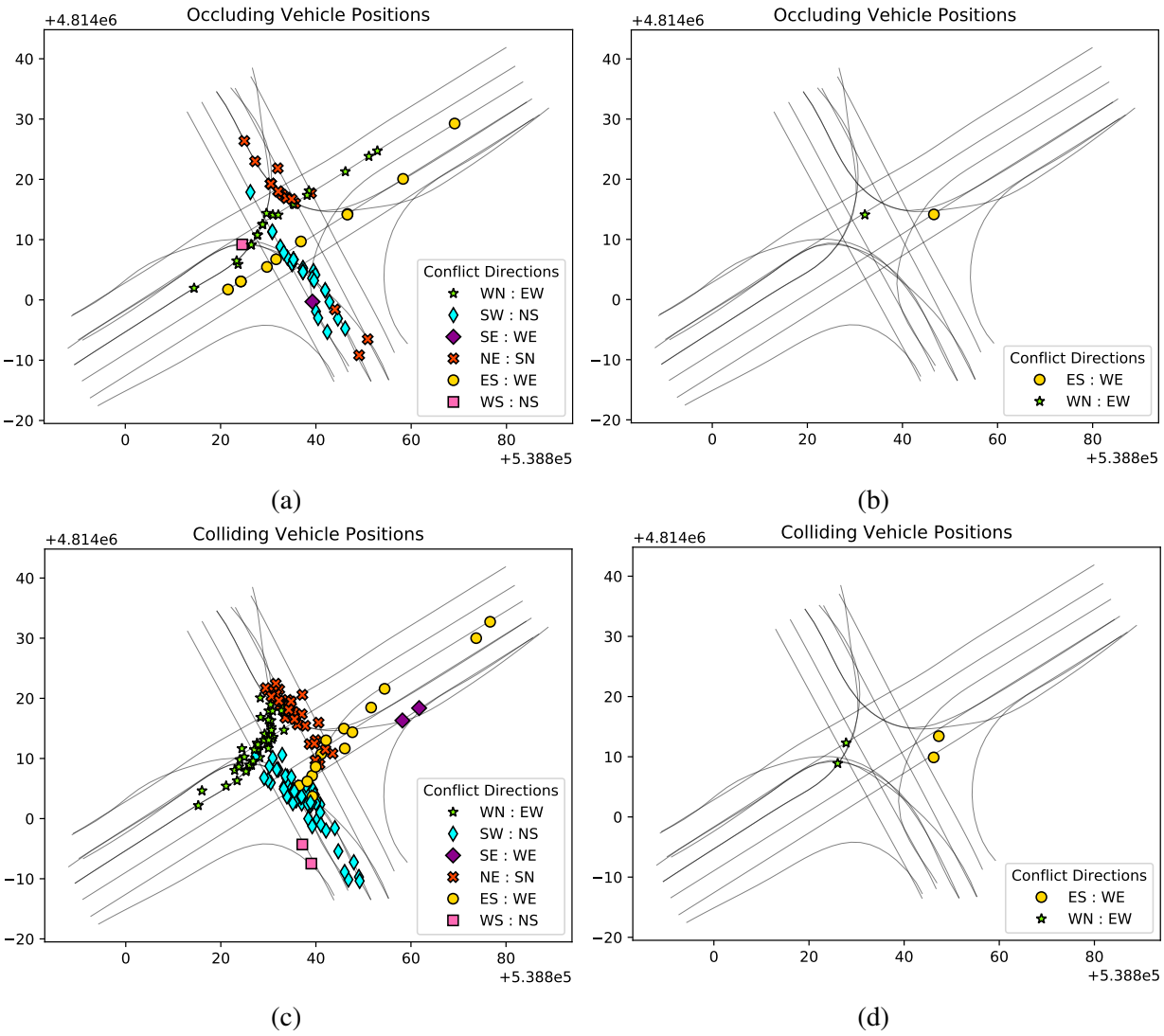
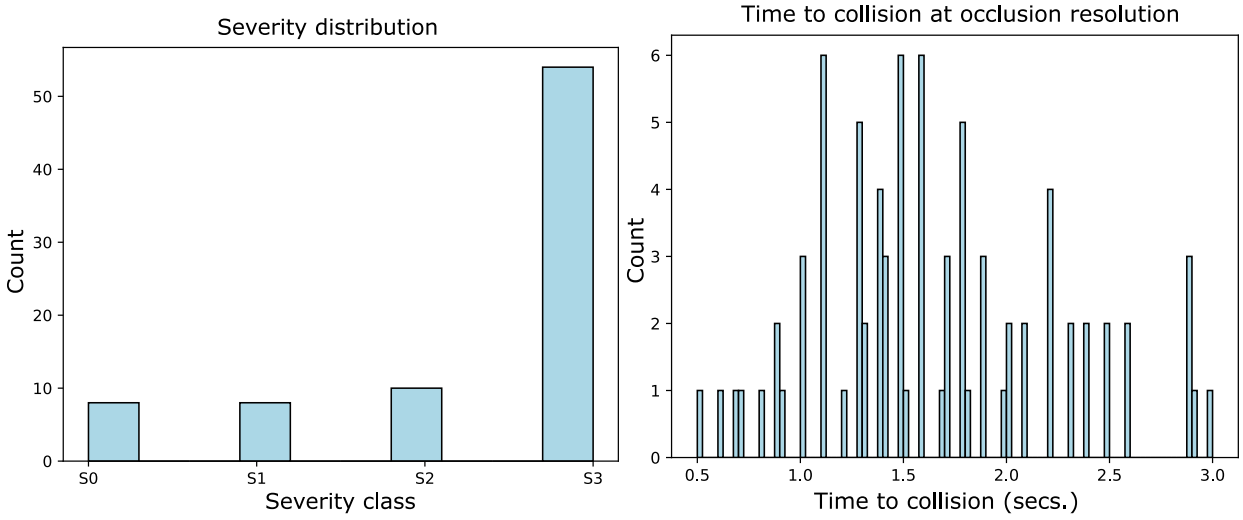


Figure 7.6: Figures (a) and (c) show the occluding vehicle and colliding vehicle positions for the 80 synthetic OCCs. Figures (b) and (d) show the same information but for the 2 naturalistic OCCs. The colliding vehicle positions are plotted at the moment of impact.



(a) (b)

Figure 7.8: (a) Severity class distribution. (b) The distribution of time duration from occlusion resolution to collision.

7.6 Related work

The literature on occlusion-aware planning can be divided into two categories: reachable set analysis and probabilistic methods. Reachable set analysis [170, 115, 96] provides a method to generate provably-safe trajectories by over-approximating the occupancy states of potential occluded vehicles. A challenge with set-based approaches is to tune the implementation such that the vehicle not only produces provably-safe trajectories but also does not behave too conservatively so as to disrupt the flow of traffic.

Probabilistic models, such as partially observable Markov decision processes [30, 143, 96], provide a method to both handle uncertainty and perform optimal decision-making. Particle filters [242, 161], use Monte Carlo sampling to approximate future positions of potential occluded vehicles, where large clusters of particles indicate a high likelihood of future occupancy. McGill et al. [150], propose a probabilistic risk assessment tool which, in addition to incorporating cross traffic, sensor errors and driver attentiveness in its risk calculation, also uses a dynamic Bayesian network to reason about the occupancy of road segments. Occlusion-aware deep reinforcement learning (DRL) [99, 108] has been used to learn safe policies for navigating unsignalized intersections. However, both these works have only been applied to situations with static occlusion and it is difficult for DRL to adapt to unseen scenarios. Unlike reachable set analysis, probabilistic methods do not allow for safety guarantees.

In the context of safety validation of AVs, there are several black-box approaches to the problem [169, 52, 250]. In contrast, our work addresses the problem of safety validation from a white-box perspective, which has received relatively less focus in this domain. Although a very recent work addresses the problem of planning [248], to our knowledge, there are no existing methodologies for the problem of validation of strategic planners.

7.7 Conclusion

In this work we presented a novel safety validation framework for strategic planners in AV. We showed how the theory of hypergames can be used to develop a novel multi-agent measure of situational risk associated with dynamic occlusion scenarios. Based on that measure, we developed an accelerated approach of safety validation by augmenting naturalistic datasets with realistic dynamic occlusion scenarios, and assessing the safety of a strategic planner. We showed that the validation method can achieve 10^4 gain in generation of dynamic occlusion scenarios, 4000% gain in generation of collision scenarios, as well as diversity and alignment with common crash situations. Ultimately, we foresee our proposed method fitting into a larger safety validation pipeline [114], where, first, failure scenarios are found in low-fidelity in an accelerated manner, followed by a high-fidelity examination of these failure scenarios. We hope that this work can be a stepping stone for performing safety validation of autonomous vehicles under situations with high levels of dynamic occlusion.

Chapter 8

A taxonomy of interactions

8.1 Introduction

Human coordination and cooperation are crucial to solving conflicts in any busy traffic situation. With self-driving cars thrown into the mix, it is clear that behaviour planning algorithms for autonomous vehicles (AVs) need to understand and act in a manner that ensures safe co-existence with other human road users for the foreseeable future. Coordination between road users is often mediated by traffic rules; however, anyone who has ever walked through a busy city intersection recognises that humans do not always act according to the prescribed rules, and *ad-hoc* strategic interactions take the place of strict adherence to the rules. To equip AVs to participate in such interactions, in recent years there has been a focus on strategic models for AVs, where a set of road users are modelled as players in a general sum game, and various solution concepts have been applied to address the problem of planning [70, 215, 136, 189], as well as the problem of modelling naturalistic human driving behaviour [196, 210, 85]. The expression of strategies generated by these models is closely tied to the choice of the action space in the game formalism, and covers a wide array of examples, such as specific control actions (acceleration, target velocity, etc.). [140], continuous trajectories [85], combination of hierarchical long- and short-horizon control actions [70], and combination of hierarchical high-level manoeuvre and trajectory patterns [196]. Although these models collectively provide a rich landscape of strategic models for engineers to choose from in the development of AVs, when it comes to a broader understanding, there is a lack of a common language to communicate to all stakeholders, including engineers, regulators, and the broader public what these strategic behaviours in a given traffic situation on the part of an AV entail. AV manufacturers can use their proprietary designs with the several design options that game-theoretic modelling provides; however, a regulatory authority may be

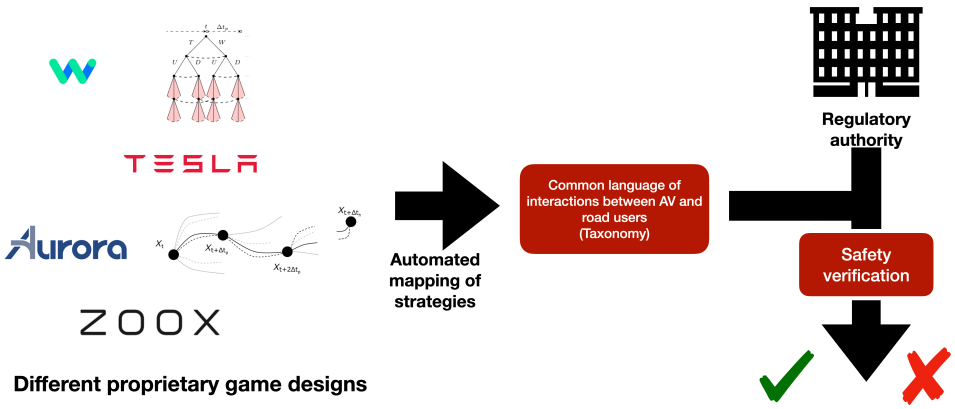


Figure 8.1: A typical use case for the taxonomy developed in this chapter. Different manufacturers may use proprietary designs which when mapped automatically to a common taxonomy of strategies can be verified by a regulatory authority.

interested in determining whether the strategies executed by an AV adhere to safety standards more than the manufacturer's internal design choices (Fig. 8.1). Therefore, as a step towards developing a common language for strategic interactions that is independent of game design, this chapter develops a simplified taxonomy of traffic interactions that encapsulates common patterns of behaviour observed in traffic, and a method of automated translation of the strategies generated by strategic planners into that taxonomy.

Along related lines, the need for a taxonomy of strategic interactions is also relevant when developing safety standards. What is considered safe action for an AV should ideally depend on verifiable safety specifications, and frameworks such as Responsibility-sensitive Safety (RSS) [202] and Safety Force Field [165] provide such frameworks of safety requirements. However, the safety requirements in the frameworks focus on short-term reactive safety (e.g. over instantaneous velocities of vehicles), whereas strategic planners generate a plan over a longer-horizon interactions. Since unsafe behaviour at the strategic level can eventually lead to hazardous situations, there is a gap in current safety specification frameworks to address strategic safety. The development of a taxonomy for strategic interactions is a first step toward addressing that gap, thereby allowing for identification of hazardous behaviours of a subject AV in strategic interactions in traffic conflicts.

In this chapter, I develop a taxonomy of strategic interactions based on common determinants of traffic interactions, such as who claims the right-of-way, whether an agent relinquishes that right, whether an agent responds to the actions of other agents, and other alternate ways of resolving conflicts. We also provide an example mapping of strategies from the form that a

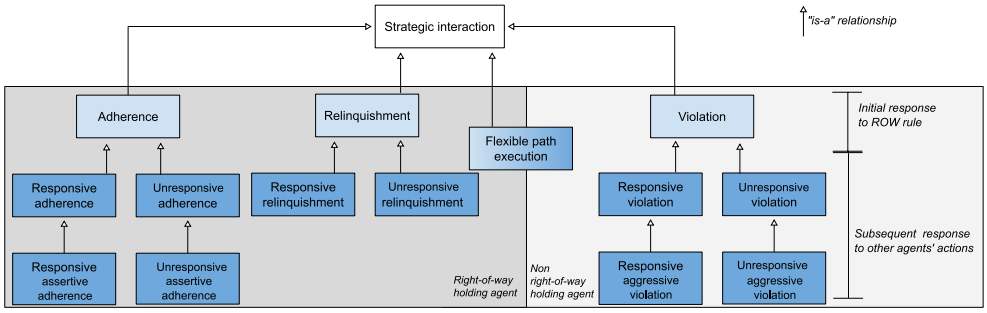


Figure 8.2: Schematic diagrams showing the relation among the taxonomy of strategies.

strategic planner AV generates to the taxonomy developed in this chapter, followed by an experimental evaluation of two popular models of strategic behaviour, QLk and SPϵNE, based on one pedestrian-vehicle and one vehicle-vehicle interaction scenario in simulation with respect to the taxonomy. Finally, we also provide real-world interaction video clips to illustrate the usefulness of the taxonomy.

8.2 Taxonomy of strategies

The taxonomy we develop is focused on traffic situations that have a static conflict point, such as intersections and roundabouts. We first present the dimensions based on which the taxonomy is organised followed by the taxonomy. **Right-of-way (ROW) rules.** One of the basic tasks in traffic navigation is conflict resolution. Conflict points are locations in the traffic network where multiple lanes intersect, merge, or intersect with a crosswalk, and road users traversing those lanes need to behave in a coordinated manner to reduce the risk of collision [175]. Traffic rules play a major role in resolving the conflicts and provide guidelines of behaviour for all the road users involved. The way traffic rules resolve conflicts is by ROW assignment to a road user (e.g. who has the priority to proceed at an unsignalised intersection), where a road user holding the ROW has priority and can proceed to be the first to cross the conflict zone. In some jurisdictions, such as Austria, traffic rules even require road users relinquishing their ROW to indicate that through a signal. Therefore, the first dimension of the taxonomy is based on how road users (agents) behave in relation to the ROW rule in a traffic conflict.
Responsiveness. The second dimension of the taxonomy is based on how agents behave in relation to the actions of other agents. A basic assumption of game-theoretic modelling of traffic is that the agents play a common game, in which each agent is aware of the other agents in the game. On the other hand, aspects such as distraction, mis-attention, and occlusion, lead to circumstances where one agent may not be aware of the other agent; and therefore, due to these

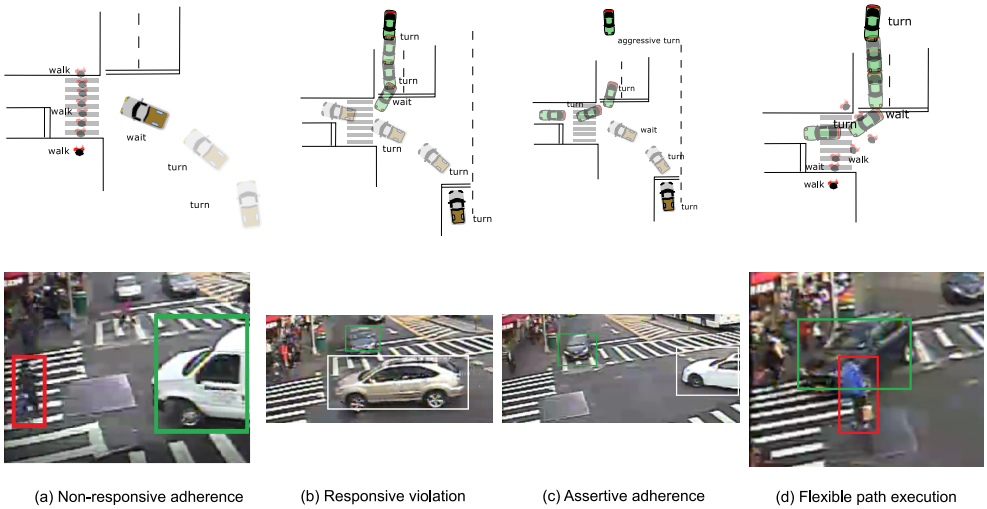
arguably natural aspects of traffic and human behaviour, the common game assumption breaks down. For humans, checking whether another road user is aware of them and at the same time making the other road user aware of their presence is a skill that we learn over time and takes the form of various non-verbal modalities of communication [181]. One such modality is through kinematic motion patterns [60, 7], where, based on changes in motion trajectories (e.g. pedestrians slowing down for a turning vehicle), a road user may communicate an acknowledgement that they are aware of another road user. Therefore, the second dimension of the taxonomy, *responsiveness*, is built upon the above idea, where we denote an agent's strategy to be responsive if it involves changes in the motion characteristics in response to the actions of other agents. Whereas following the default ROW rule can resolve the game in one round, the second dimension becomes salient when, due to miscommunication or agents not following the default rule, the game goes to subsequent rounds. Fig. 8.2 shows the taxonomy of strategies categorized based on the two dimensions of initial response to ROW rule (lighter boxes) and subsequent response to other agents' actions (darker boxes). Anonymised links to real-world snippets showing examples for each category are noted in Table 8.2.

8.2.1 Taxonomy

Adherence

Adherence is a class of strategies in which an agent holds the ROW and starts to proceed. Depending on the subsequent state of the game and whether there is a change in the characteristics of its trajectory, an adherence strategy can be further classified as responsive or unresponsive. ***Unresponsive adherence* (UA)**. In an unresponsive adherence strategy, an agent claims their ROW by starting to proceed, and irrespective of the actions of other agents' in the vicinity with whom they are in conflict with, they do not change their motion trajectory. A typical unresponsive adherence strategy is moving with a steady velocity even in the face of other conflicting agents' (who do not have the ROW) attempts to violate that right. Fig. 8.3a shows an example of unresponsive adherence. In this scenario, a pedestrian at a crosswalk has the ROW and starts crossing the crosswalk, while a white left turning vehicle in conflict (who does not have the ROW) starts to proceed with the left turn at the same time. An UA strategy on the part of the pedestrian is to continue walking at a steady speed, as opposed to slowing down in response to the white vehicle's action.

Although the UA strategy is motivated along the lines of whether or not a road user is aware of another conflicting road user, there can also be alternate explanations for this type of strategy. For example, while it is possible that in the above example, the pedestrian did not notice the white vehicle moving into the crosswalk, it is also possible that the pedestrian kept moving at a



(a) Non-responsive adherence (b) Responsive violation (c) Assertive adherence (d) Flexible path execution

Figure 8.3

steady speed because they anticipated the white vehicle would slow down and wait after noticing them to be unresponsive, in other words, UA just being the optimal strategy from the pedestrian's perspective.

***Responsive adherence* (RA)**. Similar to UA, responsive adherence also implies that a ROW holding road user claims that right by starting to proceed. However, if a non-ROW holding conflicting agent attempts to violate that right and proceeds at the same time, the ROW holding agent may demonstrate a change in their trajectory, for example, by slowing down and proceeding cautiously. In the example of Fig. 8.3a, a responsive adherence strategy on the part of the pedestrian can be a) slowing down and proceeding with caution upon observing the turning white vehicle, b) increasing their speed to move through the crosswalk fast to clear the way for the turning vehicle, and c) slowing down and waiting for the turning white vehicle to pass. Although these individual strategies represent different ways of dealing with the conflict, they demonstrate commonality in terms of the pedestrian being aware of the vehicle, a signal being established that they are playing a common game by their reaction to the white vehicle, as well as the initial willingness of the pedestrian to claim their ROW status.

***Responsive and unresponsive assertive adherence* (RAA and UAA)**. Assertive adherence is a specific type of adherence strategy, where a road user holding the ROW starts to proceed aggressively (aggression can be indicated by a higher than normal jerk and acceleration) in order to dissuade any conflicting road user from violating their right. The goal of this strategy is to signal a strong willingness to defend the ROW. Depending on whether the agent subsequently modifies their trajectory characteristics as a response to other agents' action, an assertive adherence can be responsive or unresponsive. Fig. 8.3c shows a responsive assertive adherence scenario, where

the left turning vehicle starts to turn without holding the ROW; however, the right turning vehicle in this case starts an aggressive turn to dissuade the other vehicle from violating their ROW, but subsequently changes its manoeuvre to a non-aggressive turn when the white vehicle relents.

Relinquishment

Relinquishment strategy is the opposite of adherence, where an agent holding the ROW chooses to wait and let another agent (who does not hold the ROW at that moment) proceed instead.
Unresponsive relinquishment **(UR)**. In a unresponsive relinquishment strategy, the agent continues to wait for the other agent to pass even when the other agent keeps waiting.
Responsive relinquishment **(RR)**. Under responsive relinquishment, an agent after initially relinquishing their ROW reclaims the right and proceeds if the other agent continues to wait.

Violation

As the name suggests, this type of strategy is demonstrated by an agent who does not hold the ROW in the game, but decides to proceed. Similar to adherence strategies, ROW violation strategies can also belong to two categories, responsive and unresponsive.
Unresponsive violation **(UV)**. In this strategy, an agent without holding the ROW starts to proceed, and continues while not responding to the ROW holding agents' attempt to reclaim their right. Under such a scenario, to avoid a collision, some of the strategies that the other agent can generate are unresponsive relinquishment, and responsive adherence of the type where they slow down and wait. Responsive adherence of speeding up can also avoid a collision but may be much more riskier choice.
Responsive violation **(RV)**. In this type of ROW violation, a vehicle without holding the ROW starts to proceed; however, as a response to another ROW holding agent reclaiming their right, they can respond by changing their trajectory during the course of the game. Fig. 8.3b illustrates an example of responsive violation on the part of the white left-turning vehicle, which on account of not holding the ROW should have waited for the right turning vehicle. However, after starting to execute the turn, upon observing the right turning vehicle (who had the ROW) not relinquishing their right, speeds up to complete the turn fast in order to avoid getting stuck in middle of the intersection. The white left turning vehicle in the previous example of Fig. 8.3c also demonstrates responsive violation of a different flavor, where it first starts to proceed without holding the ROW, but eventually relents when the right turning vehicle demonstrates an assertive adherence strategy.
Responsive and unresponsive aggressive violation **(RAV and UAV)**. Similar to assertive adherence, if a violating vehicle starts to proceed aggressively, this strategy is categorised as aggressive

violation (the term aggressive is commonly used as a driving behaviour indicator [191] and in this case is used instead of the word assertive to indicate that the agent did not hold the ROW). Depending on whether an agent demonstrating aggressive violation later changes its trajectory characteristics based on the response of the other agents' action, the strategy can be of responsive or unresponsive flavor.

Flexible path execution (FP)

The strategies presented until now are primarily based on a road user's choice of trajectory. Without the temporal component of a trajectory, just the sequence of locations (called a path) is generally determined by navigable traffic regions, such as vehicle lanes for vehicles, bike lanes for bicycles, and sidewalks and crosswalks for pedestrian. Common strategies in traffic are variations over the trajectories by agents changing their movement velocities, whereas changes in path are often minimal and are kept within the prescribed navigable regions. However, in some cases, road users may choose to execute an alternate path as a way to resolve a conflict. This alternate path execution strategy may be executed by an agent regardless of their ROW status, and is demonstrated by agents choosing an alternate path to their desired goal location. Fig. 8.3d shows an example of an alternate path execution strategy that follows after a (RA,RV) strategy. In this scenario, a pedestrian having the ROW starts to proceed over the crosswalk, and at the same time a right turning vehicle continues to turn through the intersection thereby attempting to violate the ROW. Both agents generate responsive strategies by observing each other's actions, and choose to wait for the other agent to cross as their next action, thereby leading to a deadlock. In order to resolve that deadlock, instead of reclaiming their ROW over the crosswalk, the pedestrian chooses to go around the back of the vehicle to the other side of the crosswalk. This type of strategy can be an initial action in the game as well as a subsequent response and can be demonstrated by both the ROW and non-ROW holding agent (noted by the placement of the block in Fig. 8.2).

8.2.2 Intermediate outcome — deadlock

Although not a strategy in and of itself, deadlocks are momentary states in the game where agents in conflict stop and wait in order to avoid a collision. Deadlocks follow from certain strategy choices (for example, the case of (RA,RV) in the previous example) and need a way of resolution in the subsequent steps of the game. It can also result from a combination of relinquishment and adherence strategies where both agents wait for the other to proceed. Attempted resolutions of a deadlock can take one of the following forms — a ROW holding agent proceeds (RA), a non-

ROW holding agent proceeds (RV), both proceeding at the same time (RA,RV), and any of the agents involved in the deadlock chooses an alternate path to their destination (FP).

8.3 Mapping strategies to taxonomy

In order for the developed taxonomy to be usable in the evaluation of AV planners, there needs to be a well-defined translation of the strategies that a strategic planner generates into one of the strategy categories developed in the taxonomy. Although developing a translation for every strategic planner proposed in the literature is beyond the scope of this chapter, in this section, we illustrate one example. Hierarchical game theoretic planners, similar to the ones proposed in [196], generate strategies that are a combination of high-level manoeuvre and a low-level trajectory segment, where each trajectory segment at a node in the game belongs to one of the two types of high-level manoeuvre, *wait* or *proceed*. We denote each manoeuvre as a symbol, where w corresponds to the *wait* manoeuvre, p corresponds to the *proceed* manoeuvre, and p_a is a *aggressive* proceed manoeuvre defined based on the maximum acceleration of the trajectory. The manoeuvre choices in the strategy can then be expressed as a regular expression (regex), where each literal in the regex corresponds to a manoeuvre choice at each node of the game starting from $t = 0$ (Table 8.1). To illustrate a few examples, for an agent holding the ROW, an unresponsive relinquishment strategy involves the generation of a manoeuvre *wait*, regardless of the actions of the other agents, which can be expressed as the regex w^+. Similarly, a regex $w^+p\{w|p\}^*$ expresses strategies in which the agent chooses the wait manoeuvre at the first node, followed by at least one proceed choice at any node in the game. For a non-ROW holding agent, this would imply responsive adherence since they adhere to the ROW rule by waiting; however, they can subsequently respond to the other agent's action by choosing a proceed manoeuvre. On the other hand, for a ROW holding agent, the same regex is responsive relinquishment type. While we defined the process of matching a strategy to one of the taxonomy purely based on the manoeuvre choices, one can also think of a more granular process of matching the strategies based on the trajectory choices; for example, based on bounds of the acceleration to define what counts as a steady movement for unresponsive strategies.

8.4 Evaluation of strategic behaviour models

One of the goals of developing a taxonomy is to establish a common language based on which different models of strategic behaviour can be compared and evaluated. For the context of this chapter, we perform an evaluation of two popular classes of models proposed for the problem

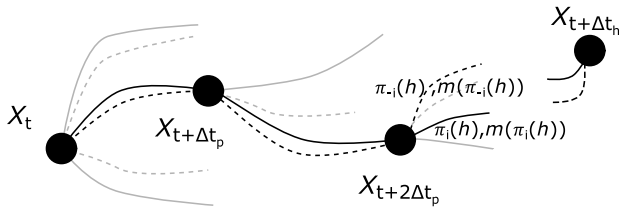


Figure 8.4: Schematic representation of the dynamic game. Each node is embedded in a spatio-temporal lattice and nodes are connected with a cubic spline trajectory.

of strategic behaviour planning in the AV literature, the QLk model [215, 140] and the Nash equilibrium [179, 85, 155], and discuss the type of strategies that each model generates. More specifically, for the Nash equilibrium based model, we select subgame perfect $\epsilon-$Nash equilibrium (SPϵNE) [72] as the solution concept for even comparison with QLk in regards to the support for bounded rational agents in a dynamic game setting. For the Qlk model, there is a choice to be made on the solution concept that a level-0 agent uses based on the caveat that level-0 behaviour should be nonstrategic. To that end, we select a maxmax model due to its higher alignment with naturalistic driving behaviour compared to other non-strategic models [196]. In our implementation of the QLk model, all agents are modelled as level-1, i.e., internally they model other agents as level-0 [234].

8.4.1 Game structure

The game is constructed as a dynamic game between two road users in conflict modelled as a sequence of simultaneous move games played every $\Delta t_p = 1.3$ secs. starting at time $t = 0$ over a horizon of $\Delta t_h = 4$ secs. The rest of the construction of the game tree is identical to the one in Chapter 5.

8.4.2 Simulation runs

For each of the models, we run simulations in two scenarios, one pedestrian-vehicle interaction scenario and one vehicle-vehicle interaction scenario. For the pedestrian-vehicle scenario (Fig. 8.6b), a pedestrian (who has the ROW) has just started to cross the crosswalk at the moment of initiation of the game, and a right turning vehicle should ideally wait for the pedestrian to cross. For the vehicle-vehicle interaction scenario (Fig. 8.6 c), a right-turning vehicle holding the ROW has just started to execute its turn, and the left-turning vehicle should wait for the right-turning vehicle to cross before executing the turn. Both scenarios are simulated based on an intersection in New York City (Fig. 8.6 a). The scenarios are initiated based on different initial velocities

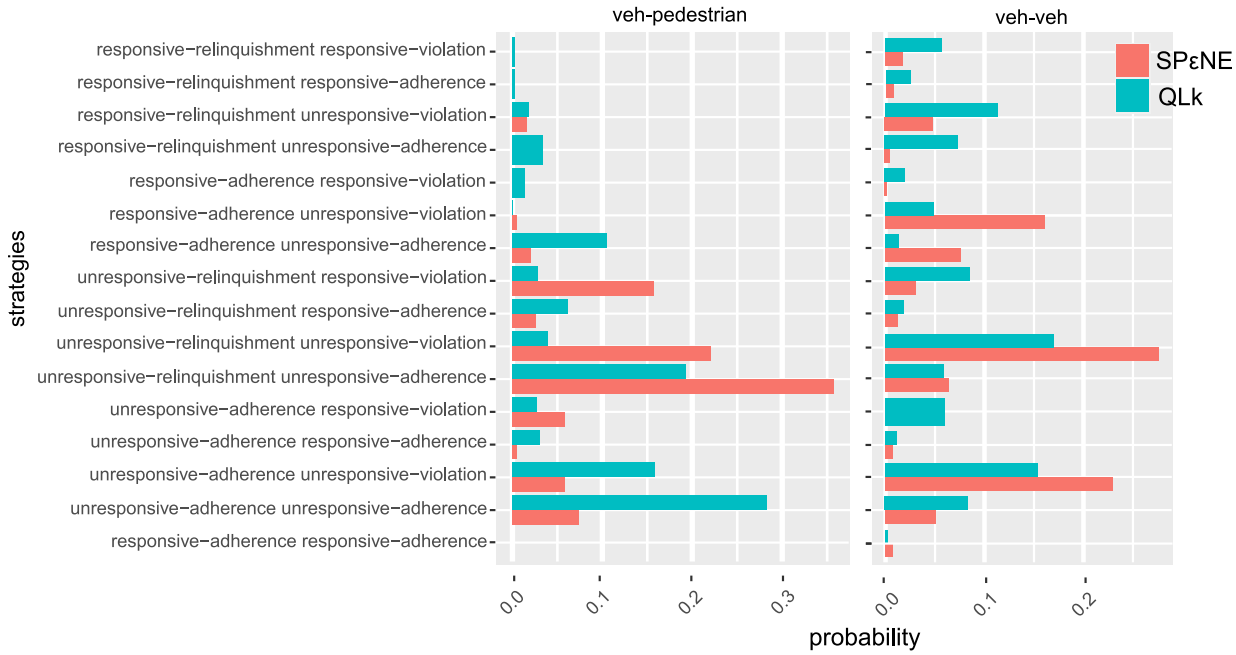
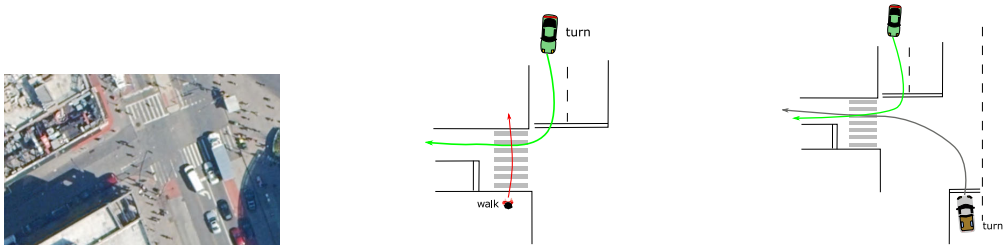


Figure 8.5: Distribution of strategies for SPϵNE and QLk model for the two simulation scenarios.

of the agents (1.3 to 1.8 ms^{-1} for pedestrians and 1 to 12 ms^{-1} for vehicles) as well as all combinations of agent types $\gamma \in$ [-1,1] with increments of 0.5. In total, there are a total of 1250 games for the pedestrian-vehicle scenario and 2,500 games for the vehicle-vehicle scenario. All games are solved in a complete information setting with respect to agent types, the parameter ϵ for SPϵNE model is 0.1, and the precision parameter (λ) for QLk is 1.

8.4.3 Results

Fig. 8.5 shows the distribution of the basic strategies (without the aggressive and assertive modifiers) corresponding to each behaviour model. We see that there is a wide range of strategies that each model generates based on the initiating situations of the game. However, for agents holding the ROW, unresponsive strategies (both relinquishment and adherence) are the most common type of strategies generated by both models. Given that the simulation did not incorporate an explicit model of distraction or misattention, the results demonstrate that the unresponsive strategies can also be the optimal strategies compared to the other alternatives in some situations. Certain strategies are frequently seen in both models, for example, (UR,UA), i.e., both pedestrian and vehicle waiting for each other in vehicle-pedestrian interaction, and (UR,UV), i.e., pedestrian waits, and the vehicle executes the turn for vehicle-vehicle interaction. There are other strategies



(a) (b) (c)

Figure 8.6: Pedestrian-vehicle and vehicle-vehicle interaction scenario

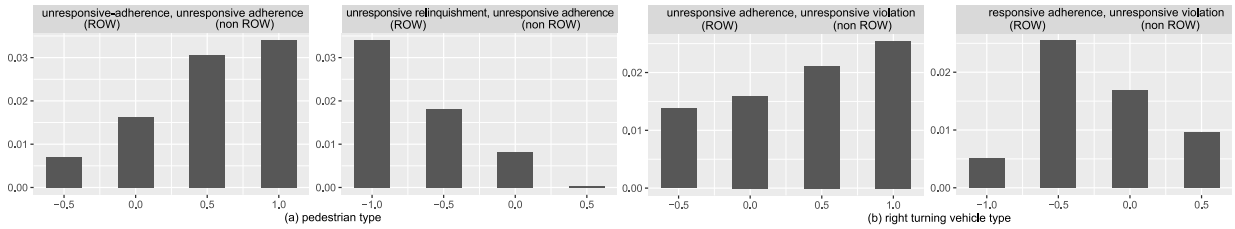


Figure 8.7: Distribution of agent types for ROW holding agent for frequently generated strategies.

in which the models show considerable disagreement. For the vehicle-pedestrian scenario, the most common strategy in the QLk model is unresponsive adherence for both agents (the vehicle waiting for the pedestrian to cross) and is observed in approximately 30% of the games. Although this is arguably the ideal strategy in this scenario, it is not a commonly observed strategy in SPϵNE model (less that 10% of the games). Similar disagreement is seen for vehicle-vehicle scenario with respect to (RA,UV) strategy, i.e., the right turning vehicle claims its ROW and starts to move but waits because the left turning vehicle violates the ROW in an unresponsive way; this strategy is more commonly observed in SPϵNE model than QLk.

The generated strategies also depend on the type value of the agents (recall that the higher the type value of an agent, the higher their risk tolerance). Figure 8.7a shows the distribution of the two most common strategies observed in the QLk model for the pedestrian-vehicle interaction scenario with respect to the pedestrian type value. Since with higher types, the chances that the pedestrian generates a proceed action increase, we see in the two graphs of Figure 8.7 a that with higher types the probability of unresponsive adherence increases and unresponsive relinquishment decreases. There is also an association between the agent type value and whether the strategies they generate are responsive or unresponsive. Fig. 8.7b contrasts the strategies (UA, UV) and (RA, UV), which differ only in responsiveness. The first graph of Fig. 8.7 b shows what we would normally expect; the chances that a vehicle claims its ROW even in the face of the right being violated will increase with its type, reflected in the increasing probability of

the unresponsive adherence strategy. However, as we see in the second graph, the probability of responsive adherence strategy peaks at medium risk tolerance (-0.5). This is because if the risk tolerance is too low, then the ROW holding agent will not claim their ROW at all and, thus, will not generate an adherence strategy. On the other hand, if their risk tolerance is too high, then they will demonstrate unresponsive adherence rather than responsive one. Based on the above analysis, first, since the two popular models generate a wide range of strategies based on different game situations, an AV will need to carefully evaluate whether a chosen strategic model is appropriate based on the type of strategy they generate in that specific situation. Second, since certain types of strategy are more commonly generated by agents with specific types, an AV can use the observed strategies to form a possible hypothesis about a road user, for example, if a road user generates a responsive strategy, then their risk tolerance may be lower than someone generating an unresponsive strategy, and an AV needs to take into account that information.

8.5 Conclusion

This chapter presented a taxonomy of human strategic interactions in traffic conflicts that is built on the basic dimensions of how agents respond to their right of way and to each other's actions in a game. We demonstrated a way to map strategies from a form generated by a typical strategic planner to those in the taxonomy. Based on our evaluation of two popular solution concepts used in strategic planning, we highlighted the relation between the types of strategy generated and the agents' risk tolerance. On top of the presented taxonomy, one can build a formal framework of the emergent communication as well as outcomes based on the different combination of strategies, e.g., resolution based on traffic rules, deadlock, severe conflict (near miss, crash, etc.), successful violation, etc. We hope that along with the development of a more formal framework for strategic safety specifications over the taxonomy as a future work, the taxonomy can also be extended for other traffic situations with discretionary actions and dynamic conflict points, such as lane changes, which were not covered in this chapter.

ROW status	Maneuver strategy regex	Matched taxonomy
For agent holding ROW	w^+	unresponsive relinquishment (UR)
	p^+	unresponsive adherence (UA)
	p_a^+	unresponsive assertive adherence (UAA)
	$w^+p\{w\|p\}^*$	responsive relinquishment (RR)
	$p^+w\{w\|p\}^*$	responsive adherence (RA)
	$p_a^+w\{w\|p\}^*$	responsive assertive adherence (RAA)
For agent not holding ROW	w^+	unresponsive adherence (UA)
	$p^+w\{w\|p\}^*$	responsive violation (RV)
	$p_a^+w\{w\|p\}^*$	responsive aggressive violation (RAV)
	p^+	unresponsive violation (UV)
	p_a^+	unresponsive aggressive violation (UAV)
	$w^+p\{w\|p\}^*$	responsive adherence (RA)

Table 8.1: Regular expression mapping strategies to taxonomy

Category	Link to an example snippet	Description
unresponsive adherence	https://git.uwaterloo.ca/a9sarkar/traffic_taxonomy_project/-/blob/master/videos/taxonomy_snippets/non_responsive_adherence.mp4	Person walking on left crosswalk
responsive adherence	https://git.uwaterloo.ca/a9sarkar/traffic_taxonomy_project/-/blob/master/videos/taxonomy_snippets/responsive_adherence.mp4	Person walking on left crosswalk
responsive assertive adherence	https://git.uwaterloo.ca/a9sarkar/traffic_taxonomy_project/-/blob/master/videos/taxonomy_snippets/responsive_assertive_adherence.mp4	Right turning vehicle
unresponsive relinquishment	https://git.uwaterloo.ca/a9sarkar/traffic_taxonomy_project/-/blob/master/videos/taxonomy_snippets/non_responsive_relinquishment.mp4	Person standing on left crosswalk
unresponsive violation	https://git.uwaterloo.ca/a9sarkar/traffic_taxonomy_project/-/blob/master/videos/taxonomy_snippets/non_responsive_violation.mp4	Person walking on bottom-right crosswalk
responsive violation	https://git.uwaterloo.ca/a9sarkar/traffic_taxonomy_project/-/blob/master/videos/taxonomy_snippets/responsive_violation.mp4	silver left-turning vehicle
flexible path execution	https://git.uwaterloo.ca/a9sarkar/traffic_taxonomy_project/-/blob/master/videos/taxonomy_snippets/flexible_path_execution.mp4	Person walking on left crosswalk
deadlock	https://git.uwaterloo.ca/a9sarkar/traffic_taxonomy_project/-/blob/master/videos/taxonomy_snippets/deadlock.mp4	Between person walking on left crosswalk and right-turning vehicle

Table 8.2: Links to real-world examples of each category of strategy developed in the taxonomy

Chapter 9

Conclusion and future work

9.1 Conclusion

Building artificial intelligence systems that can reason in a manner similar to that of humans is a challenging task. However, this is an essential step towards the advancement of human-AI interaction systems, such as self-driving cars. The focus of this dissertation is on the use of game theory to build computational models of human driving behaviour so that the models can be used as practical tools for the development and testing of autonomous vehicles (AV).

Game theory provides a mathematical framework of decision making in which rational agents reason strategically over each others' actions, and decide upon an optimal course of action in a game. The theory is based upon several assumptions, most important of which is the idea of a rational agent who (or which) selects utility maximising actions after modelling everyone else as rational agents. There are also additional assumptions in the theory, such as that agents are expected to know who they are playing the games with, the available actions of each player in the game, and some information about the preferences of other players in the game. Once these assumptions are met, the solution of the game can accurately predict the behaviour of every agent in the game. However, in order to make this theory useful for building models of decision making as observed in a naturalistic real-world setting, this dissertation systematically evaluates each assumption. The dissertation also provides empirically backed practical guidance to practitioners for the construction of the game-theoretic models of road user behaviour.

The focus of Chapter 3 is the question: *Which model of reasoning fares best with respect to model fit and predictive accuracy when evaluated against naturalistic data?* In order to make the models more intuitive and easy to integrate into existing AV development process, the models are

developed over hierarchical games, i.e., a game construction where the actions of players in the game are decomposed hierarchically into manoeuvres and trajectories. The chapter developed three sampling-based methods of action construction, namely S(1), S(1+G), and S(1+B). The evaluation of the three methods found that S(1+B) method, where agents sample a prototype trajectory along with the trajectories corresponding to spatial and velocity limits under each manoeuvre, shows higher predictive accuracy compared to alternate methods. Based on different methods of action construction, as well as the solution concepts that can be used to solve the games, the chapter evaluates thirty behaviour models with respect to naturalistic observational data of driving behaviour at a busy intersection. A model of behaviour, QlkR, where each agent believes that other agents will strictly adhere to basic traffic rules and the agent best responds based on that belief demonstrates superior predictive performance and model fit compared to alternate strategic (Nash equilibrium), non-strategic (maxmax and maxmin), and behavioural game theoretic (Quantal level-k) based models.

When human behaviour in games is studied in a laboratory setting, the utilities are often constructed by the designer, and players are given that information *a-priori*, which in turn influence their behaviour. However, the process is more difficult when game theoretic models are constructed from observational data since the utilities are beyond the control of the designer. Driving behaviour in particular is also multi-objective with safety and progress as the two main dimensions along which utilities of actions are aggregated. Chapter 4 focused on two theories of utility aggregation, weighted aggregation, in which players weigh both safety and progress as a linear combination, and satisficing aggregation, in which players evaluate utilities based on one or the other dimension depending on the traffic situation. The process of inferring the utilities of agents from observations is formulated as an estimation problem of the multiobjective aggregation parameter. The key question addressed in this regard is *how to estimate preferences of players in a game from observational data that takes into account the underlying model of reasoning?* Chapter 4 develops algorithms for estimation of utility aggregation parameters from observed data for both weighted and satisficing aggregation and strategic and nonstrategic reasoning models. Furthermore, the chapter also evaluates the predictive performance of models when utilities are constructed with the two aggregation methods for both strategic and non-strategic models. The results showed that a linear weighting of safety and progress utilities had superior predictive performance for intersection and roundabout scenarios, whereas satisficing methods showed superior performance for crosswalk scenarios. The performance of models is also significantly improved when a regression tree based learning method is applied to infer the aggregation parameters in unseen traffic situations compared to a fixed aggregation parameter.

Humans often demonstrate complicated sequence of actions that cannot be captured easily using normal form games. Therefore, the thesis also develops models for dynamic games, in which the space of strategies includes a more complicated sequence of actions. Chapter 5 addressed

problems that arise as a result of applying standard models from behavioural game theory, such as the level-k model, in dynamic games. The first question, *how to construct a model of level-0 behaviour for dynamic games that is non-strategic yet sophisticated enough for human driving behaviour*, is addressed through two finite-state transducer (FST) based models, *accomodating* and *non-accomodating* automata. When agents are allowed to be of different degrees of risk tolerance, it is shown that almost all observed behaviours in two different datasets that include right and left turning behaviours, can be explained through a combination of the two FST models. Chapter 5 also addressed the question: *how to build game theoretic models of traffic behaviour that supports heterogeneous models of reasoning?*. Based on three different layers of reasoning capabilities, nonstrategic, strategic, and robust, in which each layer can hold multiple models, a generalised cognitive hierarchy model is built to support heterogeneous models in a dynamic game. Simulation based evaluation of three different traffic scenario setups showed that an AV behaviour planner using a robust response model has superior success rates when facing a population of drivers with diverse risk tolerance.

The second part of the dissertation focused on the applications of game theoretic models for safety validation of AV planners. Chapter 6 developed a methodology to generate rare events to test AV planners for lane-change scenarios. Different values of the precision parameter of a Quantal Best Response model in a multi-objective setting are mapped to eight categories of interpretable lane change behaviour. Using a simulated annealing based optimisation method, the evaluation methodology finds the behaviour category and parameter values that maximise the likelihood of a crash. Compared to a standard technique of rare-event sampling over surrogate safety metrics combined with cross-entropy optimisation, the proposed approach showed a speedup of 39% in generating crash situations. Chapter 7 shows another application of game theoretic models for safety evaluation of AV planners. Dynamic occlusions are momentary situations in traffic where a vehicle is occluded from the view of a subject vehicle by another vehicle. These types of situation are one of the leading causes of traffic accidents at intersections. In the proposed method, we show that the framework of hypergames, in which different agents have different views of the game, can be used to develop a novel surrogate risk metric, Dynamioc Occlusion Risk (DOR). When new situations are sampled from a naturalistic dataset, the DOR metric can be used to effectively evaluate the safety of an AV planner under the conditions of dynamic occlusion.

With different methods of action and utility construction along with various solution concepts, a designer has several design choices at hand that they can use to design a game theoretic behaviour planner for an AV. Understanding these modelling choices as well as the strategies generated by an AV often requires technical expertise. From the perspective of AV operators, revealing all design choices can compromise intellectual property. At the same time, even if detailed design choices are made known to regulators, they may require a simpler representation

to express what strategies are acceptable for an AV to execute. In order to address both of these problems, the thesis concludes by developing a taxonomy of strategic interactions in Chapter 8, which includes a simpler representation of action strategies for AV compared to game solutions. The chapter also includes simulation experiment demonstrating how solutions from different behaviour models can be mapped automatically to the taxonomy using regular expressions and the diversity of strategies that different behaviour models generate based on the proposed taxonomy.

9.1.1 Model recommendation

Chapters 3-5 presented the various modelling choices available to a designer for the purpose of modelling driving behaviour as a game-theoretic problem. As is the case with any modelling paradigm, the modelling decisions depend on the specific problem that the modeler wishes to address. However, to make it easier for a practitioner, I present a synopsis of recommendations based on the empirical findings in the chapters.

One-shot hierarchical games are well suited to model instantaneous decisions taken by road users and are studied in Chapters 3 and 4. Many traffic scenarios, for example, point of turn at intersections, entry at roundabouts, where the road user needs to make stop-go decisions are examples of scenarios where this modelling construct is applicable. Such scenarios are also more strategic in nature since the conflict presents a higher likelihood of an adverse event, and the road user needs to reason over the actions over other agents. In the context of these models, the following modelling recommendations are applicable.

- *S(1+B) sampling is the preferred method of action construction for the trajectories game in a hierarchical game.*
- *Maxmax is the recommended solution concept to solve the game of trajectories in a hierarchical game.*
- *Weighted aggregation of safety and progress utilities with weights inferred from observational data is the recommended method of utility construction for intersection and roundabout navigation. For crosswalk navigation, satisficing is the recommended method of aggregation.*
- *For weighted aggregation, the parameters of aggregation should be estimated from naturalistic observations. For satisficing aggregation, a parameter value 0 for the safety aspiration level can be considered adequate.*
- *QlkR is the recommended solution concept for the game of manoeuvres if a common behaviour model is used for all scenarios (including scenarios such as approach and exit from intersections).*

- *Nash equilibrium is the recommended solution concept for a subset of scenarios and situations where chances of strategic nature of interactions are higher (the point where a road user needs to make a stop/go decision at a conflict).*
- *In the case of multiple Nash equilibria, payoff dominant equilibrium selection is the preferred method.*

Dynamic hierarchical games use similar decomposition of actions into manoeuvres and trajectories like a one-shot hierarchical game, but can also capture the sequential nature of strategic interactions, including forming and updating beliefs about other road users based on observations. The sophistication of such models comes with the drawback of having to address issues around scalability. Chapter 5 studies such games in the context of left and right-turning behaviour. The recommendations for dynamic games are as follows.

- *The dynamic level k model is the recommended model for modelling traffic behaviour as a dynamic game, with level-0 behaviour modelled as a combination of accommodating and non-accommodating finite state transducers.*
- *For autonomous vehicle behaviour planning, a robust response to a heterogeneous population of strategic and nonstrategic reasoners is the recommended approach.*

9.2 Limitations and future work

Observational data and game-theoretic models are not natural allies. Nevertheless, in the age of *big data*, the two paradigms need to be bridged for human-centred AI systems to make use of research from social sciences and economics, where there exists extensive literature on modelling human behaviour using game theory. The thesis is a step in that direction with a narrow focus on traffic behaviour and autonomous vehicles. However, the creation of models from observational data can only generate hypotheses, and conducting user studies to confirm the hypotheses is an essential step of model building. The biggest limitation of this work is the absence of user studies due to reasons beyond our control[1], and this limitation also provides a direction for future work. Another limitation of the work is around the gap between the models developed in this thesis, especially for dynamic games, and the scalability issues that arise in a practical use of dynamic game-theoretic models for motion planning. There are also assumptions made in a few of the models, the most important of which is the assumption of modelling the games as one of perfect

[1] https://www.who.int/emergencies/diseases/novel-coronavirus-2019

information. This assumption made it easier to analyse boundedly rational behaviour, but is also a limitation because in reality uncertainty about the precise state of the game (which includes estimating the speed, location, etc.) is a key behavioural factor. The thesis covered some of the more common traffic scenarios (intersections, roundabout, crosswalk); however, there are several other scenarios that road users encounter in traffic (e.g., highways, ramps, and interchanges), and whether the empirical findings of this thesis would readily translate to other traffic scenarios, and other traffic cultures (such as dense conditions like India or China) is an open question.

The limitations of this work also helps us identify potential future work one can undertake, and I present a non-exhaustive list of such directions next.

User studies. The dissertation focused on observational data and matched the game constructs that align with the observed action choices of the players. The conclusion that the thesis draws about the reasoning model of road users needs to be validated with user studies conducted in parallel to the observational data. This can be done by recording observational data with a drone, and once the road user has completed the task, an exit interview can be conducted to elicit information about their action choices. The exit interviews would consist of the user selecting the reasoning model and utility structure that aligned with their actual thought process. This would require translating the game constructs into natural language descriptions that capture the key concepts behind them.

AV driving strategy. Although the dissertation touched on the problem of behaviour planning for AV, i.e., what driving strategy an AV should follow in traffic, in Chapter 5, the study is limited to the three specific scenarios. This work should be extended to more general research on the design of AV behaviour planning strategies. An open question in this context is whether an AV should follow a pure or mixed strategy behaviour. The benefits and disadvantages of both approaches should be evaluated on the basis of safety assessment and operational feasibility. An interesting direction is the characterisation of traffic situations where one should be preferred over the other.

Behaviour driven safety analysis. Similar to the approaches in Chapters 6 and 7, there is a need for a safety analysis that takes into account behavioural factors. Attitudes of impatience, distraction, road rage, etc. are often typical causes of traffic accidents. Incorporating these attributes into a game-theoretic model and building safety analysis tools using such models is an interesting direction of future work. This line of research can result in a safety analysis test bed where a test engineer can configure and tune these typical behavioural attributes. Such a test bed can be built on top of existing microscopic simulation platforms such as SUMO, in which other attributes such as traffic density and road structures are already configurable.

Machine learning and game theoretic models. At the moment most machine learning-based methods for traffic behaviour prediction predict trajectories or manoeuvres of road users. On the other hand, this dissertation identifies that reasoning models are affected by different traffic

situations. In some cases like when a vehicle is about to enter an intersection, QlkR models are observed to be superior, whereas in more strategic situations like at the moment of left or right turns, Nash equilibrium based models are found to be superior. Since having a tabular catalogue of which models works best for which traffic situation is not scalable due to the explosion of possible traffic states, there is a scope of building machine learning models that predict both the reasoning model and the resulting trajectories given any traffic situation.

Computational complexity in dynamic games. General sum, N-player, incomplete information dynamic games are game structures that are well suited for modelling traffic interactions. However, an explicit representation of the game tree can grow exponentially with each time step and the number of players. Solving for all the equilibrium actions over such large game trees can be computationally intractable. In order for game-theoretic models to be applied as a real-time behaviour planner of an AV, we need better techniques to address the computational issue. Monte Carlo based methods are often used to solve dynamic games for planning [200], however; characterising bounded rational behaviour when using approximate techniques can be challenging. One way to address this issue is to take the cost of computation into account in the characterisation of bounded rational behaviour, similar to the approach taken in [124] for mechanism design considering that mechanism design based techniques have already been proposed for AV planning [43]. Another interesting line of research to address the computational problem is also the use of game abstraction [33] combined with behavioural game theoretic approaches for planning real-time AV behaviour.

References

[1] Safety pilot model deployment data - data.gov. https://catalog.data.gov/dataset/safety-pilot-model-deployment-data. Accessed: 2019-02-18.

[2] National Highway Traffic Safety Administration et al. National motor vehicle crash causation survey: Report to congress. *National Highway Traffic Safety Administration Technical Report DOT HS*, 811:059, 2008.

[3] National Highway Traffic Safety Administration et al. MMUCC guideline: Model minimum uniform crash criteria fourth edition.(report no. dot hs 811 631). *Washington, DC: National Highway Traffic Safety Administration*, 2012.

[4] Sydney N Afriat. The construction of utility functions from expenditure data. *International economic review*, 8(1):67–77, 1967.

[5] Michele Aghassi and Dimitris Bertsimas. Robust game theory. *Mathematical programming*, 107(1):231–273, 2006.

[6] Amir Al Jawahiri. Spline-based trajectory generation for autonomous truck-trailer vehicles in low speed highway scenarios. Master's thesis, Delft University of Technology, 2018.

[7] Dina AlAdawy, Michael Glazer, Jack Terwilliger, Henri Schmidt, Josh Domeyer, Bruce Mehler, Bryan Reimer, and Lex Fridman. Eye contact between pedestrians and drivers. *arXiv preprint arXiv:1904.04188*, 2019.

[8] John Amanatides, Andrew Woo, et al. A fast voxel traversal algorithm for ray tracing. In *Eurographics*, volume 87, pages 3–10, 1987.

[9] James Andreoni and John Miller. Giving according to garp: An experimental test of the consistency of preferences for altruism. *Econometrica*, 70(2):737–753, 2002.

[10] Eduardo Arnold, Omar Y Al-Jarrah, Mehrdad Dianati, Saber Fallah, David Oxtoby, and Alex Mouzakitis. A survey on 3d object detection methods for autonomous driving applications. *IEEE Transactions on Intelligent Transportation Systems*, 20(10):3782–3795, 2019.

[11] Robert J Aumann. Rationality and bounded rationality. In *Cooperation: Game-Theoretic Approaches*, pages 219–231. Springer, 1997.

[12] Robert J Aumann and Sylvain Sorin. Cooperation and bounded recall. *Games and Economic Behavior*, 1(1):5–39, 1989.

[13] Razia Azen and Nicole Traxel. Using dominance analysis to determine predictor importance in logistic regression. *Journal of Educational and Behavioral Statistics*, 34(3):319–347, 2009.

[14] Il Bae, Jaeyoung Moon, and Jeongseok Seo. Toward a comfortable driving experience for a self-driving shuttle bus. *Electronics*, 8(9):943, 2019.

[15] Mayank Bansal, Alex Krizhevsky, and Abhijit Ogale. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. *arXiv preprint arXiv:1812.03079*, 2018.

[16] James L Beck and Konstantin M Zuev. Rare-event simulation. *Handbook of uncertainty quantification*, pages 1075–1100, 2017.

[17] Nick Belay. Robot ethics and self-driving cars: how ethical determinations in software will require a new legal framework. *J. Legal Prof.*, 40:119, 2015.

[18] Avner Ben-Ner, Louis Putterman, Fanmin Kong, and Dan Magan. Reciprocity in a two-part dictator game. *Journal of Economic Behavior & Organization*, 53(3):333–352, 2004.

[19] Peter G Bennett. Toward a theory of hypergames. *Omega*, 5(6):749–751, 1977.

[20] Uri Benzion, Amnon Rapoport, and Joseph Yagil. Discount rates inferred from decisions: An experimental study. *Management science*, 35(3):270–284, 1989.

[21] Kush Bhatia, Ashwin Pananjady, Peter Bartlett, Anca Dragan, and Martin J Wainwright. Preference learning along multiple criteria: A game-theoretic perspective. *Advances in neural information processing systems*, 33:7413–7424, 2020.

[22] Abeba Birhane. The impossibility of automating ambiguity. *Artificial Life*, 27(1):44–61, 2021.

[23] David Blackwell. An analog of the minimax theorem for vector payoffs. *Pacific Journal of Mathematics*, 6(1):1–8, 1956.

[24] Jose Blanchet and Michel Mandjes. Rare-event simulation for queues. *Queueing Systems*, 57(2):57–59, 2007.

[25] Henk AP Blom, GJ Bakker, and Jaroslav Krystul. Rare event estimation for a large-scale stochastic hybrid system with air traffic application. *Rare event simulation using Monte Carlo methods*, pages 193–214, 2009.

[26] Glenn Blomquist. A utility maximization model of driver traffic safety behavior. *Accident Analysis & Prevention*, 18(5):371–375, 1986.

[27] Julian Bock, Robert Krajewski, Tobias Moers, Steffen Runde, Lennart Vater, and Lutz Eckstein. The ind dataset: A drone dataset of naturalistic road user trajectories at german intersections. *arXiv preprint arXiv:1911.07602*, 2019.

[28] Matthew L Bolton, Ellen J Bass, and Radu I Siminiceanu. Using formal verification to evaluate human-automation interaction: A review. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 43(3):488–503, 2013.

[29] Guillermo Flores Borda. Game theory and the law: The legal-rules-acceptability theorem. *Derecho & Sociedad*, (36):301–314, 2011.

[30] Maxime Bouton, Alireza Nakhaei, Kikuo Fujimura, and Mykel J Kochenderfer. Scalable decision making with sensor occlusions for autonomous driving. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 2076–2081. IEEE, 2018.

[31] Nancy L Broen and Dean P Chiang. Braking response times for 100 drivers in the avoidance of an unexpected obstacle as measured in a driving simulator. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 40, pages 900–904. SAGE Publications Sage CA: Los Angeles, CA, 1996.

[32] Kyle Brown, Katherine Driggs-Campbell, and Mykel J Kochenderfer. A taxonomy and review of algorithms for modeling and predicting human driver behavior. *arXiv preprint arXiv:2006.08832*, 2020.

[33] Noam Brown and Tuomas Sandholm. Simultaneous abstraction and equilibrium finding in games. In *Twenty-fourth international joint conference on artificial intelligence*, 2015.

[34] Reva Brown. Consideration of the origin of herbert simon's theory of "satisficing"(1933-1947). *Management Decision*, 2004.

[35] Peide Cai, Yuxiang Sun, Yuying Chen, and Ming Liu. Vision-based trajectory planning via imitation learning for autonomous vehicles. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 2736–2742. IEEE, 2019.

[36] Riccardo Calcagno, Yuichiro Kamada, Stefano Lovo, and Takuo Sugaya. Asynchronicity and coordination in common and opposing interest games. *Theoretical Economics*, 9(2):409–434, 2014.

[37] Eduardo F Camacho and Carlos Bordons Alba. *Model predictive control*. Springer science & business media, 2013.

[38] Fanta Camara, Richard Romano, Gustav Markkula, Ruth Madigan, Natasha Merat, and Charles Fox. Empirical game theory of pedestrian interaction for autonomous vehicles. In *Proceedings of Measuring Behavior 2018*, pages 238–244. Manchester Metropolitan University, 2018.

[39] Colin F Camerer. *Behavioral game theory: Experiments in strategic interaction*. Princeton University Press, 2011.

[40] Colin F Camerer, Teck-Hua Ho, and Juin-Kuan Chong. A cognitive hierarchy model of games. *The Quarterly Journal of Economics*, 119(3):861–898, 2004.

[41] Andrea Censi, Konstantin Slutsky, Tichakorn Wongpiromsarn, Dmitry Yershov, Scott Pendleton, James Fu, and Emilio Frazzoli. Liability, ethics, and culture-aware behavior specification using rulebooks. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8536–8542. IEEE, 2019.

[42] Christopher P Chambers, Federico Echenique, and Eran Shmaya. General revealed preference theory. *Theoretical Economics*, 12(2):493–511, 2017.

[43] Rohan Chandra and Dinesh Manocha. Gameplan: Game-theoretic multi-agent planning with human drivers at intersections, roundabouts, and merging. *IEEE Robotics and Automation Letters*, 2022.

[44] Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, et al. Argoverse: 3d tracking and forecasting with rich maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8748–8757, 2019.

[45] Chenyi Chen, Li Li, Jianming Hu, and Chenyao Geng. Calibration of mitsim and idm car-following model based on ngsim trajectory datasets. In *Proceedings of 2010 IEEE International Conference on Vehicular Electronics and Safety*, pages 48–53. IEEE, 2010.

[46] Laurens Cherchye, Thomas Demuynck, and Bram De Rock. Revealed preference analysis of non-cooperative household consumption. *The Economic Journal*, 121(555):1073–1096, 2011.

[47] Eun-Ha Choi. Crash factors in intersection-related crashes: An on-scene perspective. Technical report, National Highway Traffic Safety Administration, 2010.

[48] Jack Clark and Gillian K Hadfield. Regulatory markets for ai safety. *arXiv preprint arXiv:2001.00078*, 2019.

[49] Laurene Claussmann, Marc Revilloud, Dominique Gruyer, and Sébastien Glaser. A review of motion planning for highway autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 21(5):1826–1848, 2019.

[50] Benjamin Coifman and Lizhe Li. A critical evaluation of the next generation simulation (ngsim) vehicle trajectory dataset. *Transportation Research Part B: Methodological*, 105:362–377, 2017.

[51] Andrew M Colman, Briony D Pulford, and Catherine L Lawrence. Explaining strategic coordination: Cognitive hierarchy theory, strong stackelberg reasoning, and team reasoning. *Decision*, 1(1):35, 2014.

[52] Anthony Corso, Robert J Moss, Mark Koren, Ritchie Lee, and Mykel J Kochenderfer. A survey of algorithms for black-box safety validation. *arXiv preprint arXiv:2005.02979*, 2020.

[53] Miguel Costa-Gomes, Vincent P Crawford, and Bruno Broseta. Cognition and behavior in normal-form games: An experimental study. *Econometrica*, 69(5):1193–1235, 2001.

[54] Ian Crawford and Bram De Rock. Empirical revealed preference. *Annu. Rev. Econ.*, 6(1):503–524, 2014.

[55] Vincent P Crawford, Miguel A Costa-Gomes, and Nagore Iriberri. Structural models of nonequilibrium strategic thinking: Theory, evidence, and applications. *Journal of Economic Literature*, 51(1):5–62, 2013.

[56] Frank E Curtis, Olaf Schenk, and Andreas Wächter. An interior-point algorithm for large-scale nonlinear optimization with inexact step computations. *SIAM Journal on Scientific Computing*, 32(6):3447–3475, 2010.

[57] 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.

[58] Florian Damerow, Tim Puphal, Yuda Li, and Julian Eggert. Risk-based driver assistance for approaching intersections of limited visibility. In *2017 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, pages 178–184. IEEE, 2017.

[59] Thomas Demuynck and Per Hjertstrand. Samuelson's approach to revealed preference theory: Some recent advances. *IFN Working Paper No. 1274*, 2019.

[60] Debargha Dey and Jacques Terken. Pedestrian interaction with vehicles: roles of explicit and implicit communication. In *Proceedings of the 9th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, pages 109–113, 2017.

[61] Michael Doumpos and Evangelos Grigoroudis. *Multicriteria decision aid and artificial intelligence: links, theory and applications*. John Wiley & Sons, 2013.

[62] Michael Doumpos and Constantin Zopounidis. Preference disaggregation and statistical learning for multicriteria decision support: A review. *European Journal of Operational Research*, 209(3):203–214, 2011.

[63] Paweł Droździel, Sławomir Tarkowski, Iwona Rybicka, and Rafał Wrona. Drivers' reaction time research in the conditions in the real traffic. *Open Engineering*, 10(1):35–47, 2020.

[64] Paweł Dziewulski. Revealed time preference. *Games and Economic Behavior*, 112:67–77, 2018.

[65] Federico Echenique and Kota Saito. Savage in the market. *Econometrica*, 83(4):1467–1495, 2015.

[66] Adrian B Ellison, Stephen Greaves, and Michiel Bliemer. Examining heterogeneity of driver behavior with temporal and spatial factors. *Transportation research record*, 2386(1):158–167, 2013.

[67] Vladimir Estivill-Castro. Game theory formulation for ethical decision making. In *Robotics and Well-Being*, pages 25–38. Springer, 2019.

[68] Scott Ettinger, Shuyang Cheng, Benjamin Caine, Chenxi Liu, Hang Zhao, Sabeek Pradhan, Yuning Chai, Ben Sapp, Charles R Qi, Yin Zhou, et al. Large scale interactive motion

forecasting for autonomous driving: The waymo open motion dataset. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9710–9719, 2021.

[69] Leonard Evans. Human behavior feedback and traffic safety. *Human factors*, 27(5):555–576, 1985.

[70] Jaime F Fisac, Eli Bronstein, Elis Stefansson, Dorsa Sadigh, S Shankar Sastry, and Anca D Dragan. Hierarchical game-theoretic planning for autonomous vehicles. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 9590–9596. IEEE, 2019.

[71] Peter C Fishburn. Utility theory for decision making. Technical report, Research analysis corp McLean VA, 1970.

[72] János Flesch and Arkadi Predtetchinski. On refinements of subgame perfect $epsilon$ equilibrium. *International Journal of Game Theory*, 45(3):523–542, 2016.

[73] Daniel J Fremont, Edward Kim, Yash Vardhan Pant, Sanjit A Seshia, Atul Acharya, Xantha Bruso, Paul Wells, Steve Lemke, Qiang Lu, and Shalin Mehta. Formal scenario-based testing of autonomous vehicles: From simulation to the real world. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–8. IEEE, 2020.

[74] David Fridovich-Keil, Ellis Ratner, Anca D Dragan, and Claire J Tomlin. Efficient iterative linear-quadratic approximations for nonlinear multi-player general-sum differential games. *arXiv preprint arXiv:1909.04694*, 2019.

[75] Drew Fudenberg and Jean Tirole. *Game theory*. MIT press, 1991.

[76] R Fuller. Recent developments in driver control theory: from task difficulty homeostasis to risk allostasis. In *Proceedings of the International Conference on Traffic and Transport Psychology, Washington, USA*, volume 31, 2008.

[77] Ray Fuller. A conceptualization of driving behaviour as threat avoidance. *Ergonomics*, 27(11):1139–1155, 1984.

[78] Ray Fuller. Towards a general theory of driver behaviour. *Accident analysis & prevention*, 37(3):461–472, 2005.

[79] Functional Safety Committee. Considerations for ISO 26262 ASIL Hazard Classification. Technical report, SAE International, 2018.

[80] Enric Galceran, Edwin Olson, and Ryan M Eustice. Augmented vehicle tracking under occlusions for decision-making in autonomous driving. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3559–3565. IEEE, 2015.

[81] Mario Garzón and Anne Spalanzani. Game theoretic decision making based on real sensor data for autonomous vehicles' maneuvers in high traffic. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5378–5384. IEEE, 2020.

[82] John Geanakoplos. Common knowledge. *Journal of Economic Perspectives*, 6(4):53–82, 1992.

[83] Jack Geary, Subramanian Ramamoorthy, and Henry Gouk. Resolving conflict in decision-making for autonomous driving. *arXiv preprint arXiv:2009.06394*, 2020.

[84] Philipp Geiger and Christoph-Nikolas Straehle. Learning game-theoretic models of multiagent trajectories using implicit layers. *arXiv preprint arXiv:2008.07303*, 2020.

[85] Philipp Geiger and Christoph-Nikolas Straehle. Learning game-theoretic models of multi-agent trajectories using implicit layers. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35:4950–4958, May 2021.

[86] Andrew Gelman and Jennifer Hill. *Data analysis using regression and multilevel/hierarchical models*. Cambridge university press, 2006.

[87] Shane Gilroy, Edward Jones, and Martin Glavin. Overcoming occlusion in the automotive environment—a review. *IEEE Transactions on Intelligent Transportation Systems*, 22(1):23–35, 2019.

[88] Peter G Gipps. A behavioural car-following model for computer simulation. *Transportation Research Part B: Methodological*, 15(2):105–111, 1981.

[89] Jacob K Goeree and Charles A Holt. Ten little treasures of game theory and ten intuitive contradictions. *American Economic Review*, 91(5):1402–1422, 2001.

[90] Russell Golman. Quantal response equilibria with heterogeneous agents. *Journal of Economic Theory*, 146(5):2013–2028, 2011.

[91] John A Groeger. *Understanding driving: Applying cognitive psychology to a complex everyday task*. Routledge, 2013.

[92] Jeffrey Hawke, Richard Shen, Corina Gurau, Siddharth Sharma, Daniele Reda, Nikolay Nikolov, Przemyslaw Mazur, Sean Micklethwaite, Nicolas Griffiths, Amar Shah, et al. Urban driving with conditional imitation learning. *arXiv preprint arXiv:1912.00177*, 2019.

[93] Dirk Helbing, Illés Farkas, and Tamas Vicsek. Simulating dynamical features of escape panic. *Nature*, 407(6803):487, 2000.

[94] Teck-Hua Ho and Xuanming Su. A dynamic level-k model in sequential games. *Management Science*, 59(2):452–469, 2013.

[95] Cars H Hommes. Heterogeneous agent models in economics and finance. *Handbook of computational economics*, 2:1109–1186, 2006.

[96] Constantin Hubmann, Nils Quetschlich, Jens Schulz, Julian Bernhard, Daniel Althoff, and Christoph Stiller. A pomdp maneuver planner for occlusions in urban scenarios. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 2172–2179. IEEE, 2019.

[97] Marko Ilievski. Wisebench: A motion planning benchmarking framework for autonomous vehicles. Master's thesis, University of Waterloo, 2020.

[98] 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.

[99] David Isele, Reza Rahimi, Akansel Cosgun, Kaushik Subramanian, and Kikuo Fujimura. Navigating occluded intersections with autonomous vehicles using deep reinforcement learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2034–2039. IEEE, 2018.

[100] Rebecca Ivers, Teresa Senserrick, Soufiane Boufous, Mark Stevenson, Huei-Yang Chen, Mark Woodward, and Robyn Norton. Novice drivers' risky driving behavior, risk perception, and crash risk: findings from the drive study. *American journal of public health*, 99(9):1638–1644, 2009.

[101] Eric Jacquet-Lagreze and Jean Siskos. Assessing a set of additive utility functions for multicriteria decision-making, the uta method. *European journal of operational research*, 10(2):151–164, 1982.

[102] Eric Jacquet-Lagreze and Yannis Siskos. Preference disaggregation: 20 years of mcda experience. *European Journal of Operational Research*, 130(2):233–245, 2001.

[103] Ang Ji and David Levinson. A review of game theory models of lane changing. *Transportmetrica A: transport science*, 16(3):1628–1647, 2020.

[104] Fredrik Johansson, Anders Peterson, and Andreas Tapani. Waiting pedestrians in the social force model. *Physica A: Statistical Mechanics and its Applications*, 419:95–107, 2015.

[105] I Kageyama and HB Pacejka. On a new driver model with fuzzy control. *Vehicle System Dynamics*, 20(sup1):314–324, 1992.

[106] Daniel Kahneman and Amos Tversky. Prospect theory: An analysis of decision under risk. In *Handbook of the fundamentals of financial decision making: Part I*, pages 99–127. World Scientific, 2013.

[107] Nidhi Kalra and Susan M Paddock. Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability? *Transportation Research Part A: Policy and Practice*, 94:182–193, 2016.

[108] Danial Kamran, Carlos Fernandez Lopez, Martin Lauer, and Christoph Stiller. Risk-aware high-level decisions for automated driving at occluded intersections with reinforcement learning. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, pages 1205–1212. IEEE, 2020.

[109] Alonzo Kelly and Bryan Nagy. Reactive nonholonomic trajectory generation via parametric optimal control. *The International Journal of Robotics Research*, 22(7-8):583–601, 2003.

[110] Esko Keskinen, Mika Hatakka, Sirkku Laapotti, Ari Katila, and M Peräaho. Driver behaviour as a hierarchical system. *Traffic and Transport Psychology*, pages 9–29, 2004.

[111] Beomjoon Kim and Joelle Pineau. Socially adaptive path planning in human environments using inverse reinforcement learning. *International Journal of Social Robotics*, 8(1):51–66, 2016.

[112] B Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A Al Sallab, Senthil Yogamani, and Patrick Pérez. Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[113] Hideyuki Kita. A merging–giveway interaction model of cars in a merging section: a game theoretic analysis. *Transportation Research Part A: Policy and Practice*, 33(3-4):305–312, 1999.

[114] Mark Koren. *Approximate Methods for Validating Autonomous Systems in Simulation*. PhD thesis, Stanford University, 2021.

[115] Markus Koschi and Matthias Althoff. Set-based prediction of traffic participants considering occlusions and traffic rules. *IEEE Transactions on Intelligent Vehicles*, 6(2):249–265, 2020.

[116] Nicholas S Kovach, Alan S Gibson, and Gary B Lamont. Hypergame theory: a model for conflict, misperception, and deception. *Game Theory*, 2015, 2015.

[117] Robert Krajewski, Tobias Moers, Julian Bock, Lennart Vater, and Lutz Eckstein. The round dataset: A drone dataset of road user trajectories at roundabouts in germany. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6, 2020.

[118] Daniel Krajzewicz, Georg Hertkorn, Christian Rössel, and Peter Wagner. Sumo (simulation of urban mobility)-an open-source traffic simulation. In *Proceedings of the 4th middle East Symposium on Simulation and Modelling (MESM20002)*, pages 183–187, 2002.

[119] Jonas Krampe and Mirko Junge. Injury severity for hazard & risk analyses: calculation of iso 26262 s-parameter values from real-world crash data. *Accident Analysis & Prevention*, 138:105321, 2020.

[120] Stefan Krauß, Peter Wagner, and Christian Gawron. Metastable states in a microscopic model of traffic flow. *Physical Review E*, 55(5):5597, 1997.

[121] Volodymyr Kuleshov and Okke Schrijvers. Inverse game theory: Learning utilities in succinct games. In *International Conference on Web and Internet Economics*, pages 413–427. Springer, 2015.

[122] Minae Kwon, Erdem Biyik, Aditi Talati, Karan Bhasin, Dylan P Losey, and Dorsa Sadigh. When humans aren't optimal: Robots that collaborate with risk-aware humans. In *Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*, pages 43–52, 2020.

[123] Kim G Larsen and Axel Legay. Statistical model checking: past, present, and future. In *International Symposium on Leveraging Applications of Formal Methods*, pages 3–15. Springer, 2016.

[124] Kate Larson. *Mechanism design for computationally limited agents*. Carnegie Mellon University, 2004.

[125] William Larson and Weihua Zhao. Self-driving cars and the city: Effects on sprawl, energy consumption, and housing affordability. *Regional Science and Urban Economics*, 81:103484, 2020.

[126] Jean-Patrick Lebacque. First-order macroscopic traffic flow models: Intersection modeling, network modeling. In *Transportation and Traffic Theory. Flow, Dynamics and Human Interaction. 16th International Symposium on Transportation and Traffic TheoryUniversity of Maryland, College Park*, 2005.

[127] Minchul Lee, Myoungho Sunwoo, and Kichun Jo. Collision risk assessment of occluded vehicle based on the motion predictions using the precise road map. *Robotics and Autonomous Systems*, 106:179–191, 2018.

[128] Stéphanie Lefèvre, Dizan Vasquez, and Christian Laugier. A survey on motion prediction and risk assessment for intelligent vehicles. *ROBOMECH journal*, 1(1):1–14, 2014.

[129] Jacqueline P Leighton, Robert J Sternberg, et al. *The nature of reasoning*. Cambridge University Press, 2004.

[130] Neil D Lerner. Brake perception-reaction times of older and younger drivers. In *Proceedings of the human factors and ergonomics society annual meeting*, volume 37, pages 206–210. SAGE Publications Sage CA: Los Angeles, CA, 1993.

[131] Jonathan Levin and Paul Milgrom. Introduction to choice theory. 2004.

[132] Ben Lewis-Evans. *Testing models of driver behaviour*. PhD thesis, University Library Groningen], 2012.

[133] Ben Lewis-Evans, Dick De Waard, and Karel A Brookhuis. That's close enough—a threshold effect of time headway on the experience of risk, task difficulty, effort, and comfort. *Accident Analysis & Prevention*, 42(6):1926–1933, 2010.

[134] Ben Lewis-Evans and Talib Rothengatter. Task difficulty, risk, effort and comfort in a simulated driving task—implications for risk allostasis theory. *Accident Analysis & Prevention*, 41(5):1053–1063, 2009.

[135] Changjian Li and Krzysztof Czarnecki. Urban driving with multi-objective deep reinforcement learning. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, page 359–367, 2019.

[136] Nan Li, Ilya Kolmanovsky, Anouck Girard, and Yildiray Yildiz. Game theoretic modeling of vehicle interactions at unsignalized intersections and application to autonomous vehicle control. In *2018 Annual American Control Conference (ACC)*, pages 3215–3220. IEEE, 2018.

[137] Nan Li, Dave Oyler, Mengxuan Zhang, Yildiray Yildiz, Anouck Girard, and Ilya Kolmanovsky. Hierarchical reasoning game theory based approach for evaluation and testing of autonomous vehicle control systems. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 727–733. IEEE, 2016.

[138] Nan Li, Yu Yao, Ilya Kolmanovsky, Ella Atkins, and Anouck R Girard. Game-theoretic modeling of multi-vehicle interactions at uncontrolled intersections. *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[139] Nan Li, Mengxuan Zhang, Yildiray Yildiz, Ilya Kolmanovsky, and Anouck Girard. Game theory-based traffic modeling for calibration of automated driving algorithms. In *Control Strategies for Advanced Driver Assistance Systems and Autonomous Driving Functions*, pages 89–106. Springer, 2019.

[140] Sisi Li, Nan Li, Anouck Girard, and Ilya Kolmanovsky. Decision making in dynamic and interactive environments based on cognitive hierarchy theory, bayesian inference, and predictive control. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 2181–2187. IEEE, 2019.

[141] Yuying Li. Centering, trust region, reflective techniques for nonlinear minimization subject to bounds. Technical report, Cornell University, 1993.

[142] Ming Liang, Bin Yang, Rui Hu, Yun Chen, Renjie Liao, Song Feng, and Raquel Urtasun. Learning lane graph representations for motion forecasting. In *European Conference on Computer Vision*, pages 541–556. Springer, 2020.

[143] Xiao Lin, Jiucai Zhang, Jin Shang, Yi Wang, Hongkai Yu, and Xiaoli Zhang. Decision making through occluded intersections for autonomous driving. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 2449–2455. IEEE, 2019.

[144] Alexander Liniger and John Lygeros. A noncooperative game approach to autonomous racing. *IEEE Transactions on Control Systems Technology*, 28(3):884–897, 2019.

[145] Chang Liu, Seungho Lee, Scott Varnhagen, and H Eric Tseng. Path planning for autonomous vehicles using model predictive control. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 174–179. IEEE, 2017.

[146] Frank Lovett. A positivist account of the rule of law. *Law & Social Inquiry*, 27(1):41–78.

[147] George J Mailath. Do people play nash equilibrium? lessons from evolutionary game theory. *Journal of Economic Literature*, 36(3):1347–1374, 1998.

[148] Gustav Markkula, Ruth Madigan, Dimitris Nathanael, Evangelia Portouli, Yee Mun Lee, André Dietrich, Jac Billington, Anna Schieben, and Natasha Merat. Defining interactions: A conceptual framework for understanding interactive behaviour in human and automated road traffic. *Theoretical Issues in Ergonomics Science*, 21(6):728–752, 2020.

[149] Robert E Marks. *Repeated games and finite automata*. Citeseer, 1990.

[150] Stephen G McGill, Guy Rosman, Teddy Ort, Alyssa Pierson, Igor Gilitschenski, Brandon Araki, Luke Fletcher, Sertac Karaman, Daniela Rus, and John J Leonard. Probabilistic risk metrics for navigating occluded intersections. *IEEE Robotics and Automation Letters*, 4(4):4322–4329, 2019.

[151] Richard D McKelvey and Thomas R Palfrey. Quantal response equilibria for normal form games. *Games and economic behavior*, 10(1):6–38, 1995.

[152] Alexander J McNeil, Rüdiger Frey, and Paul Embrechts. *Quantitative risk management: concepts, techniques and tools-revised edition*. Princeton university press, 2015.

[153] Juan Medina-Lee, Antonio Artuñedo, Jorge Godoy, and Jorge Villagra. Merit-based motion planning for autonomous vehicles in urban scenarios. *Sensors*, 21(11):3755, 2021.

[154] Jean Mercat, Thomas Gilles, Nicole El Zoghby, Guillaume Sandou, Dominique Beauvois, and Guillermo Pita Gil. Multi-head attention for multi-modal joint vehicle motion forecasting. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9638–9644. IEEE, 2020.

[155] Umberto Michieli and Leonardo Badia. Game theoretic analysis of road user safety scenarios involving autonomous vehicles. In *2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pages 1377–1381. IEEE, 2018.

[156] John A Michon. A critical view of driver behavior models: what do we know, what should we do? In *Human behavior and traffic safety*, pages 485–524. Springer, 1985.

[157] Miloš N Mladenović, Sanna Lehtinen, Emily Soh, and Karel Martens. Emerging urban mobility technologies through the lens of everyday urban aesthetics: case of self-driving vehicle. *Essays in Philosophy*, 20(2):146–170, 2019.

[158] Ranju Mohan and Gitakrishnan Ramadurai. State-of-the art of macroscopic traffic flow modelling. *International Journal of Advances in Engineering Sciences and Applied Mathematics*, 5(2):158–176, 2013.

[159] Charles Musselwhite. Attitudes towards vehicle driving behaviour: Categorising and contextualising risk. *Accident Analysis & Prevention*, 38(2):324–334, 2006.

[160] 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.

[161] Patiphon Narksri, Eijiro Takeuchi, Yoshiki Ninomiya, and Kazuya Takeda. Deadlock-free planner for occluded intersections using estimated visibility of hidden vehicles. *Electronics*, 10(4):411, 2021.

[162] Gordon Frank Newell. A simplified car-following theory: a lower order model. *Transportation Research Part B: Methodological*, 36(3):195–205, 2002.

[163] Andrew Y Ng, Stuart J Russell, et al. Algorithms for inverse reinforcement learning. In *International Conference on Machine Learning*, volume 1, page 2, 2000.

[164] Cao Ningbo, Wei Wei, Qu Zhaowei, Zhao Liying, and Bai Qiaowen. Simulation of pedestrian crossing behaviors at unmarked roadways based on social force model. *Discrete Dynamics in Nature and Society*, 2017, 2017.

[165] David Nistér, Hon-Leung Lee, Julia Ng, and Yizhou Wang. The safety force field. *NVIDIA White Paper*, 2019.

[166] Justin Norden, Matthew O'Kelly, and Aman Sinha. Efficient black-box assessment of autonomous vehicle safety. *arXiv preprint arXiv:1912.03618*, 2019.

[167] Peter D Norton. *Fighting traffic: the dawn of the motor age in the American city*. MIT Press, 2011.

[168] Sven Nyholm. The ethics of crashes with self-driving cars: A roadmap, i. *Philosophy Compass*, 13(7):e12507, 2018.

[169] Matthew O'Kelly, Aman Sinha, Hongseok Namkoong, Russ Tedrake, and John C Duchi. Scalable end-to-end autonomous vehicle testing via rare-event simulation. In *Advances in Neural Information Processing Systems*, pages 9849–9860, 2018.

[170] Piotr F Orzechowski, Annika Meyer, and Martin Lauer. Tackling occlusions & limited sensor range with set-based safety verification. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 1729–1736. IEEE, 2018.

[171] Martin J Osborne and Ariel Rubinstein. Games with procedurally rational players. *American Economic Review*, pages 834–847, 1998.

[172] Saskia Ossen and Serge P Hoogendoorn. Driver heterogeneity in car following and its impact on modeling traffic dynamics. *Transportation Research Record*, 1999(1):95–103, 2007.

[173] Brian Paden, Michal Čáp, 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.

[174] Andrea Palazzi, Davide Abati, Francesco Solera, Rita Cucchiara, et al. Predicting the driver's focus of attention: the dr (eye) ve project. *IEEE transactions on pattern analysis and machine intelligence*, 41(7):1720–1733, 2018.

[175] MR Parker Jr and Charles V Zegeer. Traffic conflict techniques for safety and operations: Observers manual. Technical report, United States. Federal Highway Administration, 1989.

[176] Miguel Perez, Shane McLaughin, Takayuki Kondo, Jonathan Antin, Julie McClafferty, Suzanne Lee, Jonathan Hankey, and Thomas Dingus. Transportation safety meets big data: the shrp 2 naturalistic driving database. *Journal of the Society of Instrument and Control Engineers*, 55(5):415–421, 2016.

[177] Randal C Picker. An introduction to game theory and the law. *Coase-Sandor Institute for Law Economics Working Paper No. 22*, 1994.

[178] Maria Porter and Abi Adams. For love or reward? characterising preferences for giving to parents in an experimental setting. *The Economic Journal*, 126(598):2424–2445, 2016.

[179] Sasinee Pruekprasert, Xiaoyi Zhang, Jérémy Dubut, Chao Huang, and Masako Kishida. Decision making for autonomous vehicles at unsignalized intersection in presence of malicious vehicles. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 2299–2304. IEEE, 2019.

[180] Nijat Rajabli, Francesco Flammini, Roberto Nardone, and Valeria Vittorini. Software verification and validation of safe autonomous cars: A systematic literature review. *IEEE Access*, 9:4797–4819, 2020.

[181] Amir Rasouli, Iuliia Kotseruba, and John K Tsotsos. Agreeing to cross: How drivers and pedestrians communicate. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 264–269. IEEE, 2017.

[182] Stefan Riedmaier, Thomas Ponn, Dieter Ludwig, Bernhard Schick, and Frank Diermeyer. Survey on scenario-based safety assessment of automated vehicles. *IEEE access*, 8:87456–87477, 2020.

[183] Albert Rizaldi, Fabian Immler, Bastian Schürmann, and Matthias Althoff. A formally verified motion planner for autonomous vehicles. In *International Symposium on Automated Technology for Verification and Analysis*, pages 75–90. Springer, 2018.

[184] Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Learning social etiquette: Human trajectory understanding in crowded scenes. In *European conference on computer vision*, pages 549–565. Springer, 2016.

[185] Emmanuel Roche and Yves Schabes. *Finite-state language processing*. MIT press, 1997.

[186] Gerardo Rubino and Bruno Tuffin. *Rare event simulation using Monte Carlo methods*. John Wiley & Sons, 2009.

[187] Ariel Rubinstein. Finite automata play the repeated prisoner's dilemma. *Journal of economic theory*, 39(1):83–96, 1986.

[188] Christopher Thomas Ryan, Albert Xin Jiang, and Kevin Leyton-Brown. Computing pure strategy nash equilibria in compact symmetric games. In *Proceedings of the 11th ACM conference on Electronic commerce*, pages 63–72, 2010.

[189] Dorsa Sadigh, Nick Landolfi, Shankar S Sastry, Sanjit A Seshia, and Anca D Dragan. Planning for cars that coordinate with people: leveraging effects on human actions for planning and active information gathering over human internal state. *Autonomous Robots*, 42(7):1405–1426, 2018.

[190] Dorsa Sadigh, Shankar Sastry, Sanjit A Seshia, and Anca D Dragan. Planning for autonomous cars that leverage effects on human actions. In *Robotics: Science and Systems*, volume 2. Ann Arbor, MI, USA, 2016.

[191] Fridulv Sagberg, Selpi, Giulio Francesco Bianchi Piccinini, and Johan Engström. A review of research on driving styles and road safety. *Human factors*, 57(7):1248–1275, 2015.

[192] Larry Samuelson. Bounded rationality and game theory. *Quarterly Review of Economics and finance*, 36:17–36, 1995.

[193] Paul A Samuelson. A note on the pure theory of consumer's behaviour. *Economica*, 5(17):61–71, 1938.

[194] Werner Sandmann. Rare event simulation methodologies in systems biology. *Rare Event Simulation Using Monte Carlo Methods*, pages 243–266, 2009.

[195] A. Sarkar and K. Czamecki. A behavior driven approach for sampling rare event situations for autonomous vehicles. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6407–6414, 2019.

[196] Atrisha Sarkar and Krzysztof Czarnecki. Solution concepts in hierarchical games under bounded rationality with applications to autonomous driving. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 5698–5708, 2021.

[197] Atrisha Sarkar, Krzysztof Czarnecki, Matt Angus, Changjian Li, and Steven Waslander. Trajectory prediction of traffic agents at urban intersections through learned interactions. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–8. IEEE, 2017.

[198] Georg Schildbach and Francesco Borrelli. Scenario model predictive control for lane change assistance on highways. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, pages 611–616. IEEE, 2015.

[199] Martin Schmidt-Daffy. Prospect balancing theory: Bounded rationality of drivers' speed choice. *Accident Analysis & Prevention*, 63:49–64, 2014.

[200] Wilko Schwarting, Javier Alonso-Mora, and Daniela Rus. Planning and decision-making for autonomous vehicles. *Annual Review of Control, Robotics, and Autonomous Systems*, 2018.

[201] Wilko Schwarting, Alyssa Pierson, Javier Alonso-Mora, Sertac Karaman, and Daniela Rus. Social behavior for autonomous vehicles. *Proceedings of the National Academy of Sciences*, 116(50):24972–24978, 2019.

[202] Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. On a formal model of safe and scalable self-driving cars. *arXiv preprint arXiv:1708.06374*, 2017.

[203] Yoav Shoham and Kevin Leyton-Brown. *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press, 2008.

[204] Herbert A Simon. Theories of bounded rationality. decision and organization. *CBR a. R. Radner. Amsterdam, NorthHolland*, 1972.

[205] Dale O Stahl and Paul W Wilson. On players models of other players: Theory and experimental evidence. *Games and Economic Behavior*, 10(1):218–254, 1995.

[206] Jack Stilgoe. Machine learning, social learning and the governance of self-driving cars. *Social studies of science*, 48(1):25–56, 2018.

[207] Wynn C Stirling. *Satisficing Games and Decision Making: with applications to engineering and computer science*. Cambridge University Press, 2003.

[208] Mark Sullman. *Advances in traffic psychology*. CRC Press, 2019.

[209] Heikki Summala. Risk control is not risk adjustment: The zero-risk theory of driver behaviour and its implications. *Ergonomics*, 31(4):491–506, 1988.

[210] Liting Sun, Mu Cai, Wei Zhan, and Masayoshi Tomizuka. A game-theoretic strategy-aware interaction algorithm with validation on real traffic data. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11038–11044. IEEE, 2020.

[211] Liting Sun, Wei Zhan, Masayoshi Tomizuka, and Anca D Dragan. Courteous autonomous cars. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 663–670. IEEE, 2018.

[212] Nilesh Suriyarachchi, Rohan Chandra, John S. Baras, and Dinesh Manocha. Gameopt: Optimal real-time multi-agent planning and control at dynamic intersections, 2022.

[213] Steven Tadelis. *Game theory: an introduction*. Princeton University Press, 2013.

[214] Thompson Products. Want to build a car that drives itself? http://www.fulltable.com/VTS/f/fut/f/cars/SH435.jpg, 1958. Thompson products ad from 1958. Accessed: 2022-03-31.

[215] Ran Tian, Sisi Li, Nan Li, Ilya Kolmanovsky, Anouck Girard, and Yildiray Yildiz. Adaptive game-theoretic decision making for autonomous vehicle control at roundabouts. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 321–326. IEEE, 2018.

[216] Ran Tian, Sun Liting, Tomizuka Masayoshi, and Isele David. Anytime game-theoretic planning with active reasoning about humans' latent states for human-centered robots. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021.

[217] Ran Tian, Liting Sun, Andrea Bajcsy, Masayoshi Tomizuka, and Anca D Dragan. Safety assurances for human-robot interaction via confidence-aware game-theoretic human models. *arXiv preprint arXiv:2109.14700*, 2021.

[218] Martin Treiber, Ansgar Hennecke, and Dirk Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical review E*, 62(2):1805, 2000.

[219] Cumhur Erkan Tuncali, Georgios Fainekos, Danil Prokhorov, Hisahiro Ito, and James Kapinski. Requirements-driven test generation for autonomous vehicles with machine learning components. *IEEE Transactions on Intelligent Vehicles*, 5(2):265–280, 2019.

[220] Truls Vaa. Drivers' information processing, decision-making and the role of emotions: Predictions of the risk monitor model. In *Human Modelling in Assisted Transportation*, pages 23–32. Springer, 2011.

[221] Hugo H Van der Molen and Anton MT Bötticher. A hierarchical risk model for traffic participants. *Ergonomics*, 31(4):537–555, 1988.

[222] Hal R Varian. Revealed preference and its applications. *The Economic Journal*, 122(560):332–338, 2012.

[223] John Von Neumann and Oskar Morgenstern. Theory of games and economic behavior, 2nd rev. 1947.

[224] Jingkang Wang, Ava Pun, James Tu, Sivabalan Manivasagam, Abbas Sadat, Sergio Casas, Mengye Ren, and Raquel Urtasun. Advsim: Generating safety-critical scenarios for self-driving vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9909–9918, 2021.

[225] Mingyu Wang, Zijian Wang, John Talbot, J Christian Gerdes, and Mac Schwager. Game theoretic planning for self-driving cars in competitive scenarios. In *Robotics: Science and Systems*, 2019.

[226] Muhong Wang, Keith W Hipel, and Niall M Fraser. Modeling misperceptions in games. *Behavioral Science*, 33(3):207–223, 1988.

[227] Pin Wang, Ching-Yao Chan, and Arnaud de La Fortelle. A reinforcement learning based approach for automated lane change maneuvers. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1379–1384. IEEE, 2018.

[228] Junqing Wei, John M Dolan, and Bakhtiar Litkouhi. A prediction-and cost function-based algorithm for robust autonomous freeway driving. In *2010 IEEE Intelligent Vehicles Symposium*, pages 512–517. IEEE, 2010.

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

[230] Katarzyna M Werner and Horst Zank. A revealed reference point for prospect theory. *Economic Theory*, 67(4):731–773, 2019.

[231] Tim Allan Wheeler and Mykel J Kochenderfer. Critical factor graph situation clusters for accelerated automotive safety validation. In *2019 IEEE Intelligent Vehicles Symposium*, pages 2133–2139. IEEE, 2019.

[232] Meredith Whittaker, Meryl Alper, Cynthia L Bennett, Sara Hendren, Liz Kaziunas, Mara Mills, Meredith Ringel Morris, Joy Rankin, Emily Rogers, Marcel Salas, et al. Disability, bias, and ai. *AI Now Institute*, 2019.

[233] Gerald JS Wilde. The theory of risk homeostasis: implications for safety and health. *Risk analysis*, 2(4):209–225, 1982.

[234] James R Wright and Kevin Leyton-Brown. Beyond equilibrium: Predicting human behavior in normal-form games. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010.

[235] James R Wright and Kevin Leyton-Brown. Behavioral game theoretic models: a bayesian framework for parameter analysis. In *AAMAS*, pages 921–930, 2012.

[236] James R Wright and Kevin Leyton-Brown. Level-0 meta-models for predicting human behavior in games. In *Proceedings of the fifteenth ACM conference on Economics and computation*, pages 857–874, 2014.

[237] James R Wright and Kevin Leyton-Brown. A formal separation between strategic and nonstrategic behavior. In *Proceedings of the 21st ACM Conference on Economics and Computation*, pages 535–536, 2020.

[238] Dongfang Yang, Linhui Li, Keith Redmill, and Ümit Özgüner. Top-view trajectories: A pedestrian dataset of vehicle-crowd interaction from controlled experiments and crowded campus. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 899–904. IEEE, 2019.

[239] Hsin-Hsiang Yang and Huei Peng. Development and evaluation of collision warning/collision avoidance algorithms using an errable driver model. *Vehicle system dynamics*, 48(S1):525–535, 2010.

[240] Mehmet Yilmaz and Buse Buyum. Parameter estimation methods for two-component mixed exponential distributions. *İstatistik Türk İstatistik Derneği Dergisi*, 8(2):51–59.

[241] Hongtao Yu, H Eric Tseng, and Reza Langari. A human-like game theory-based controller for automatic lane changing. *Transportation Research Part C: Emerging Technologies*, 88:140–158, 2018.

[242] Ming-Yuan Yu, Ram Vasudevan, and Matthew Johnson-Roberson. Occlusion-aware risk assessment for autonomous driving in urban environments. *IEEE Robotics and Automation Letters*, 4(2):2235–2241, 2019.

[243] Shengcheng Yuan, Yi Liu, Hui Zhang, Soon Ae Chun, and Nabil R Adam. Agent driving behavior modeling for traffic simulation and emergency decision support. In *2015 Winter Simulation Conference (WSC)*, pages 312–323. IEEE, 2015.

[244] Roman Zakharenko. Self-driving cars will change cities. *Regional science and urban economics*, 61:26–37, 2016.

[245] E Christopher Zeeman. On the unstable behaviour of stock exchanges. *Journal of mathematical economics*, 1(1):39–49, 1974.

[246] Weiliang Zeng, Peng Chen, Hideki Nakamura, and Miho Iryo-Asano. Application of social force model to pedestrian behavior analysis at signalized crosswalk. *Transportation research part C: emerging technologies*, 40:143–159, 2014.

[247] Wei Zhan, Liting Sun, Di Wang, Haojie Shi, Aubrey Clausse, Maximilian Naumann, Julius Kümmerle, Hendrik Königshof, Christoph Stiller, Arnaud de La Fortelle, and Masayoshi Tomizuka. INTERACTION Dataset: An INTERnational, Adversarial and Cooperative moTION Dataset in Interactive Driving Scenarios with Semantic Maps. *arXiv:1910.03088 [cs, eess]*, September 2019.

[248] Zixu Zhang and Jaime F Fisac. Safe occlusion-aware autonomous driving via game-theoretic active perception. *arXiv preprint arXiv:2105.08169*, 2021.

[249] Ding Zhao, Yaohui Guo, and Yunhan Jack Jia. Trafficnet: An open naturalistic driving scenario library. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–8. IEEE, 2017.

[250] Ding Zhao, Henry Lam, Huei Peng, Shan Bao, David J LeBlanc, Kazutoshi Nobukawa, and Christopher S Pan. Accelerated evaluation of automated vehicles safety in lane-change scenarios based on importance sampling techniques. *IEEE transactions on intelligent transportation systems*, 18(3):595–607, 2017.

[251] 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.

[252] Markus Zimmermann, David Schopf, Niklas Lütteken, Zhengzhenni Liu, Konrad Storost, Martin Baumann, Riender Happee, and Klaus J Bengler. Carrot and stick: A game-theoretic approach to motivate cooperative driving through social interaction. *Transportation Research Part C: Emerging Technologies*, 88:159–175, 2018.

APPENDICES

Appendix A

Appendix

A.1 Trajectory generation

In order to generate realistic future movements of road users, I use a lattice based trajectory generation process. A planing lattice is a discretization of the continuous planning space along three dimensions, the spatial lattices $(X, Y)_L$ and the velocity lattice V_L. The spatial lattice is constructed on a $\mathbf{R}^2$ Cartesian coordinate system with unit in metres. The resolution of $(X, Y)_L$ depends on the task of the vehicle. Planning over a higher resolution lattice is computationally more demanding, but can generate trajectories with higher accuracy (with respect to the target reference points) compared to a lattice with lower resolution. For the roundabout, left turn, right turn, and crosswalk navigation tasks, I use a resolution of 0.5 metres, and for straight-through navigation at intersections, I use a resolution of 5 metres for $(X, Y)_L$. For pedestrians, the resolution $(X, Y)_L$ is also 0.5 metres. I use a higher resolution lattice for tasks that need fine grained control over the planned future paths. The velocity lattice V_L is constructed with a resolution of 0.3 ms^{-1}.

A.1.1 Trajectory constraints

Proceed trajectory

The input to the trajectory generation process is in the form of two sets of constraints for each of the lattices. The spatial constraints are the reference points on $(X, Y)_L$ that represent the centerline along which the vehicle path needs to be generated. Depending on the sampling scheme

presented in Chapter 3, i.e., S(1), S(1+B), S(1+G), the reference centerline can be constructed in different ways. The velocity constraints on the trajectory generation process map each reference point in $(X, Y)_L$ to a range of target velocities for that reference point. Formally, a velocity constraint $\mathcal{C}_v : (X, Y)_L \rightarrow V_L^2$ maps a reference point in $(X, Y)_L$ to a point in V_L^2 that represents the upper and lower bound of the target velocities corresponding to that point. It is not necessary to specify the constraint explicitly for every reference point. For the reference points with missing constraints, I interpolate the constraint velocity range from the ones for which the range has been provided.

The input constraints for the trajectories that must be generated under the *wait* manoeuvre are expressed in terms of the stopping distance range ($\mathcal{C}_d^{v=0} \in \mathbf{R}^2$) and the stopping time range ($\mathcal{C}_t^{v=0} \in \mathbf{R}^2$), where $\mathcal{C}_d^{v=0}$ and $\mathcal{C}_t^{v=0}$ represent the distance range in metres and the time range in seconds at which the vehicle comes to a stop.

A.1.2 Path generation

Let $\mathcal{C}_X = [x_0, ..., x_K]$ and $\mathcal{C}_Y = [y_0, ..., y_K]$ be the x-axis and y-axis reference points of the path centerline. Two second degree univariate splines, $p_X(s), p_Y(s)$ with smoothing factor $\frac{1}{|\mathcal{C}_X|}$ are fit to the two sets of reference points as follows.

$$p_X^{\text{ref}} = \{(x_i, s_i) : x_i = \mathcal{C}_{X,i}; s_i = \frac{\sum_{j=0}^{i} \|(x_j, y_j) - (x_0, y_0))\|^2}{\sum_{j=0}^{k} \|(x_j, y_j) - (x_0, y_0))\|^2}\}$$

$$p_Y^{\text{ref}} = \{(y_i, s_i) : y_i = \mathcal{C}_{X,i}; s_i = \frac{\sum_{j=0}^{i} \|(x_j, y_j) - (x_0, y_0))\|^2}{\sum_{j=0}^{k} \|(x_j, y_j) - (x_0, y_0))\|^2}\}$$

The points $s \in [0, 1]$ upon which the two splines are fit is a reference axis constructed as the proportion of the planning horizon in terms of the Euclidean distance measured along the centerline. Let s^{ref} be the points in $[0, 1]$ that are used to construct the set p_X^{ref} and p_Y^{ref}. The final generated path of the vehicle is $\{(p_X(s), p_Y(s))\}$ where $s \in [0, 1]$. The curvature of the path can be calculated at any point s using the formula $\kappa(s) = \frac{x'y'' - y'x''}{(x'^2 + y'^2)^{\frac{3}{2}}}$, where $x' = \frac{\mathrm{d}p_X}{\mathrm{d}s}$, $x'' = \frac{\mathrm{d}^2 p_X}{\mathrm{d}s^2}$ and $y' = \frac{\mathrm{d}p_Y}{\mathrm{d}s}$, $y'' = \frac{\mathrm{d}^2 p_Y}{\mathrm{d}s^2}$ are the first and second order derivatives of the splines with respect to the reference axis s. We can also calculate the arc length of the generated path at any point in s as

follows.

$$l(s) = \int_0^s (x'(s)^2 + y'(s)^2)^{\frac{1}{2}}\, ds$$

A.1.3 Trajectory generation

Proceed trajectory

The next step in the trajectory generation process is the construction of velocity profiles. Recall that $\mathcal{C}_v$ maps each reference point in $\mathcal{C}_\mathcal{X} \times \mathcal{C}_\mathcal{Y}$ to a set of points in the velocity lattice. Since the path generated in the previous step is over the reference axis s, in order to generate a trajectory over a time axis, we map the axis s to a time axis t. The time scaling is performed with the following sequence of steps.

- *Step 1.* A cubic spline $\phi(l(s)) : [0, 1] \rightarrow \mathbf{R}$ is fitted based on the velocity lattice points in the reference s^{ref}, which maps the arc length $l(s)$ to a velocity value. This represents a velocity profile with respect to the path length from the origin.

- *Step 2.* A set of time reference points $t_i^{\text{ref}} = \sum_{j=0}^{i} \frac{2 \cdot |l(s_{j+1}) - l(s_j)|}{\phi(l(s_j)) + \phi(l(s_{j+1}))}; i \in \{0, .., K-1\}$ is constructed for each point s_i in s^{ref} that represents the time taken by the vehicle to reach a reference point s_i following the velocity profile $\phi(l(s))$ on the generated path.

- *Step 3.* Finally, another cubic spline $v(t)$ is constructed with the time reference points t^{ref} from the previous step with the same target velocities in the velocity lattice as used in the construction of $\phi(l(s))$.

The above steps establish a one-to-one correspondence between the axis s and the time axis t. For any arbitrary time point t, we can calculate the corresponding value on the s axis as $\frac{\int_0^t v(t)\, dt}{l(1)}$. The integral gives the length traversed by the vehicle in time t, $l(1)$ is the total length of the trajectory until the end of the planning horizon, and the ratio gives the proportion that the vehicle travels in time t. With some compromise in notation, let us refer to this conversion as $s(t)$. Therefore, the final trajectory is calculated at a frequency of 10 Hz. where each trajectory point at time t is given by

time = t,
position = ($p_X(s(t))$,$p_Y(s(t))$),
velocity = $v(t)$,

longitudinal acceleration = $v'(t)$,
lateral acceleration = $v(t)^2 \cdot \kappa(s(t))$,
jerk = $v''(t)$,
yaw = *atan2*$(p'_Y(s(t)), p'_X(s(t)))$

A final set of constraints on velocity, longitudinal and lateral acceleration, jerk, and yaw, taken from [14] selects the set of trajectories that are considered executable by the vehicle.

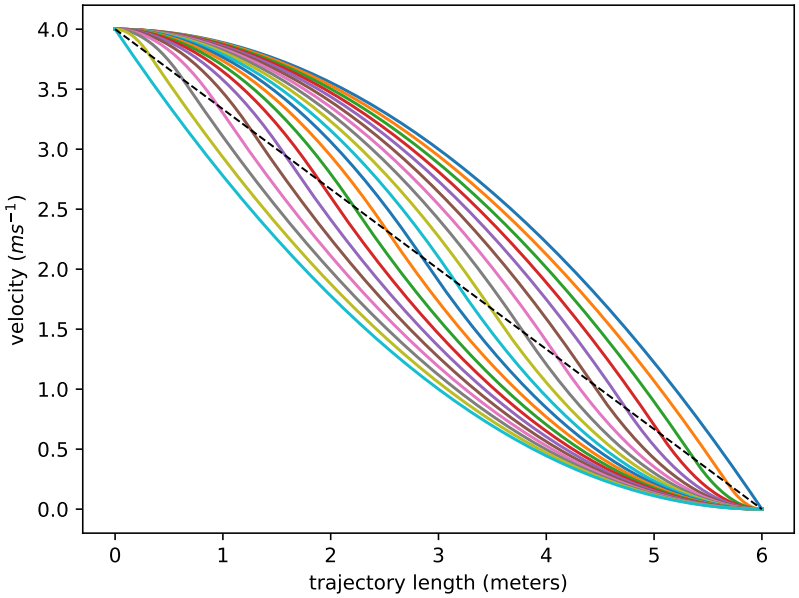


Figure A.1: Sample deceleration profiles of a vehicle moving at 4 ms^{-1} with a target stopping distance of 6 meters. The convexity parameter θ for the different profiles are shown as the black dashed line.

Wait trajectory

The trajectory generation for trajectories corresponding to the *wait* manoeuvre is identical with the exception of *Step 1*. Instead of using a cubic spline to model each velocity profile, I use a

sigmoidal shaped curve as follows:

$$v(l,\theta) = \begin{cases} v^{t=0} \cdot \frac{2(d^{v=0}-l)}{d^{v=0}\cdot\theta}, & \text{if } d^{v=0} - l \leqslant \theta \\ v^{t=0} \cdot \frac{2l}{d^{v=0}\cdot(d^{v=0}-\theta)}, & \text{otherwise} \end{cases}$$

where $d^{v=0}$ is the desired stopping distance, l is the trajectory length, $v^{t=0}$ is the starting velocity and θ is a convexity parameter that controls the shape of the curve in a way that when $\theta \leqslant d^{v=0}-l$, the deceleration profile is convex and for $\theta > d^{v=0}-l$, the profile is concave. I use different values of $\theta \in [0, d^{v=0}]$ to generate the velocity profiles. Fig. A.1 shows 20 such deceleration profiles for a desired stopping distance of 6 metres and $v^{t=0} = 4ms^{-1}$. The deceleration profile curve formulation is an adaptation of the cumulative density function of a Triangular distribution. In summary, the set of steps for the generation of vehicle trajectories consists of the following steps.

- 1. Build the lattice $(X, Y)_L$ and V_L with chosen resolution.
- 2. Setup constraints $\mathcal{C}_X$,$\mathcal{C}_Y$ for the path, $\mathcal{C}_v$ for the velocity profiles corresponding to the *proceed* manoeuvre, and $\mathcal{C}_d^{v=0}$ for stopping distance of the *wait* manoeuvres.
- 3. Generate path based on constraints $\mathcal{C}_X$ and $\mathcal{C}_Y$.
- 4. Generate trajectory based on constraints $\mathcal{C}_v$ and $\mathcal{C}_d^{v=0}$
- 5. Select valid trajectories based on velocity, acceleration, jerk, and yaw limits.

For the generation of trajectories of pedestrian movements, I use a constant acceleration (CA) movement model. In this model, first an acceleration / deceleration value is sampled from a uniform distribution in the range of [0.1,0.5] ms^{-2}. Next, a point mass kinematic motion model is constructed to follow the sampled acceleration / deceleration value until the pedestrian stops (under the *wait* manoeuvre) or reaches the target walking velocity. The target walking velocity is sampled from an uniform distribution in the range [0.8, 2.3] ms^{-1}. The sampling ranges are based on the observed ranges in the crosswalk dataset.

A.2 Rules table for manoeuvres

Chapter 3 uses a set of rules to assign a high level manoeuvre based on the traffic segment, the state of the traffic signal, the task and other conditions such as presence of pedestrians, a lead vehicle or an oncoming vehicle. The assignment of high-level maneuvers based on the condition

is shown in Table A.1. The columns SEGMENT and MANEUVER are self explanatory, TRAFFIC SIGNAL state is one of Green (G), Red (R), Any (*), and presence of other road users is shown as Yes (Y), No (N), Any (*). The condition Any (*) is a free variable that returns true for all values.

SEGMENT	MANEUVER	TRAFFIC SIGNAL	TASK	LEAD PRESENT, PEDESTRIAN PRESENT, ONCOMING VEHICLE PRESENT	IS RULE
prep-left-turn	wait-for-oncoming	G	LEFT_TURN	N,N,Y	Y
exec-left-turn	proceed-turn	*	LEFT_TURN	N,N,N	Y
left-turn-lane	decelerate-to-stop	R	LEFT_TURN	*,*,*	Y
prep-left-turn	proceed-turn	G	LEFT_TURN	N,N,N	Y
left-turn-lane	wait_for_lead_to_cross	G	LEFT_TURN	Y,*,*	Y
exit-lane	follow_lead	*	*	Y,*,*	Y
left-turn-lane	proceed-turn	G	LEFT_TURN	N,N,N	Y
left-turn-lane	follow_lead_into_intersection	G	LEFT_TURN	Y,N,N	Y
prep-left-turn	wait_for_lead_to_cross	G	LEFT_TURN	Y,*,*	Y
exec-left-turn	wait_for_lead_to_cross	G	LEFT_TURN	Y,*,*	Y
left-turn-int-entry	wait-on-red	R	LEFT_TURN		
left-turn-int-entry	proceed-turn	G	LEFT_TURN		
left-turn-int-entry	wait-for-oncoming	G	LEFT_TURN		
left-turn-int-entry	wait_for_lead_to_cross	G	LEFT_TURN		
exec-left-turn	wait-for-oncoming	G	LEFT_TURN	*,*,Y	Y
through-lane-entry	decelerate-to-stop	R	STRAIGHT	*,*,*	Y
through-lane-entry	follow_lead	G	STRAIGHT	Y,*,*	Y
through-lane	follow_lead	*	STRAIGHT	Y,*,*	Y
through-lane-entry	wait-on-red	R	STRAIGHT	*,*,*	Y
left-turn-lane	wait-on-red	R	LEFT_TURN	*,*,*	Y
exit-lane	track_speed	*	*	*,N,*	Y
through-lane-entry	track_speed	G	STRAIGHT	N,*,*	Y
through-lane	track_speed	*	STRAIGHT	N,*,*	Y
through-lane-entry	cut-in	G	LEFT_TURN		
left-turn-lane	yield-to-merging	*	LEFT_TURN		
right-turn-lane	decelerate-to-stop	*	RIGHT_TURN		
right-turn-lane	wait_for_lead_to_cross	*	RIGHT_TURN	Y,*,*	Y
right-turn-lane	proceed-turn	*	RIGHT_TURN	N,N,N	Y
prep-right-turn	wait-for-oncoming	*	RIGHT_TURN	N,N,Y	Y
prep-right-turn	proceed-turn	*	RIGHT_TURN	N,N,N	Y
prep-right-turn	wait_for_lead_to_cross	*	RIGHT_TURN	Y,*,*	Y
exec-right-turn	proceed-turn	*	RIGHT_TURN	N,N,N	Y
exec-right-turn	wait_for_lead_to_cross	*	RIGHT_TURN	Y,N,*	Y
exec-right-turn	wait-for-oncoming	*	RIGHT_TURN	*,*,Y	Y
right-turn-lane	follow_lead_into_intersection	*	RIGHT_TURN		
right-turn-lane	wait-for-pedestrian	*	RIGHT_TURN	*,Y,*	Y
prep-right-turn	wait-for-pedestrian	*	RIGHT_TURN	*,Y,*	Y
prep-left-turn	wait-for-pedestrian	G	LEFT_TURN	*,Y,*	Y
exec-left-turn	wait-for-pedestrian	*	LEFT_TURN	*,Y,*	Y
left-turn-lane	wait-for-oncoming	G	LEFT_TURN	N,*,Y	Y
exec-right-turn	follow_lead_into_intersection	*	RIGHT_TURN	Y,N,*	
prep-right-turn	follow_lead_into_intersection	*	RIGHT_TURN		
prep-left-turn	follow_lead_into_intersection	*	LEFT_TURN		
exec-left-turn	follow_lead_into_intersection	*	LEFT_TURN		
through-lane-entry	wait_for_lead_to_cross	G	LEFT_TURN		
through-lane-entry	proceed-turn	G	LEFT_TURN		
through-lane-entry	wait-for-oncoming	G	LEFT_TURN		
left-turn-lane	wait-for-pedestrian	G	LEFT_TURN	N,Y,*	Y
exec-right-turn	wait-for-pedestrian	*	RIGHT_TURN	N,Y,*	Y
exec-left-turn	proceed-turn	R	LEFT_TURN	*,N,*	Y
exec-left-turn	proceed-turn	Y	LEFT_TURN	*,N,*	Y
prep-left-turn	decelerate-to-stop	Y	LEFT_TURN	*,Y,*	Y
prep-left-turn	proceed-turn	Y	LEFT_TURN	*,Y,Y	Y
prep-left-turn	decelerate-to-stop	R	LEFT_TURN	*,*,*	Y
prep-left-turn	decelerate-to-stop	Y	LEFT_TURN	*,*,Y	Y
prep-left-turn	proceed-turn	Y	LEFT_TURN	Y,N,N	Y
exec-right-turn	wait-for-pedestrian	*	RIGHT_TURN	*,Y,*	Y

Table A.1: Rule table for generating manoeuvres for vehicles in Chapter 3

Appendix B

Waterloo multi-agent traffic dataset

B.1 Introduction

A main goal of the dissertation is the study of interactive road user behaviourbehaviour in a naturalistic setting using game theoretic models. A necessary artefact to achieve that goal is access to naturalistic data that capture interactive behaviour among a group of road users in traffic situations that may involve strategic reasoning. At the time when the work on the dissertation started, existing large naturalistic datasets such as SHRP2 Naturalistic Driving Study (NDS) captured driving of a single driver and had little information about the trajectories of all other road users in the vicinity, which is necessary for the creation of interactive models. Other datasets such as The Next Generation Simulation (NGSIM) dataset had limited coverage, and the accuracy in terms of trajectory information was lower than that needed for the evaluation of game-theoretic models [50].

To address the gap in the availability of an accessible dataset for the study of interactive road user behavior in a naturalistic setting, one of the contributions of this thesis is the Waterloo multi-agent (WMA) traffic dataset. The dataset was recorded at three locations in the city of Waterloo, Ontario, with the help of an overhead drone. The locations covered a signalised intersection (University and Weber), a roundabout (Ira Needles Boulevard and Erb Street West) and a pedestrian crosswalk at the University of Waterloo. The choice of locations was due to the intersection and roundabout being the ones with the highest number of crashes for their respective traffic scenarios in the city of Waterloo [1] [2]. The three datasets that are part of the WMA traffic dataset are

[1] https://spokeonline.com/2018/11/the-most-dangerous-intersections-in-waterloo-region/ [2] https://www.northumberlandnews.com/news-story/6140022-road-ahead-roundabouts-are-safe-if-you-re-happy-with-lots-of-fender-benders/

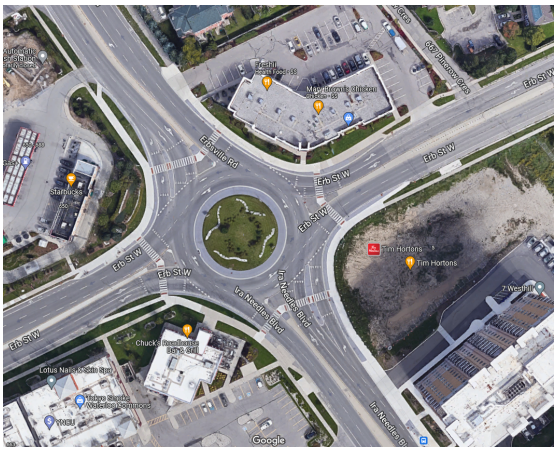
Dataset	Location	No. of road users	Road user types	Relevant Agent information	Lane conflict information
DUT/CITR [238]	United States	2202	Vehicles, pedestrians, golf carts	No	No
inD [27], rounD [117]	Germany	25246	Vehicles, pedestrians, bicycles	No	No
Stanford Drone Dataset [184]	United States	10240	Vehicles, pedestrians, bicycles	No	No
Interaction dataset [247]	United States, Germany, China	40054	Vehicles, pedestrians	No	No
Waterloo multi-agent dataset	**Canada**	**11985**	**Vehicles, pedestrians, bicycles**	**Yes**	**Yes**

Table B.1: Details of other naturalistic multiagent datasets.

available to anyone under the Creative Commons Attribution-NonCommercial 4.0 International licence. The dataset, annotated videos, and documentation of the schema can be found at http://wiselab.uwaterloo.ca/waterloo-multi-agent-traffic-dataset/.

Multiagent datasets provide complete trajectory information about a group of interacting road users for a traffic scenario. Due to advances in aerial photography, drones have been an effective method for capturing naturalistic traffic data. During the time of development of the WMA dataset, there were other similar datasets released using a drone-based data collection method. Table B.1 provides an overview of the other available datasets of similar nature, including the Stanford interaction dataset [184], the interaction dataset [247], and the xD set of datasets [27, 117]. Compared to some of the larger datasets. WMA has a relatively lower number of road users. This isn't a reflection of the traffic density of the chosen locations, but rather WMA dataset is shorter in duration (3 hours) compared to other ones, most notable of which is the interaction dataset spanning 16 hours total. However, the WMA data set contains other information, such as relevant agent information and lane conflict information. This set of information is essential in the creation of game-theoretic models of traffic behaviour, and I discuss these attributes in detail in Section B.3.

(a) Intersection location at University Ave. and Weber Street.



(b) Roundabout location at Ira Needles boulevard and Erb Street

(c) Crosswalk location at University of Waterloo

Figure B.1: Location of the three datasets in the Waterloo multiagent traffic dataset.

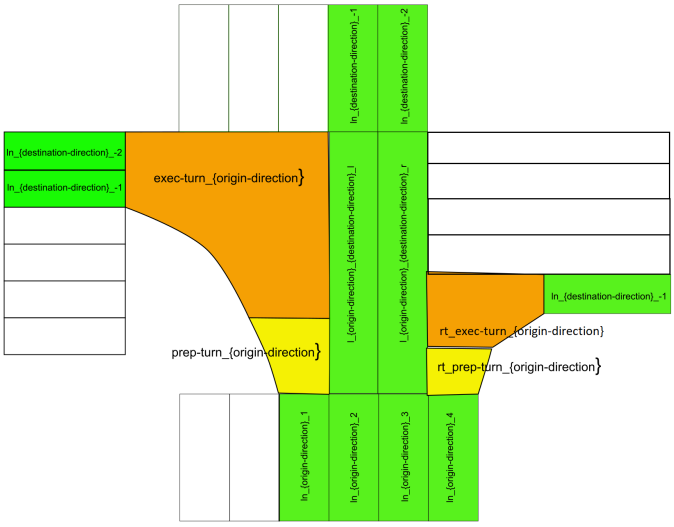


Figure B.2: Lane segments in the intersection dataset. Entry lanes are coded with a suffix of positive integers and exit lanes are coded with a suffix of negative integers. The turn segments are shown for south to west left turn and south to east right turn.

B.2 Data collection and processing

Data were collected using an overhead drone during midday traffic at the three locations in Waterloo, Ontario, shown in Fig. B.1. The drone operation was carried out by a licenced drone operator [3] approved by Transport Canada. The collection method did not include any identifying information about the individuals, such as their faces, clothing, or physical features. Nevertheless, in order to protect the privacy of individuals, attention was paid to placing the drone at a height that minimises the recording of such information. The raw drone footage was then sent to a commercial company[4], for the detection of the type of traffic user (vehicles, buses, pedestrians, motorcycles, bicycles, etc.) and their geolocated trajectories.

B.3 Lane and conflict relations

The traffic map of each of the three locations in the dataset is divided into a set of lane segments. For the intersection and roundabout dataset, the datasets also contain additional information on whether two lane sequences are in conflict with each other.

[3] http://hawkeyefilms.ca/ [4] https://datafromsky.com/

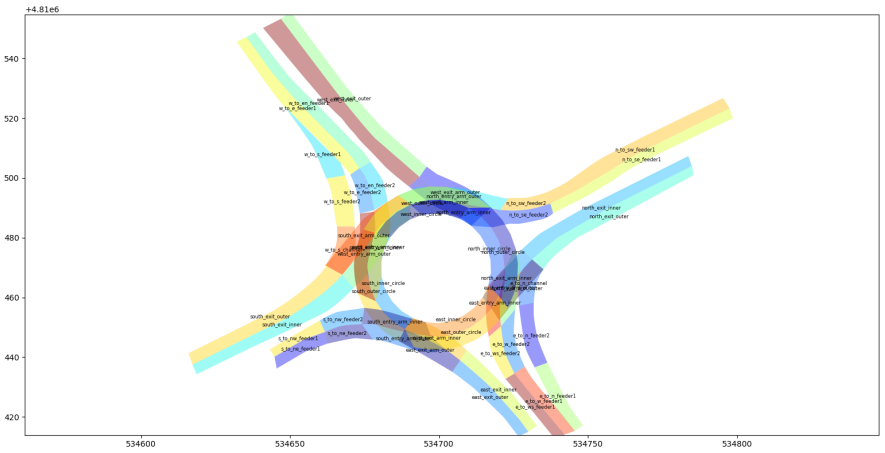


Figure B.3: Lane segments in the roundabout dataset.

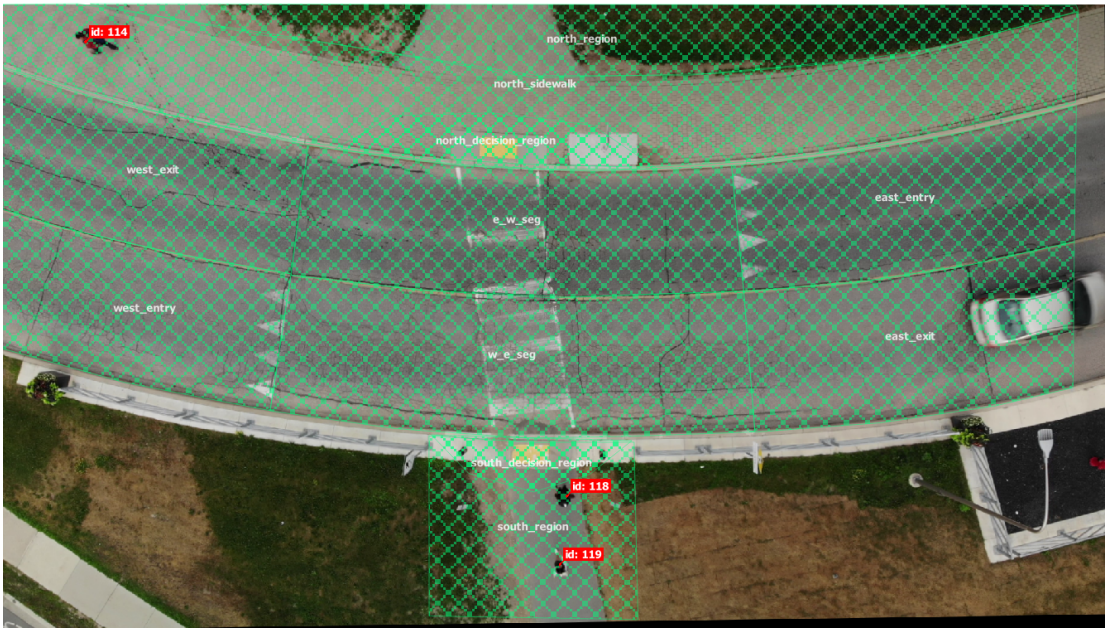


Figure B.4: Lane segments in the crosswalk dataset.

B.3.1 Intersection dataset lane information

Fig. B.2 shows all lane segments in the intersection dataset. The table `CONFLICT_POINTS` (schema shown in Table B.2) lists all conflict points in the intersection scenario. Conflict points belong to two or more paths (a sequence of lane segments) that are in conflict with each other. Currently the data definition supports only conflict between two paths. If there are multiple lanes in conflict, there is one conflict point for all two-path combinations of the conflicting lanes.

B.3.2 Roundabout dataset lane conflict information

Fig. B.3 shows all the lane segments in the roundabout dataset. In the roundabout data set, the relation between lanes, including conflicts, is present in the `LANE_RELATIONS` table (schema shown in Table B.3). This table contains information between lane segments that are i) next to each other in the direction of the lawful vehicle travel sequence, ii) in conflict with each other, and iii) adjacent to each other.

B.3.3 Crosswalk dataset lane conflict information

Fig. B.4 shows all the lane segments in the crosswalk dataset. Since pedestrians can travel in any direction, the conflict relations are maintained not only with respect to the lane regions, but also the direction of the pedestrian.

B.4 Relevant agent information

In order to help researchers and practitioners easily construct the games in an interactive traffic scenario, the dataset includes the set of relevant agents from the perspective of each road user. The relevant agents ($R(i)$) from the perspective of a vehicle i for the three datasets are constructed as follows.

Intersection dataset.

A vehicle in conflict with the subject vehicle i is included in $R(i)$ if the vehicle has not yet crossed the conflict point. A leading vehicle of the vehicle in direct conflict is included in $R(i)$

Column Name	Description
ID	Conflict point id
PATH_1	One of the paths that lead to the conflict point. The form of the path is : [ln_<direction_of_entry>_<entry_lane_id>, [ln_<direction_of_exit>_<exit_lane_id>]. E.g. [ln_s_3, ln_n_-2] : Path from South entry lane 3 to North exit lane -2. Lane id are numbered as follows: Entry lanes: Numbered + (positive) from left to right of lane direction. Exit lanes: Numbered - (negative) from left to right of lane direction.
PATH_1_GATES	Lane Gate ids along PATH_1.
PATH_2	One of the paths that lead to the conflict point. Format is same as PATH_1.
PATH_2_GATES	Lane Gate ids along PATH_2.
X_POSITION	UTM lateral co-ordinate (zone 17T) of the conflict point location
Y_POSITION	UTM longitudinal co-ordinate (zone 17T) of the conflict point location
SIGNAL_STATE_PATH_1	The traffic signal states of PATH_1 for the current conflict point to be active from PATH_1's perspective
SIGNAL_STATE_PATH_2	The traffic signal states of PATH_2 for the current conflict point to be active from PATH_1's perspective
POINT_LOCATION	Indicates whether the conflict point is location on the intersection or after (e.g. conflict between North to West turning vehicles using dedicated right turn and South to West turning vehicles.) ON_INTERSECTION : Conflict point is located within the intersection. AFTER_INTERSECTION : Conflict point is located after crossing the intersection.

Table B.2: `CONFLICT_POINTS` table in intersection dataset contain all the conflict points between any two lane segment sequence.

Column Name	Description
LANE_1	Lane segment 1
LANE_2	Lane segment 2
RELATION_TYPE	One of [ADJACENT_SEGMENT, IN_CONFLICT,NEXT_SEGMENT]

Table B.3: `LANE_RELATIONS` table in roundabout dataset contain all relations between two lane segments.

Roundabout dataset.

Relevant vehicles for the roundabout dataset are constructed with respect to vehicles about to enter the roundabout, i.e., in lane segments with suffix `feeder2` in Fig. B.3. Any vehicle that is in the three closest inner and outer circle arms along with the two closes feeder lanes is included in the relevant agent set. Fig. B.5 shows a sketch of these lane segments from the perspective of a subject vehicle on the lane segment `s_to_nw_feeder2` and about to enter the roundabout. This set of lane segments represents the lane segments that a vehicle entering the roundabout should notice for the presence of a vehicle. The closest lane segments are calculated based on the information in the table `LANE_RELATIONS`. The relevant agent information is stored in the table `RELEVANT_AGENTS` in the data set.

Crosswalk dataset

In the crosswalk dataset, from the perspective of a vehicle about to navigate the crosswalk, any pedestrian currently on the crosswalk or about to enter the crosswalk in the next 6 seconds is included in the set of relevant road users. The relevant agent information is stored in the table `RELEVANT_AGENTS` in the data set.

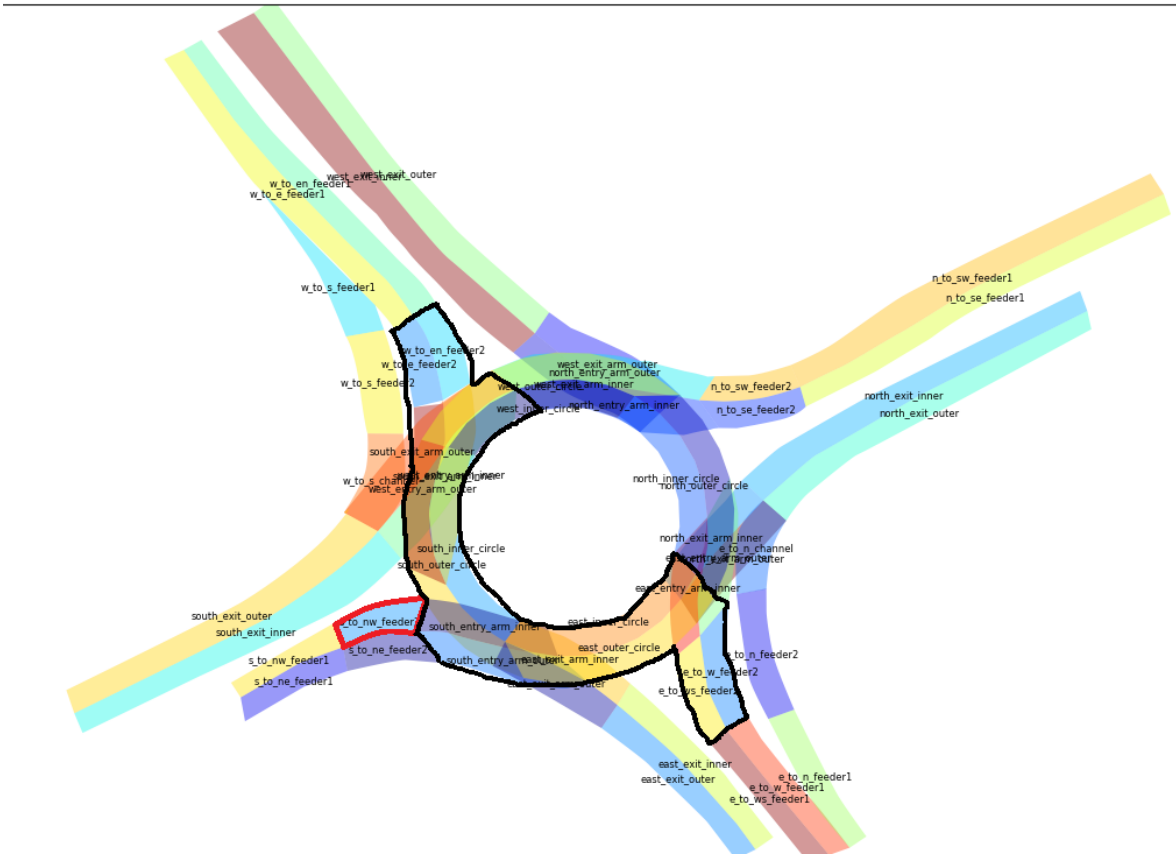


Figure B.5: Sketch of lane segments that are used for in the assignment of relevant vehicles for an example vehicle on lane segment `s_to_nw_feeder2`

.

	Intersection	Roundabout	Crosswalk
Car	3352	6192	193
Pedestrian	264	178	1153
Medium Vehicle	182	174	27
Heavy Vehicle	61	53	4
Bus	22	39	12
Motorcycle	17	0	6
Bicycle	15	3	38
TOTAL	**3913**	**6639**	**1433**

Table B.4: Count of each type of road user in the three datasets.

B.5 Agent statistics

The dataset contain trajectories of seven types of road users, car, pedestrian, medium vehicle, heavy vehicle, bus, motorcycle, and bicycle. Table B.4 shows the distribution of the numbers of each type in the three datasets.

The average speed of vehicles when not stopped was similar for intersection and roundabout datasets, inter-quartile range within 5-10ms^{-1} (Fig. B.6). The maximum speed limit for both locations is 11ms_{-1}. In the intersection dataset, we observe an association between vehicle size and average speed, with larger vehicles having lower average speed. This association is not observed for the roundabout dataset. For the crosswalk data set, the average speed of vehicles, as well as the official speed limit ($5ms^{-1}$) were lower than those of the other two data sets. It is interesting to note that for the crosswalk data set, the average vehicle speeds were higher than the prescribed speed limit for that location.

Fig. B.7 shows the distribution of the minimum distance gap between a subject vehicle and a relevant road user, for each type of relevant road user type. We do not observe a significant difference among the different types of road user in terms of the minimum distance gap; however, the gap values vary between different scenarios. The crosswalk scenario has the lowest interquartile range of 10-25m for most types of road users, whereas intersection and roundabout scenarios have higher gaps, IQR 10-25m and 12-40m, respectively.

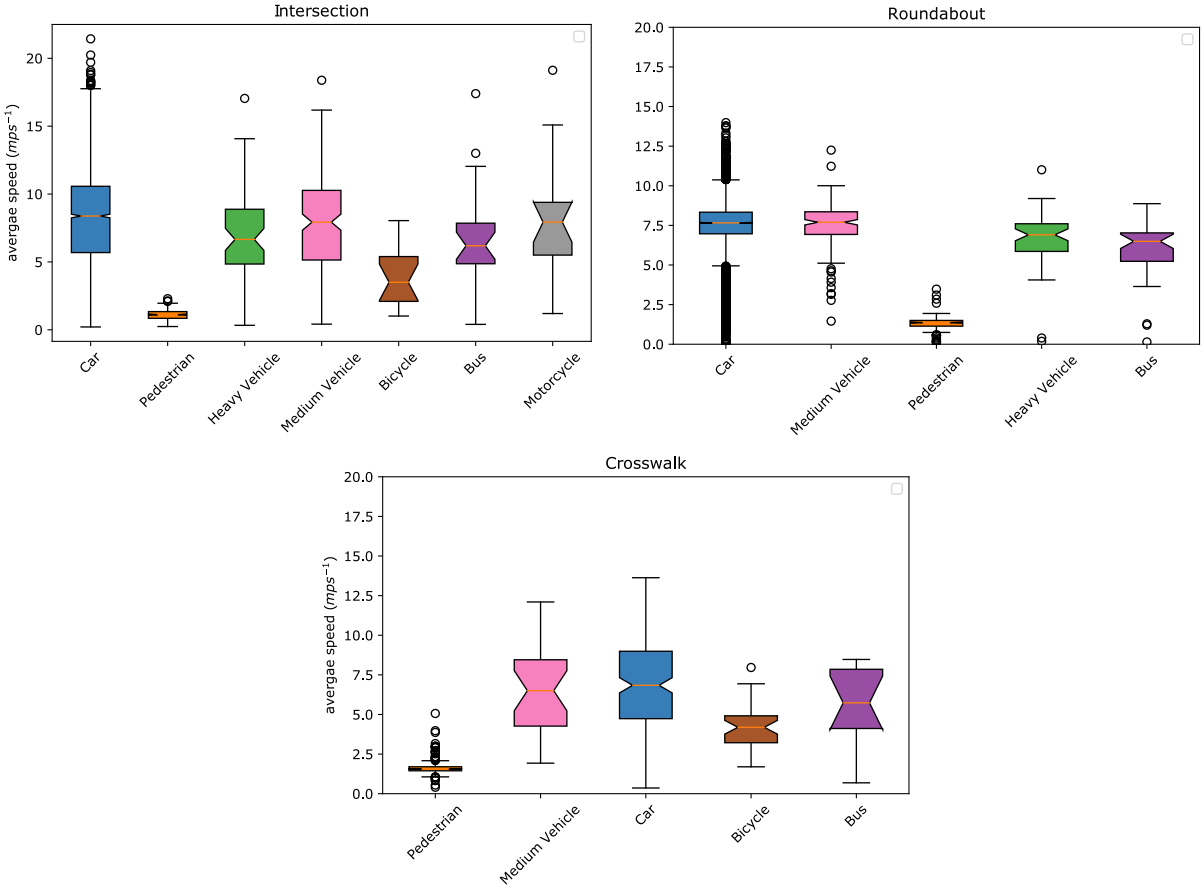


Figure B.6: Distribution of average speed of different types of agents in the datasets

.

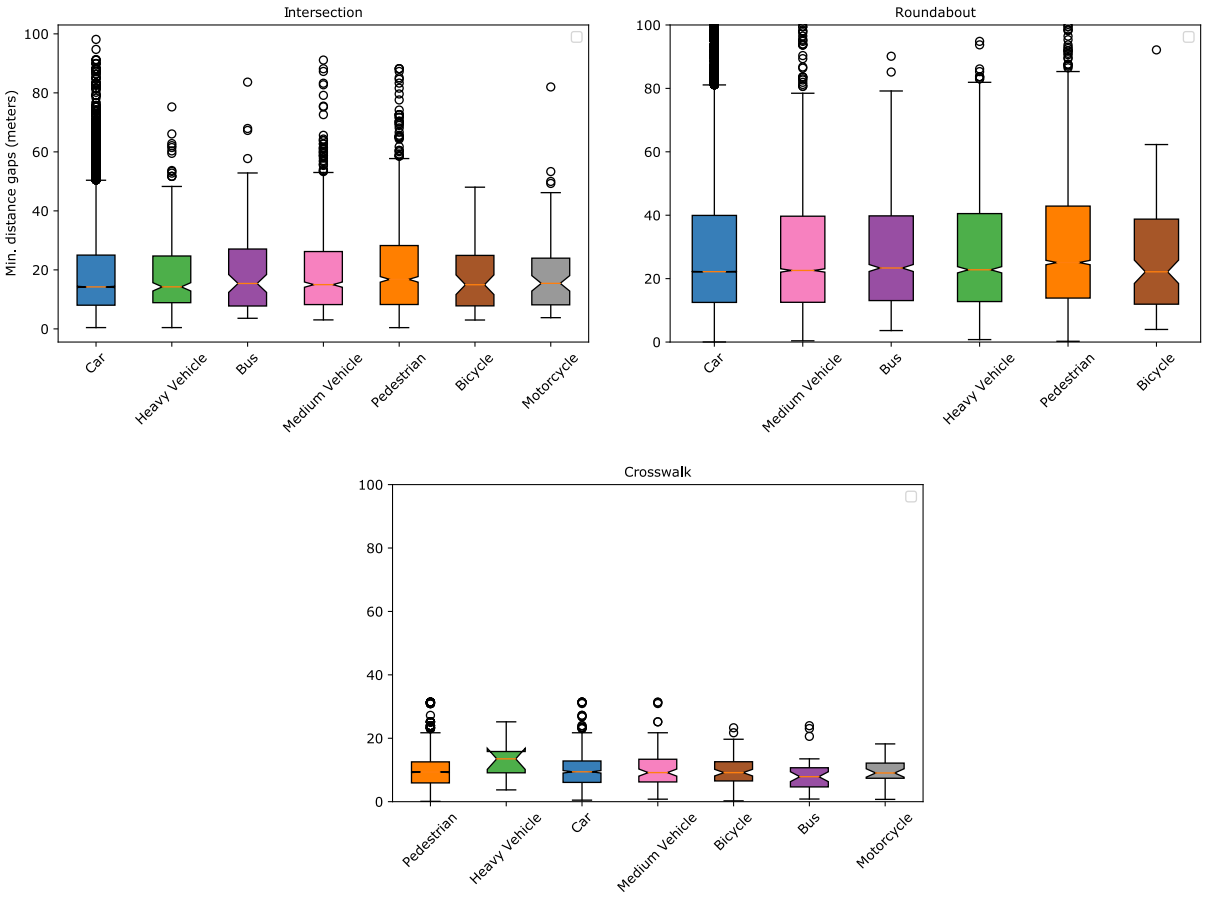


Figure B.7: Distribution of minimum distance gap in each interactive scenario of different types of agents in the datasets

.