Title
Interactive Prediction and Planning for Autonomous Driving: from Algorithms to Fundamental Aspects

Author
Zhan, Wei

Publication Date
2019

Peer reviewed|Thesis/dissertation

Interactive Prediction and Planning for Autonomous Driving: from Algorithms to Fundamental Aspects

by

Wei Zhan

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering - Mechanical Engineering

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Masayoshi Tomizuka, Chair
Professor Francesco Borrelli
Dr. Ching-Yao Chan
Professor Anca Dragan

Fall 2019

Interactive Prediction and Planning for Autonomous Driving: from Algorithms to Fundamental Aspects

Abstract

Interactive Prediction and Planning for Autonomous Driving: from Algorithms to Fundamental Aspects

by

Wei Zhan

Doctor of Philosophy in Engineering - Mechanical Engineering

University of California, Berkeley

Professor Masayoshi Tomizuka, Chair

Inevitably, autonomous vehicles need to interact with other road participants in a variety of highly complex or critical driving scenarios. It is still an extremely challenging task even for the forefront companies or institutes to enable autonomous vehicles to interactively predict the behavior of others, and plan safe and high-quality motions accordingly. The major obstacles are not just originated from prediction and planning algorithms with insufficient performances. Several fundamental problems in the fields of interactive prediction and planning still remain open, such as formulation, representation and evaluation of interactive prediction methods, motion dataset with densely interactive driving behavior, as well as interface of interactive prediction and planning algorithms.

The aforementioned fundamental aspects of interactive prediction and planning are addressed in this dissertation along with various kinds of algorithms. First, generic environmental representation for various scenarios with topological decomposition is constructed, and a corresponding planning algorithm is designed by combining graph search and optimization. Hard constraints in optimization-based planners are also incorporated into the training loss of imitation learning so that the policy net can generate safe and feasible motions in highly constrained scenarios. Unified problem formulation and motion representation are designed for different paradigms of interactive predictors such as planning-based prediction (inverse reinforcement learning), as well as probabilistic graphical models (hidden Markov model) and deep neural networks (mixture density network), which are utilized for the prediction/planning interface design and prediction benchmark. A framework combing decision network and graph-search/optimization/sample-based planner is proposed to achieve a driving strategy which is defensive to potential violations of others, but not overly conservatively to threats of low probabilities. Such driving strategy is achieved via experiments based on the aforementioned interactive prediction and planning algorithms with proper interface designed. These predictors are also evaluated from closed loop perspective considering plan-

ning fatality when using the prediction results instead of pure data approximation metrics. Finally, INTERACTION (INTERnational, Adversarial and Cooperative moTION) dataset with highly interactive driving scenarios and behavior from international locations is constructed with interaction density metric defined to compare different datasets. The dataset has been utilized for various behavior-related research areas such as prediction, planning, imitation learning and behavior modeling, and is inspiring new research fields such as representation learning, interaction extraction and scenario generation.

To my family and friends

Contents

List of Figures

List of Tables

Acknowledgments

It has been an incredible journey for me to pursue my Ph.D. at Berkeley. Without tremendous help and support from many people, it would not be possible for me to finish this dissertation. I want to thank everyone who helped me, although I may not be able to list all of the names here.

I want to express my sincere and deepest gratitude to my advisor, Professor Masayoshi Tomizuka. Without his unreserved and tremendous support, insightful research guidance, as well as the freedom provided, it is not possible for me to have such an academically enjoyable journey to work on fascinating topics for my Ph.D. study. His scientific insights, academic seriousness, enthusiasm of work, as well as optimism of life guided my Ph.D. study and will deeply impact my future career. I also want to thank Mrs. Miwako Tomizuka for her warm care for all of our lab members.

I want to specially thank Dr. Ching-Yao Chan for guiding me into the field of autonomous driving from scratch, and providing me great support for my research projects. He has witnessed all my checkpoints for this journey with his tremendous help. I would like to thank Wei-Bin Zhang who also guided me into the field of autonomous driving and provided me his unconditioned help all the way long. I want to thank Professor Francesco Borrelli, Professor Anca Dragan and Dr. Ching-Yao Chan to serve as my dissertation committee members. Meanwhile, I am thankful for Professor Kameshwar Poolla, Professor Mark Mueller, Professor Anca Dragan and Professor Anil Aswani for serving in my qualifying exam committee.

I am extremely grateful to have the fortune to work with a number of excellent graduate students in Mechanical Systems Control (MSC) Laboratory. I would like to specially thank Liting Sun, Changliu Liu, Jianyu Chen, Jiachen Li, Yeping Hu and Hengbo Ma for their extraordinary research collaborations, who greatly assisted me on the work in this dissertation. I also want to express my gratitude to Cheng Peng, Zining Wang, Chen Tang, Lingfeng Sun and Yiyang Zhou for our great research collaborations during my Ph.D. study. I also would like to thank Kiwoo Shin, Taohan Wang, Zhuo Xu, Saman Fahandezhsaadi, Catherine Weaver and Jinning Li for their cooperations on the research projects. Many thanks to Wenlong Zhang, Junkai Lu, Chung-Yen Lin, Chen-Yu Chan, Minghui Zheng, Te Tang and Hsien-Chung Lin for their great help at the beginning of this journey. I also would like to thank everyone in MSC Lab, including Yizhou Wang, Cong Wang, Raechel Tan, Kevin Haninger, Xiaowen Yu, Yu Zhao, Shiying Zhou, Shuyang Li, Dennis Wai, Yongxiang Fan, Daisuke Kaneishi, Yujiao Cheng, Shiyu Jin, Yu-Chu Huang, Jessica Leu, Ge Zhang, Mona Gao, Changhao Wang, Xinghao Zhu, Ting Xu, Zheng Wu, Wu-Te Yang, and Xiang Zhang for creating such an enjoyable and collaborative environment for research, study and life.

I am also grateful to the visitors of MSC Lab who collaborated with me. Special thanks to Di Wang, Maximilian Naumann and Haojie Shi for their great assists and research collaborations for this dissertation. Many thanks to Weisong Wen, Professor Hiroyuki Okuda and Professor Angel Cuenca for their research collaborations. I also would like to thank Dr. Yi-Ta Chuang from California PATH, as well as Yinghan Jin, Chenran Li, Fan Yang, Zhaoting Li and many others for their collaborations and help.

My Ph.D. study has been greatly benefitting from collaborations with a lot of outstanding researchers. I would like to thank Professor Arnaud de La Fortelle at Mines ParisTech for chairing my research project and providing insightful advices on my research. I want to thank Dr. Yi-Ting Chen at Honda Research Institute, Aubrey Clausse at Mines ParisTech, as well as Julius Kümmerle, Hendrik Königshof and Professor Christoph Stiller at Karlsruhe Institute of Technology for their research collaborations for this dissertation.

I want to acknowledge Chair Drive for All - Mines ParisTech Foundation and Berkeley DeepDrive for supporting the work in this dissertation.

Last but not least, my deepest love goes to my parents, as well as my grandparents. I could never go as far without their unyielding support and encouragement.

Chapter 1

Introduction

1.1 Background and Challenges

Tremendous amount of research and implementation efforts has been devoted to autonomous vehicles in the past decade. Successful demos or applications can be found in specific scenarios. However, it is still an extremely challenging task for autonomous vehicles to navigate themselves in dense urban scenarios with highly interactive behavior, even for leading companies or research institutes in this field. In such scenarios, autonomous vehicles need to accurately predict potential behavior of other road participants for each possible future motion of themselves, and plan safe and high-quality motions accordingly. Such interactive prediction and planning problem is one of the major blocks to enable fully autonomy. In order to decompose the key components and aspects to support interactive *prediction and planning algorithms*, we provide a block diagram in Fig. 1.1 (a) to illustrate what we need to explore.

The lower part in Fig. 1.1 (a) corresponds to prediction. We need a *motion dataset* with highly interactive scenarios and behavior, based on which the prediction algorithms can be trained or designed. Then appropriate simplification of *environmental and motion representation* are required to construct the input and output of the prediction algorithms, as well as the ground truth from the motion dataset. Also, the *problem formulation* of the prediction algorithms depends on which entities' motions are contained in the input and output. Then appropriate metrics are required to compare the ground truth and predicted distribution and provide *predictor evaluation*. Moreover, we need to incorporate various kinds of *prior knowledge* such as high-definition (HD) map, traffic rules and vehicle models, to facilitate representation, evaluation and algorithm design.

The upper part in Fig. 1.1 (a) corresponds to planning. Similar to the prediction counterparts, we need *environmental representation* to construct the safety constraints and cost of the planning algorithms, as well as metrics for *planner evaluation*. More importantly, appropriate *interface for interactive prediction and planning* is needed. The predictor should be aware of potential planned motions, while the planner needs to obtain the level of threat

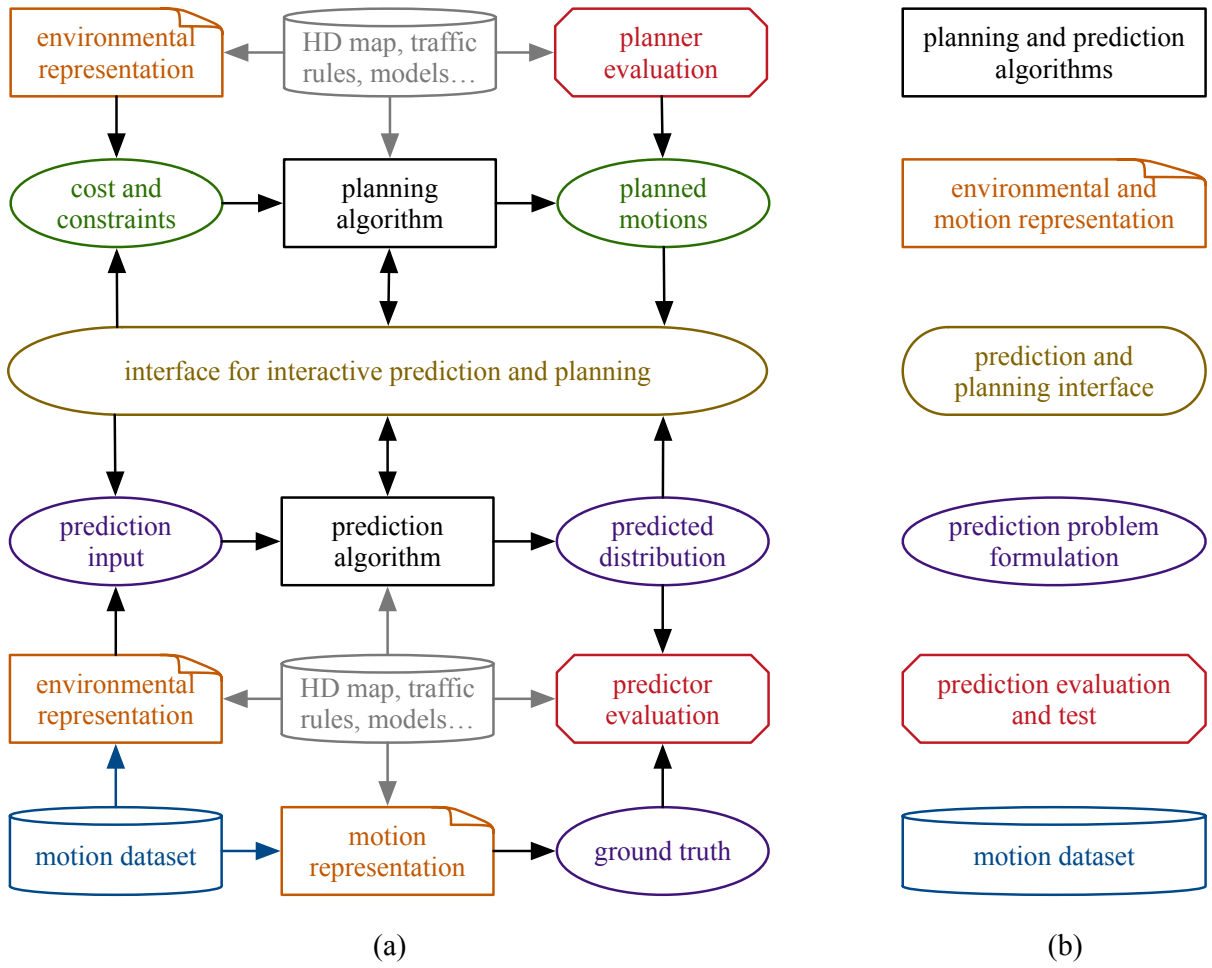


Figure 1.1: Fundamental aspects for interactive prediction and planning.

and probabilities of potential future cases.

Comparing to lots of prediction and planning algorithm proposed, the aforementioned fundamental aspects shown in Fig. 1.1 (b) are under-explored by researchers. In this dissertation, therefore, we would like to provide our answers to the following fundamental questions.

- What kind of motion data is required for interactive prediction and planning? How do we construct a highly interactive *motion dataset*? How do we measure interaction density in a motion dataset?
- What is the *problem formulation* to construct a unified framework for different paradigms of interactive prediction algorithms, such as planning-based prediction, as well as pre-

dictors based on probabilistic graphical models and neural networks?

- What are the appropriate *motion representation* for the ground truth of predictor evaluation? Is there a generic *environmental representation* for different kinds of scenarios by decomposing the topological elements of interaction?
- What are appropriate metrics for *predictor evaluation*? Is there any difference when evaluating predictors from data approximation perspective and potential fatality perspective?
- Is there an interface for interactive prediction and planning, which can incorporate different paradigms of predictors, as well as decision-maker and planner under uncertainty? If so, how to design the interface and the framework for decision and planning?

1.2 Dissertation Structure and Contributions

In order to answer the questions listed above, we need several building blocks, including deterministic planning, as well as decision and planning under uncertainty. Then in-depth discussions on the aforementioned fundamental problems will be provided. We will briefly introduce the dissertation structure in this section, see Fig. 1.2.

Part I Deterministic Planning

Deterministic motion planning is the most basic but essential part. It generates motions for an autonomous vehicle given deterministic future trajectories of other road participants. Regardless of the complexity of the prediction and planning problems, it is always necessary to have a planner which can generate safe and high-quality motions in various scenarios with relatively long horizon in real time. In this part, we will answer the following two questions.

- What is a generic environmental representation for various scenarios and how to utilize the representation to construct a planner with fast computation, long horizon and guarantees on safety and feasibility?
- How do we incorporate hard constraints in an optimization-based planner into the training of imitation learning, so that the trained policy can generate safe and feasible motions in highly constrained scenarios?

Chapter 2

Conventional layered planning architecture temporally partitions the spatiotemporal motion planning by the path and speed, which is not suitable for lane change and overtaking scenarios with moving obstacles. In this chapter, we propose to spatially partition the motion planning by longitudinal and lateral motions along the rough reference path in the Frenét

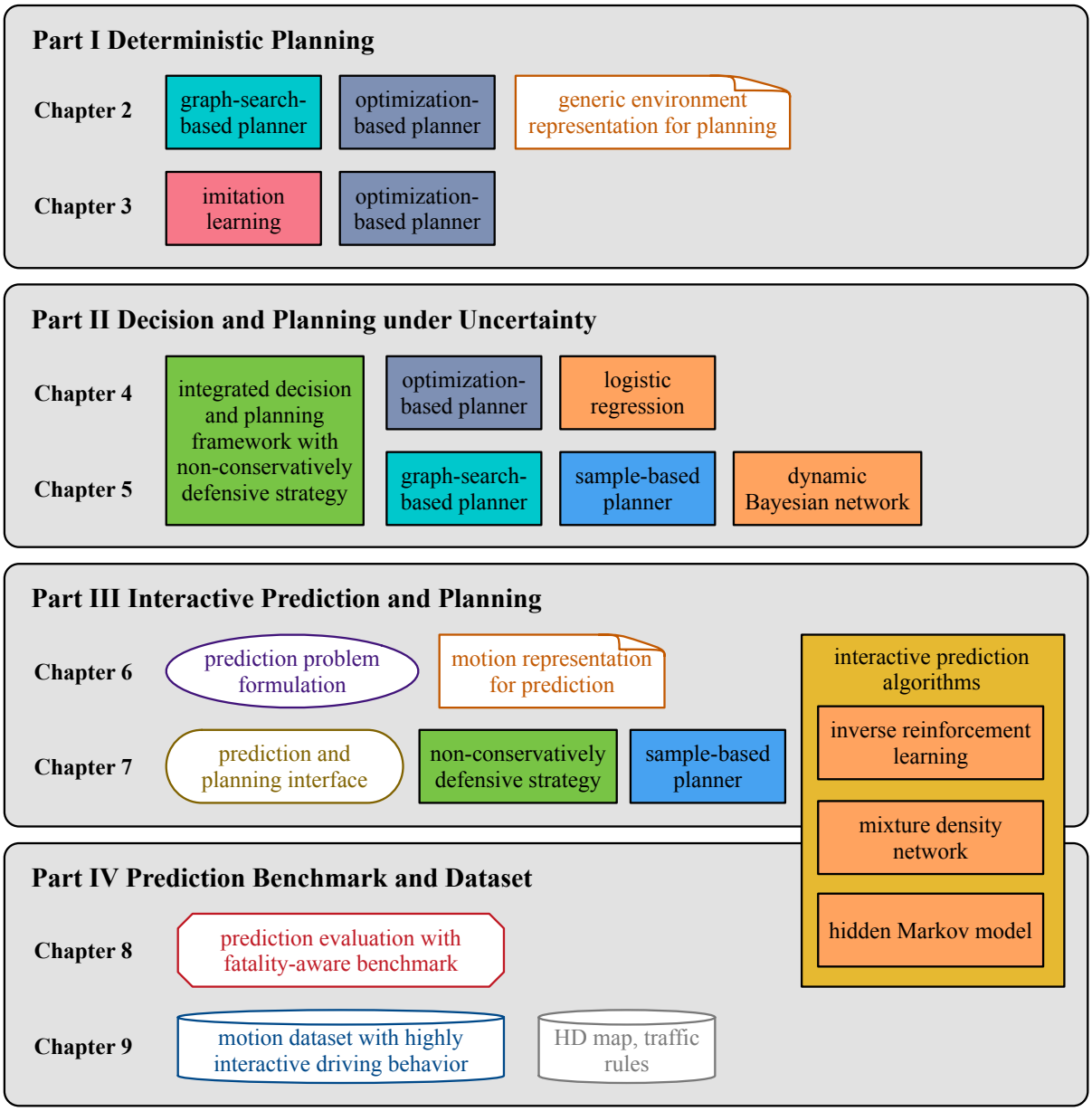


Figure 1.2: Structure of the dissertation.

Frame, which makes it possible to create linearized safety constraints for each layer in a variety of on-road driving scenarios. A generic environmental representation methodology is proposed with three topological elements and corresponding longitudinal constraints to compose all driving scenarios mentioned in this chapter according to the overlap between the potential path of the autonomous vehicle and predicted path of other road users. Planners

combining A* search and quadratic programming (QP) are designed to plan both rough long-term longitudinal motions and short-term trajectories to exploit the advantages of both search-based and optimization-based methods. Limits of vehicle kinematics and dynamics are considered in the planners to handle extreme cases. Simulation results show that the proposed framework can plan collision-free motions with high driving quality under complicated scenarios and emergency situations.

Chapter 3

Imitation learning has great potential to learn sophisticated driving policy under complicated interaction between human drivers. However, it is hard for policy networks to satisfy safety and feasibility constraints, which is not a challenging task for conventional motion generation methods, such as optimization-based approach. In this chapter, we propose Constrained Policy Net (CPN), which can learn safe and feasible driving policy from arbitrary inequality-constrained optimization-based expert planners. Instead of supervised learning with L_2 norm as the loss, we incorporate the domain knowledge of the expert planner directly into the training loss of the policy net by applying barrier functions to the safety and feasibility constraints of the optimization problem. An exemplar scenario with obstacles on both sides is used to implement the proposed CPN. Test results demonstrate that the policy net can learn to generate motions near boundaries of safety and feasibility constraints to achieve high driving quality as the baseline optimization while the constraints are satisfied.

Part II Decision and Planning under Uncertainty

With satisfactory deterministic planners in Part I, we go one step further to design decision making and motion planning framework and algorithms under uncertainty from perception and prediction modules. Practically, the information we obtain perception and prediction modules are probabilistic with uncertainties. We need to design appropriate decision and planning framework to handle such uncertainties. We will answer the following question in this part.

- How do we design a decision and planning framework to achieve a driving strategy, which is defensive to potential violations of ours, but not overcautious to fake threats with low probability?

Chapter 4

From the driving strategy point of view, a major challenge for autonomous vehicles in urban environment is to behave defensively to potential dangers, yet to not overreact to threats with low probability. As it is overwhelming to program the action rules case-by-case, a unified planning framework under uncertainty is proposed in this chapter, which achieves a non-conservatively defensive strategy (NCDS) in various kinds of scenarios for urban autonomous driving. First, uncertainties in urban scenarios are simplified to two

probabilistic cases, namely passing and yielding. Two-way-stop intersection is used as an exemplar scenario to illustrate the derivation of probabilities for different intentions of others via a logistic regression model. Then a deterministic planner is designed as the baseline. Also, a safe set is defined, which considers both current and preview safety. The planning framework under uncertainty is then proposed, in which safety is guaranteed and overcautious behavior is prevented. Finally, the proposed planning framework is tested by simulation in the exemplar scenario, which demonstrates that an NCDS can be realistically achieved by employing the proposed framework.

Chapter 5

In this chapter, we first illustrate the essence of the proposed non-conservatively defensive strategy (NCDS), which is a combination of decision network and arbitrary motion planner with cost and constraints. Then we provide the geometrical intuition of the proposed NCDS via showing how the cost maps are changed by the probabilities in the space of augmented decision variables. Then the NCDS is extended to incorporate graph-search-based planner and sample-based planner so that real-time computation can be achieved. We also utilize a dynamic Bayesian network (DBN) to provide predictions and corresponding probabilities. Experiments on real vehicles with both of the planners are conducted with driving scenarios and vehicle motions from real world, showing the effectiveness of the proposed strategy in the integrated decision and planning framework.

Part III Interactive Prediction and Planning

Based on the efforts on deterministic planning in Part I, as well as decision and planning under uncertainty in Part II, we dive into interactive prediction and planning in this part. The core of this part is to answer "what if the planner takes a specific action" in the prediction module, and enable the planner to generate desirable motions accordingly in an interactive way. The following problems are addressed in this part.

- What are the problem formulations for prediction algorithms, and how to unify the formulation for different paradigms of interactive prediction algorithms, such as planning-based prediction, as well as prediction based on probabilistic graphical models and neural networks?
- What are the simplified motion representations for prediction, and which of them can be considered as the ground truth?
- How do we design the interface to incorporate all the aforementioned paradigms of interactive prediction algorithms, as well as interactive decision and planning framework based on the NCDS we proposed?

Chapter 6

Accurate interactive, probabilistic prediction for intention and motion of road users is a key prerequisite to achieve safe and high-quality decision-making and motion planning for autonomous driving. The evaluation of probabilistic prediction and its utilization for planning highly rely on the problem formulation variation and motion representation simplification, both of which lack a formal foundation in a comprehensive framework. To address such concerns, we provide a survey, and address the omitted but crucial problems of two under-explored aspects of probabilistic prediction: problem formulation and representation simplification. We also discuss the reasons why "intention" is not suitable to serve as a motion indicator in highly interactive scenarios. Then we choose the representation and provide the transformation between reaction prediction and situation prediction to construct a unified framework for different paradigms of interactive prediction methods in Chapter 7.

Chapter 7

Reactive predictions are necessary in highly interactive driving scenarios to answer "what if I take this action in the future" for autonomous vehicles. Many recently proposed methods based on probabilistic graphical models (PGM), neural networks (NN) and inverse reinforcement learning (IRL) have great potential to solve the problem. In this Chapter, the aforementioned three paradigms of interactive prediction algorithms are briefly introduced with the unified formulation and representation proposed in Chapter 8. Then we propose an interface of interactive prediction and planning. With such interface, we can incorporate all the three paradigms of interactive prediction algorithms, as well as an integrated decision and planning framework with NCDS and a sample-based planner. Results in highly interactive driving scenarios show the effectiveness of the proposed framework and corresponding methodologies.

Part IV Prediction Benchmark and Dataset

With the unified formulation and representation of interactive prediction in Part III, we dive into evaluation of prediction algorithms in this part. Prediction evaluation should take into account not only the data approximation performance, but also the consequences of the decision and planning when the prediction results are adopted, as inspired by Part III. Also, a motion dataset with highly interactive scenarios and behavior is one of the most essential assets for all behavior-related research areas, such as prediction, planning, imitation learning, behavior modeling and representation learning. In this part, we will answer the following questions.

- What are evaluation metrics for probabilistic prediction, and how do we evaluate different paradigms of interactive prediction algorithms considering both data approximation and fatality when adopting the prediction results?

- What kind of motion data is required for various behavior-related research areas? How do we construct the dataset and how do we measure the interaction density of the dataset?

Chapter 8

How to evaluate probabilistic prediction is always one of the most significant problems in behavior-related research areas. In this chapter, we first provide a survey of the existing metrics considering not only the data approximation but also prior knowledge and decision-related perspective. Then we utilize the unified framework proposed in Chapter 6 to homogenize the problem formulation and representation simplification of the three paradigms of interactive prediction methods in Chapter 7 in order to provide benchmark for these methods. We propose to use Brier score as the baseline metric for data approximation evaluation. In order to reveal the fatality of the consequences when the predictions are adopted by decision-making and planning, we propose a fatality-aware metric, which is a weighted Brier score based on the criticality of the trajectory pairs of the interacting entities. Conservatism and non-defensiveness are defined from the weighted Brier score to indicate the consequences caused by inaccurate predictions. Modified methods based on PGM, NN and IRL are provided to generate probabilistic reaction predictions in an exemplar scenario of nudging from a highway ramp. The results are evaluated by the baseline and proposed metrics to construct a mini benchmark. Analysis on the properties of each method is also provided by comparing the baseline and proposed metric scores.

Chapter 9

Interactive motion datasets of road participants are vital to the research and development of autonomous vehicles in both industry and academia. Research areas such as motion prediction, motion planning, representation learning, imitation learning, behavior modeling, behavior generation, and algorithm testing, require support from high-quality motion datasets containing interactive driving scenarios with different driving cultures. In this chapter, we present an INTERnational, Adversarial and Cooperative moTION dataset (INTERACTION dataset) in interactive driving scenarios with semantic maps. Five features of the dataset are highlighted. 1) The interactive driving scenarios are diverse, including urban/highway/ramp merging and lane changes, roundabouts with yield/stop signs, signalized intersections, intersections with one/two/all-way stops, and so on. 2) Motion data from different countries and different continents are collected so that driving preferences and styles in different cultures are naturally included. 3) The driving behavior is highly interactive and complex with adversarial and cooperative motions of various traffic participants. Highly complex behavior such as negotiations, aggressive/irrational decisions and traffic rule violations are densely contained in the dataset, while regular behavior can also be found from cautious car-following, stop, left/right/U-turn to rational lane-change and cycling and pedestrian crossing, and so on. 4) The levels of criticality span wide topics, from regular safe operations to danger-

ous, near-collision maneuvers. Real collision, although relatively slight, is also included. 5) Maps with complete semantic information are provided with physical layers, reference lines, lanelet connections and traffic rules. The data is recorded from drones and traffic cameras, and the processing pipelines for both are briefly described. Statistics of the dataset in terms of number of entities and interaction density are also provided, along with some utilization examples in the areas of motion prediction, imitation learning, decision-making and planning, representation learning, interaction extraction and social behavior generation. The dataset can be downloaded via `https://interaction-dataset.com`.

Part I

Deterministic Planning

Chapter 2

Generic Representation and Planner

2.1 Introduction

Motion planning is one of the most challenging areas in autonomous driving, which should comprehensively consider the following factors:

1. driving qualities such as comfort and time-efficiency;
2. soft traffic rules such as speed limit and lane keeping;
3. hard traffic rules such stop sign and traffic signal;
4. avoiding collisions with all obstacles;
5. vehicle kinematics feasibility such as path curvature;
6. vehicle dynamics feasibility such as tire friction circle and engine traction limits.

For on-road driving in urban areas or on highway, the planner need to optimize 1) and 2), and guarantee 3) to 6) in a variety of driving scenarios with different types of roadway geometries and various kinds of road participants as moving obstacles.

There has been a considerable amount of research efforts devoted to motion planning of autonomous vehicles [90][37][57]. Some of the work were focused on designing a planner in a spatiotemporal framework. For example, a state-lattice-based spatiotemporal representation and search algorithm was proposed in [83] for motion planning of highway driving. A trajectory planner based on a single spatiotemporal optimization was proposed in [145]. The approach was applicable to various kinds of scenarios, but it still needed an ad hoc process for specific situations. The initial value of the numerical optimization significantly affected the final result due to the existence of local optima. Also, the runtime was not desirable for time-critical replanning.

It is computationally intractable to include all factors in 1)-6) precisely in a single spatiotemporal framework to handle various kinds of scenarios. For search-based methods,

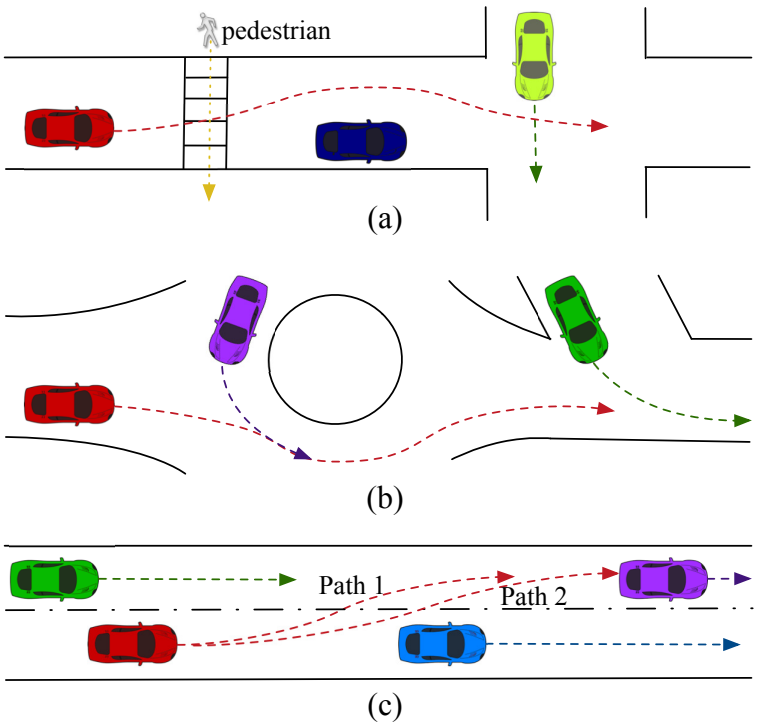


Figure 2.1: Three typical categories of driving scenarios.

wasteful explorations exist in the search space of trajectories. For optimization-based approaches, the complicated expressions of feasibility and safety constraints result in undesirable runtime. Therefore, a layered planning architecture temporally partitions the spatiotemporal planning by the path and speed profile planning, which significantly reduces computational complexity.

An elastic-band-based path planner and a speed-constraint-based temporal planner were proposed in [43] to generate trajectories in complicated scenarios for urban driving. Multiple on-road driving scenarios were handled in [97] by interpolating piecewise-linear paths with quintic Bézier curve, and designing a velocity planner based on model predictive control. A driving strategy planning method via A* search was proposed in [52] by providing a long-term rough speed profile on a predefined lane. The predefined lane acted as the planned path and [52] essentially followed the conventional temporally-partitioned architecture.

Three typical categories of driving scenarios are shown in Fig. 2.1. The path of the host vehicle (red car) does not depend on the speed profile in: (a) scenarios with static obstacles (blue car) and moving obstacles crossing the path (pedestrian, green car), and (b) scenarios with merging obstacles (purple car in roundabout and green car at T-intersection). The path depends on the speed profile for (c) lane change or overtaking scenarios with moving obstacles (all other cars).

Temporally-partitioned architecture is suitable for scenarios in Fig. 2.1 (a) and (b), in which path and speed profile can be well decoupled. However, for scenarios in Fig. 2.1 (c), the temporally-partitioned architecture is not applicable. Normally, scenarios in Fig. 2.1 (a), (b) and (c) may appear simultaneously in one planning horizon for on-road driving, thus our primary motivation is as follows:

Motivation 1: propose a novel planning architecture instead of the temporally-partitioned one, to handle on-road driving scenarios which include, but are not limited to scenarios in Fig. 2.1.

The combinatorial aspect of planning was considered in [11] based on the planner in [145]. Difficulties still existed in generating generic constraints. A behavioral planner was proposed in [125] with spatiotemporal candidate strategies and reaction prediction of surroundings. Adaptive cruise control and lateral control were required to achieve the desirable strategy. Safety constraints was only in the behavioral strategy level, and the final trajectory generated by the controllers may be unsafe in highly constrained environments. Thus our second motivation is:

Motivation 2: create generic environmental representation for each planning layer to generate safety constraints so that the final trajectory is guaranteed to be collision-free.

Besides the normal scenarios in Fig. 2.1, extreme scenarios may exist to push the autonomous vehicles to the boundaries of the vehicle kinematics and dynamics. A planner was proposed in [31] for highly constrained environments such as narrow roads with sharp turns. Planners for emergency maneuvers were designed in [58] and [106]. These planners targeted at desirable performance for specific extreme scenarios, but lack the portability for normal on-road driving scenarios. Extreme scenarios may appear simultaneously with normal ones, thus our third motivation is:

Motivation 3: include factors in 5) and 6) in the planning framework to deal with situations pushing the vehicle to the kinematics boundary such as U-turn, or dynamics boundary such as emergency maneuvers.

To the best of our knowledge, this work is the first to propose spatially-partitioned environmental representation and planning architecture, which are generic for all types of aforementioned on-road driving scenarios (see in Fig. 2.1). The planning framework comprehensively takes into account factors 1) to 6), which can also deal with extreme cases with guaranteed safety and feasibility.

The remainder of the chapter is organized as follows. Section 2.2 provides the architecture of the proposed planning framework and Section 6.4 discusses generic representation of the environment for a variety of scenarios; the generic planning framework based on the proposed environmental representation is discussed in Section 2.4; simulation results are provided in Section 2.5; and Section 9.7 concludes the chapter.

2.2 Overview of the Architecture

The motion planning module is presented in Fig. 2.2 from the viewpoint of the overall system architecture. The motion planning module receives the traffic-free reference path and desirable speed from the routing and reference planning module, as well as the driving task from the task planning module. At the final stage, the motion planning module outputs a trajectory satisfying factors 1)-6) described in Section 9.1 for the control module to execute. The perception and localization module provides the necessary inputs to all three stages of planning and control.

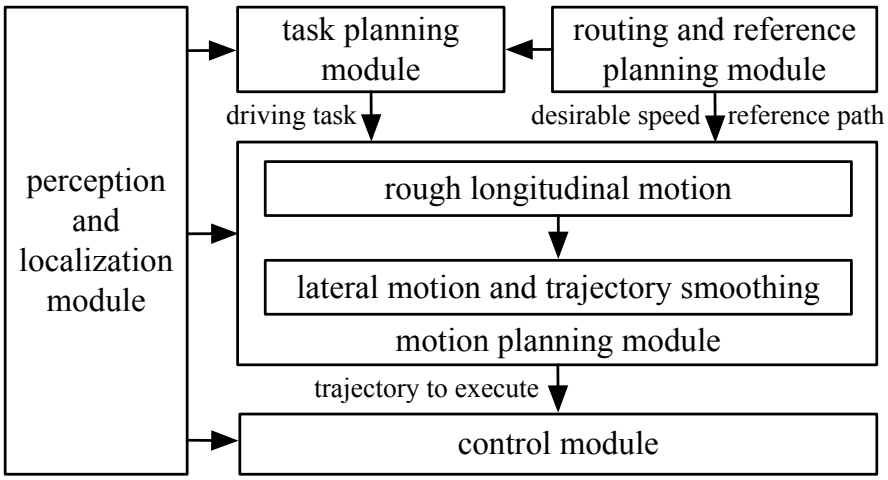


Figure 2.2: Overview of the architecture

The processing of the routing and reference planning module can be completed partially offline, and will be further discussed in Section 2.3. The task planning module determines the combinatorial aspect of planning in long term, such as following or overtaking a bicyclist, changing or keeping the lane, etc. Receiving a driving task does not mean that the final decision has been made. The task planning module may compare the costs of potential choices and select the best choice to execute, or plan ambiguous next-step motions with several potential decisions according to the intention probabilities of other road participants [134].

This chapter is focused on the design of the motion planning module. Instead of adopting the temporally-partitioned architecture, we propose a novel spatially-partitioned planning architecture since the information associated with the temporal dimension is extremely important for any potential layer to deal with moving obstacles. Due to the nonholonomic motion constraints of the vehicle, longitudinal motion plays a more significant role than the lateral motion given the reference. Longitudinal constraints are often associated with higher priority, such as traffic signal, stop sign, car following, speed limit, turning speed on curvy road, yielding those who hold the right-of-way, etc. Also, when lane change and overtaking

scenarios with moving obstacles (shown in Fig. 2.1 (c)) are involved, longitudinal motion is more influential in deciding the trajectory.

Therefore, we propose to spatially partition the spatiotemporal planning by the longitudinal and lateral motions along the reference. Such architecture also makes it possible to linearize the safety constraints for each layer. We suggest that the planning framework should first plan long-term (at least 10 s) rough longitudinal motions according to the driving task and longitudinal constraints on safety and feasibility. Given the rough longitudinal motions, the rough short-term lateral motions are determined and smooth trajectories are further planned by considering both lateral and longitudinal constraints on safety and feasibility.

2.3 Generic Environmental Representation

In order to design a generic motion planning framework, we need a simplified and general representation of the environment to represent the longitudinal constraints for the long-term longitudinal motion planning. The methodology of generic environmental representation is proposed in this section, which is applicable for all types of scenarios mentioned in Section 9.1. .

Representation in Frenét Frame

The lane centerline is a baseline as the traffic-free reference path (refer to [42] for comfortable and human-like improvement). A Frenét Frame can be defined on the reference path (shown in Fig. 2.3), in which the motion in Cartesian coordinates $(x(t), y(t))$ can be represented with the longitudinal position along the path $s(t)$, and lateral deviation to the path $d(t)$.

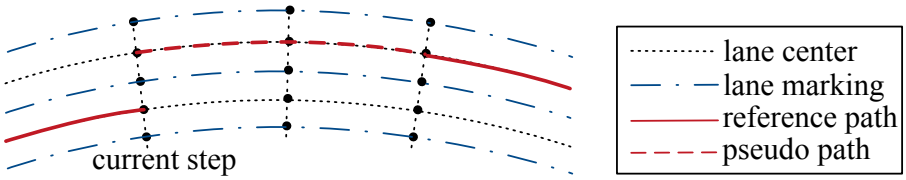


Figure 2.3: Representation within Frenét Frame

Note that when changing the lanes, the reference path is switched. In order to keep the scale consistency of $s(t)$, a pseudo path is created on the target lane at the current time step for the horizon if a lane-change task is received (as shown in Fig. 2.3). The real reference path starts at the point where the lane change should be completed, and the decision on when to start and complete the lane change is left to the motion planning module.

Suppose $v_{\text{limit}}(s)$ is the speed limit introduced by traffic law, and $\kappa(s)$ is the curvature of the reference path in cartesian coordinate. The desirable speed v^{d} can be written as $v^{\text{d}}(s) = \min\left\{v_{\text{limit}}(s), \sqrt{a_{\text{lat}}^{\text{des}}/\kappa(s)}\right\}$, where $a_{\text{lat}}^{\text{des}}$ is the desirable lateral acceleration.

Boundaries and indicators

To guarantee safety, an autonomous vehicle needs to create the longitudinal boundaries at each time step for longitudinal motion planning. There are three categories of the boundaries according to the topological overlap of the reference path and the prediction of the obstacles.

Suppose $s_i(t)$ is the predicted longitudinal position (along the reference path of the host vehicle) of the i-th obstacle (can also be a stop bar for stop sign or traffic signal) which may affect the trajectory of the host vehicle. $v(t)$ and $v_i(t)$ are the speeds of the host vehicle and the i-th obstacle along the reference path, respectively. Then the safety factors can be represented by the following boundaries, as well as activeness indicators.

An obstacle can generate two kinds of boundaries. One is *front bound* blocking the host vehicle, which is written as $f_i(s(t), v(t), s_i(t), v_i(t)) \leq 0$; The other is *rear bound* forcing the host vehicle to pass, which is written as $g_i(s(t), v(t), s_i(t), v_i(t)) \leq 0$. More explanations on the boundaries can be found in Section 2.3.

The change of the lateral positions of the host vehicle and obstacles can make the aforementioned longitudinal boundaries ineffective. Also, when lane change or overtaking is completed, the original target vehicle will be behind the host vehicle and the *rear bound* is no longer considered. Moreover, after a full stop near the stop sign, the boundary created by the stop bar can be discarded if the intersection is clear. Therefore, an indicator function is needed to indicate the activeness of the boundary. The boundaries of each obstacle have corresponding indicator functions $I_{fi}^{\text{act}}(t)$ and $I_{gi}^{\text{act}}(t)$, indicating whether the boundary is *active*, *potentially active* or *inactive* at each time step.

The *active* range of the boundary f_i or g_i can be written as $\left[t_{fi}^{\text{s}}, t_{fi}^{\text{e}}\right]$ or $\left[t_{gi}^{\text{s}}, t_{gi}^{\text{e}}\right]$ with start and ending time. If an obstacle only creates one boundary, it can be written as $[t_i^{\text{s}}, t_i^{\text{e}}]$.

Topological elements

Given the reference path and motion prediction of the obstacles (including traffic signal phase, stop sign, static and moving obstacles), the topological relationship between the potential paths of the host vehicle and the predicted path of the obstacle can be decomposed to three basic topological elements (shown in Fig. 2.4), in which red solid lines represent the potential path of the host autonomous vehicle and blue dashed lines corresponds to predicted path of the obstacles.

(a) Point-overlap corresponds to scenarios with crossing traffic (see scenarios in Fig. 2.1 (a) as examples), traffic signal or stop sign. When the prediction of the i-th obstacle crosses

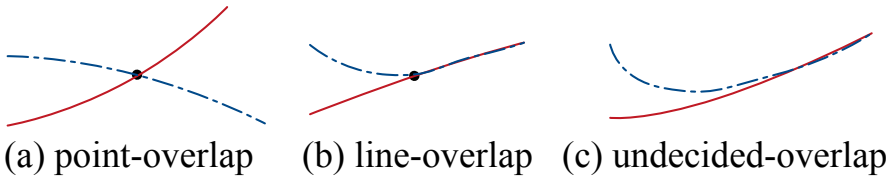


Figure 2.4: Topological elements to represent the relationship of the potential path of the host autonomous vehicle and predicted path of the obstacles.

the reference path, it forms a *front bound* or a *rear bound*, that is,

$$f_i(t) = s(t) + \Delta s - s_i(t) \leq 0, \tag{2.1}$$
$$g_i(t) = s_i(t) + \Delta s - s(t) \leq 0, \tag{2.2}$$

where Δs is a safe margin containing the geometry of the obstacle. The *active* range of the boundary can be obtained from prediction of crossing traffic occupying the reference path, or the red-light phase of traffic signal. For motions at a stop sign, the boundary is not *inactive* until a full stop $v = 0$ appears in a position range before the stop bar.

(b) Line-overlap corresponds to scenarios of merging and car following (see scenarios in Fig. 2.1 (b) as examples), including mergings at roundabout, highway ramp, intersections, etc. These scenarios have fixed merging or demerging point, which forms a *line-overlap* between the prediction and reference path.

If the obstacle is the front obstacle to follow, then it is a *front bound*

$$f_i(t) = s(t) + \frac{v(t)^2}{2\left|a_{\text{long}}^{\min}\right|} + \Delta s - s_i(t) - \frac{v_i(t)^2}{2\left|a_i^{\min}\right|} \leq 0, \tag{2.3}$$

where $a_{\text{long}}^{\min}$ and $a_i^{\min}$ are the maximum deceleration of the host vehicle and the i-th obstacle, respectively. The *active* range starts when the obstacle reaches the fixed merging point, and ends when it reaches the fixed demerging point.

If the obstacle follows the host vehicle, then it is a *rear bound*

$$g_i(t) = s_i(t) + \frac{v_i(t)^2}{2\left|a_i^{\min}\right|} + \Delta s - s(t) - \frac{v(t)^2}{2\left|a_{\text{long}}^{\min}\right|} \leq 0, \tag{2.4}$$

The *active* range starts when the host vehicle reaches the fixed merging point, and ends after a specific time range when the obstacle is regarded as a vehicle to follow the host.

(c) Undecided-overlap corresponds to lane change and overtaking scenarios with moving obstacles (see scenarios in Fig. 2.1 (c) as examples). These scenarios do not have fixed merging or demerging point since the desirable path depends on the speed profile. It is undecided where the overlap of the prediction and the reference path starts or end.

For obstacles to follow on the original lane or the target lane, (2.3) can be used as a *potentially active* boundary. When the host vehicle leaves the original lane and reaches the target lane, the boundary corresponds to the obstacle to follow on the original lane is *inactive* and that of the target lane is *active*.

When the host vehicle plans to cut in front of an obstacle, (2.4) can be used as a *potentially active* boundary. When the host vehicle reaches the target lane, the boundary becomes *active*. Then the boundary is *inactive* when the obstacle follows the host.

For an oncoming vehicle on the lane used temporarily by the host vehicle to accomplish the overtaking maneuver, (2.1) can be used as a *potentially active* boundary. When the host vehicle reaches the lane of the oncoming vehicle, the boundary is *active*. When the host vehicle completely leaves the lane of the oncoming vehicle, the boundary is *inactive*.

2.4 Motion Planning Framework

In this section, planners for long-term longitudinal motion, short-term lateral motion and trajectory smoothing are designed. Optimization-based methods perform well in providing motions with high driving quality, but they are inefficient in dealing with nonlinear and nonconvex terms. Search-based approaches are efficient in handling complex terms, but hard to provide smooth motions. In this section, search-based and optimization-based methods are combined to find desirable motions for the long-term rough speed and short-term trajectory. We first use A* search to find a rough reference, then formulate a quadratic programming (QP) optimization problem which can be solved extremely fast with global minimum.

Long-term longitudinal motions

The generic representation described in Section 6.4 is used to plan long-term rough speed via A* search, as is inspired by [52]. The state of this problem is defined as $z = [s, v, t, I^{\text{act}}]^T$, where t is the time stamp which will increase by Δt_l once an action is applied, and I^{act} is a vector containing all relevant activeness indicators within the horizon.

The action is discretized as a set of accelerations $a = [-2, -1, 0, 1]$ m/s^2. Such discretization was used in [52] which well covered normal actions. When no sequence of action can be found within the action set to obtain a collision-free speed profile, the planner will calculate the minimum constant acceleration or deceleration to avoid collision, which corresponds to an emergency situation. s and v are updated by applying a constant acceleration in a period of sampling time Δt_l. Since actions are discretized and the sampling time is fixed, there will be lots of repeated states. Thus graph search is applied instead of tree search to prevent revisiting visited states.

The cost function of A* search basically includes penalty for accelerations and deviation

from the desired speed, which is expressed as

$$\min_{a} \quad \sum_{k=1}^{T_l/\Delta t_l} w_1 a\left(k\right)^2 + w_2 \left|v\left(k\right) - v^{\mathrm{d}}\left(s\left(k\right)\right)\right| + I^{\mathrm{c}}\left(z\left(k\right)\right),$$

where I^{c} is a function indicating whether any of the constraints related to driving task and collision avoidance is violated. At each step, if the boundary is *active*, the collision avoidance conditions are checked directly. A large cost will be added to I^{c} if there is a violation to the boundary. During forward expansion of the nodes, if the boundary is *potentially active*, the planner will check if the switch conditions are satisfied. When the algorithm expands a node satisfying the switch condition of a boundary, the indicator will be set as *active* or *inactive* at the current step. Inevitable Collision States (ICS) in the sense of all *active* boundaries are used as a heuristic.

To smooth the reference rough speed profile in the short-term horizon and reduce the sampling time, we formulate an optimization problem. Suppose $s_l\left(t\right)$ is the position provided by the long-term rough speed planning layer with sampling time Δt_l, and $s\left(t\right)$ is the position provided by the short-term speed smoothing layer with sampling time Δt_s, where $\Delta t_l = n\Delta t_s$. Then the optimization problem can be written as the QP form,

$$\begin{aligned} \min_{s} \quad & \sum_{t=\Delta t_s}^{T_s} w_1 \ddot{s}\left(t\right)^2 + w_2 \dddot{s}\left(t\right)^2 + \sum_{t=\Delta t_l}^{T_s} w_3 \left(s_l\left(t\right) - s\left(t\right)\right)^2, \\ \text{s.t.} \quad & A_s s \leq b_s, \end{aligned}$$

where T_s is the short-term preview horizon, $\ddot{s}$ and $\dddot{s}$ are longitudinal acceleration and jerk, corresponding to the driving comfort. A_s and b_s are constructed via linear hard constraints, such as longitudinal acceleration $a_{\min} \leq \ddot{s}\left(t\right) \leq a_{\max}$, and collision avoidance for (2.1)-(2.4) by setting $v = v_i$. An alternative for designing the longitudinal motion planner can be found in [79].

Short-term lateral motions

A simple illustration of the trajectory planner is shown in Fig. 2.5. Given the longitudinal positions from the smoothed speed profile, we can project them on the reference path, and create points with the same longitudinal positions in a Frenét Frame with lateral sample interval h. Then an A* search algorithm is designed with constraints on collision avoidance to lateral obstacles and feasibility according to the vehicle kinematics and dynamics.

Suppose $q_r\left(t\right)$ is the position of the rear axle center of the vehicle in A* search. $e\left(t\right)$ represents the edge connecting the points $q_r\left(t\right)$ and $q_r\left(t - \Delta t_s\right)$. The cost for A* search is set as

$$\min_{q_r} \quad \sum_{t=\Delta t_s}^{T_s} w_1 \left\|e\left(t\right)\right\| + w_2 \left|d\left(t\right)\right| + I_{\mathrm{colli}}\left(e\left(t\right)\right),$$

where I_{colli} is a function to set the cost to be a large value when lateral collision exists. Then heuristic is "holonomic-with-obstacles" distance used in [27], which can be calculated by dynamic programming. The admissibility can be easily proved and the optimality can be guaranteed.

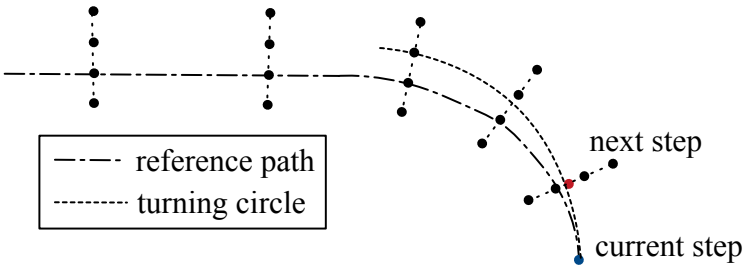


Figure 2.5: Illustration of the A*-based trajectory planner with tight turns.

When the vehicle is making a tight turn such as a U-turn, it is possible that the minimum cost node under forward expansion is out of the kinematics boundary of the vehicle. If so, the A* algorithm will also check the point on the kinematics boundary, as is shown in Fig. 2.5. The turning circle corresponds to the maximum curvature $\kappa_{\max}$. When expanding the nodes of the next step, the point with the minimal cost may be out of the kinematics boundary. Then the A* algorithm will check the red point on the kinematics boundary, and it may have the lowest cost. Then the point on the boundary will be expanded so that the autonomous vehicle can plan a trajectory with a maximal steering angle. Bicycle model can also be used as the kinematic model when lateral acceleration is low [95].

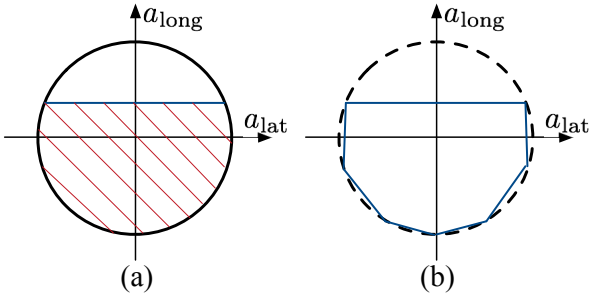


Figure 2.6: G-G diagram and its linearized approximation.

Since time stamps are associated with the each point, we can also check whether vehicle dynamics limits are satisfied. G-G diagram (shown in Fig. 2.6 (a)) is widely used for the high-level control of racing cars [33] to push the vehicle dynamics to the limits. The circle

corresponds to tire friction circle, and the blue line which bounds the maximum longitudinal acceleration corresponds to the engine traction limit. When expanding the node, longitudinal and lateral acceleration can be obtained and we can check the dynamics feasibility with emergency maneuvers according to the G-G diagram. More detailed discussions on kinematic and dynamic vehicle model near limits can be found in [4].

Trajectory smoothing

Finally, the trajectory smoothing for the autonomous vehicles can be formulated as a QP problem. The motion representation can be found in Figure 2.7 (a). $q(t) = [x(t) \quad y(t)]^T$ is the position vector of the rear axle center of the vehicle at time step t. The velocity, acceleration and jerk $\dot{q}$, $\ddot{q}$ and $\dddot{q}$ can be obtained via backward differences, and yaw angle $\phi(t) = \arctan \dot{y}(t)/\dot{x}(t)$.

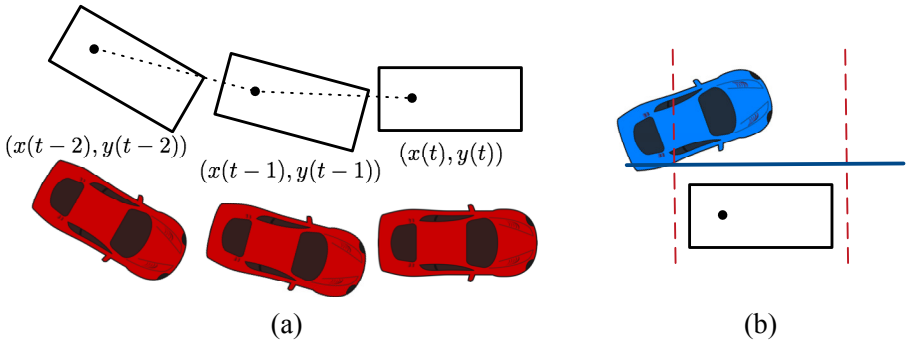


Figure 2.7: Motion representation of the QP problem and linearization of lateral obstacle avoidance constraints.

By taking into account the driving comfort and position error to the rough trajectory as the objective of optimization, the QP problem can be written as

$$\begin{aligned} \min_{q} \quad & \sum_{t=\Delta t_s}^{T_s} w_1 \|\ddot{q}(t)\|^2 + w_2 \|\dddot{q}(t)\|^2 + w_3 \|q_r(t) - q(t)\|^2, \\ \text{s.t.} \quad & Aq \leq b, \end{aligned}$$

where A and b are constructed via linear hard constraints. The linearized constraints can be found in Section 2.4 for longitudinal obstacles and in Fig. 2.7 (b) for lateral obstacles. The rough trajectory obtained from A* provides the rough position and yaw angle at each time step, from which we draw the rough vehicle body represented by a rectangle. We first draw two lines (red dashed lines) which are perpendicular to the rough direction with specific margin to the vehicle body. Then obstacles in the area between the two lines are considered

as lateral obstacles at the current time step. Then the boundary line (blue solid line), which is parallel to the direction, is pushed as far as possible until it hits the obstacle. Then the boundary line can be used as a linearized safety constraints for lateral obstacles. Moreover, given the rough direction of the vehicle, boundaries for vehicle dynamics can be constructed with linearized G-G diagram shown in Fig. 2.6 (b).

2.5 Illustrative Examples

In this section, exemplar use cases are shown to illustrate the performance of the proposed generic representation and motion planning framework. *Scenario 0* is not necessarily practical in real world, but involves all the three topological elements in one planning horizon to explain how the generic environmental representation works (without planning results). *Scenario 1* is a practical lane change scenario with a pedestrian to yield, which contains two topological elements to test the planning capability of the proposed framework. *Scenario 2* is an extreme case to test the capability of the proposed framework to handle emergency situations. When running the algorithm in python on a laptop with Intel Core i7-6600U 2.6 GHz CPU, the runtime corresponding to the worst case can be bounded within 0.2 s with $T_l = 10$ s, and 0.1 s with $T_l = 9$ s. Particularly, the A* search for rough speed profile dominated the runtime.

Scenario 0

An exemplar scenario in Fig. 2.8 to explain how the environmental representation works. The task of the host autonomous vehicle V_0 (red) is to overtake V_1 (blue) in front, return to the original lane to avoid the oncoming V_2 (purple) in the opposite lane, and then to follow the V_3 (green) that is merging into the lane from a ramp junction. V_0 does not have to yield to the pedestrian P_4 since it is relatively far away from the crosswalk.

The environmental representation in the position-time domain is shown in Fig. 2.9, in which T_l is the long-term planning horizon. Note that in order to represent the boundaries in Eq. (2.3) and (2.4) in the position-time domain, the speed of the host vehicle in Eq. (2.3) and (2.4) is set to be $v = v_i$. Note that it is not the real speed of the host vehicle.

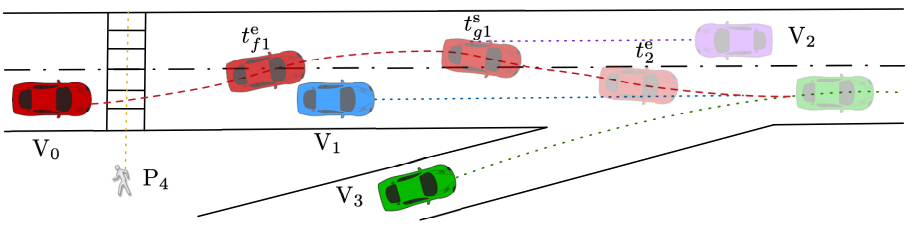


Figure 2.8: A complicated exemplar scenario.

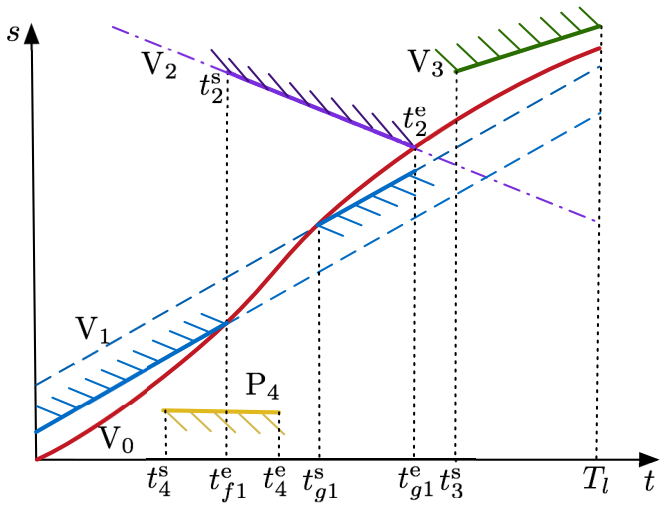


Figure 2.9: Environmental representation in position-time domain.

The activeness indicators of the boundaries corresponding to P_4 (*point-overlap*) and V_3 (*line-overlap*), namely I_4^{act} and I_3^{act}, are *active* within specific time periods (represented by solid lines). The indicators of the boundaries corresponding to V_1 and V_2 (*undecided-overlap*), I_{f1}^{act}, I_{g1}^{act} and I_2^{act}, are *potentially active* (represented by dashed lines). Given a rough speed profile of V_0 (speed profile candidates in search-based algorithms), the *active* range of boundaries of V_1 and V_2 can be determined. Before t_{f1}^{e}, V_0 follows V_1 and I_{f1}^{act} is *active*. Between t_{f1}^{e} and t_{g1}^{s}, V_1 should be on the right side of V_0, and V_1 can be regarded as a lateral obstacle. After t_{g1}^{s}, V_0 merges into the original lane, and V_1 can be regarded as a longitudinal obstacle behind. I_{g1}^{act} is *active* to achieve a safe and courteous cut-in, and it should be *inactive* after V_0 is completely on the target lane (t_{g1}^{e}). Between t_{f1}^{e} and t_{g1}^{e}, I_{f2}^{act} is *active* since V_0 is on the lane of V_2, which decides t_2^{s} and t_2^{e}.

Scenario 1

The first practical scenario that we used for the proposed planning framework was a challenging lane change with pedestrian yielding shown in Fig. 2.10. The task planner required the host autonomous vehicle (V_0) to merge to the target lane on the left ahead of a vehicle on that lane (V_1), which is relatively slow (5 m/s). Before completing the lane change, V_0 had to follow a vehicle on its lane (V_2), which is quite slow (3 m/s). Also, a pedestrian (P_3) approached the crosswalk to cross the street, and V_0 had to yield to the pedestrian.

The parameter settings were as follows. The long-term preview horizon for rough speed profile planning was $T_l = 10$ s, and the sampling time was $\Delta t_l = 1$ s. The short-term preview horizon for speed profile smoothing and trajectory planning was $T_s = 5$ s, and the sampling

time was $\Delta t_s = 1/3$ s. Lateral position sample interval $h = 0.25$ m. A rectangle was used to represent the vehicle body. $v_{\text{limit}} = 10$ m/s and $\kappa_{\max} = 0.2$ m^{-1}.

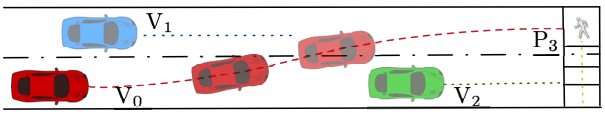


Figure 2.10: *Scenarios 1* illustration.

Fig. 2.11 reveals the boundaries for rough speed profile planning and the distance traveled by V_0 over time with the planned rough speed profile. The activeness indicators of the boundaries corresponding to P_3 (*point-overlap*), I_3^{act}, was *active* within specific time periods (represented by solid cyan lines). When the rough speed profile of V_0 is not planned, the indicators of the boundaries corresponding to V_1 and V_2 (*undecided-overlap*), I_1^{act} and I_2^{act}, are *potentially active* (represented by blue and green dashed lines, respectively).

We used the planning framework described in Section and 2.4 to plan a rough speed profile for V_0 (the solid red curve), and the *active* range of boundaries of V_1 and V_2 can be determined. At first, the boundary for following V2, $f_2 \leq 0$, remained *active*. V_0 accelerated to reach the boundary for overtaking V_1, namely, $g_1 \leq 0$. When V_0 reached the *potentially active* boundary, I_1^{act} is *active*. When the range was enough for V_0 to accomplish a relatively courteous cut-in, V_1 can be regarded as a longitudinal obstacle behind and I_1^{act} is *inactive*. Moreover, when approaching the crosswalk, V_0 tended to make a relatively sharp brake first and accelerate when the pedestrian had not completely left the conflict zone. The reason was that such maneuver maximized the time-efficiency of V_0 so that it could maintain higher speed which was closer to the desirable speed.

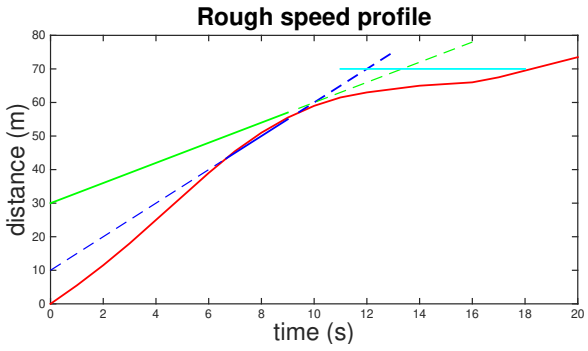


Figure 2.11: Rough speed profile and boundaries.

The rough speed profile helped the trajectory planner determine whether an obstacle should be a longitudinal or lateral obstacle at each time step. When I_1^{act} is *active*, V_1 was

regarded as a longitudinal obstacle behind V_0. Before that, V_1 was a lateral obstacle on the left; after that, V_1 was no longer considered. When I_2^{act} is *active*, V_2 was a longitudinal obstacle to follow. After that, V_2 was a lateral obstacle on the right.

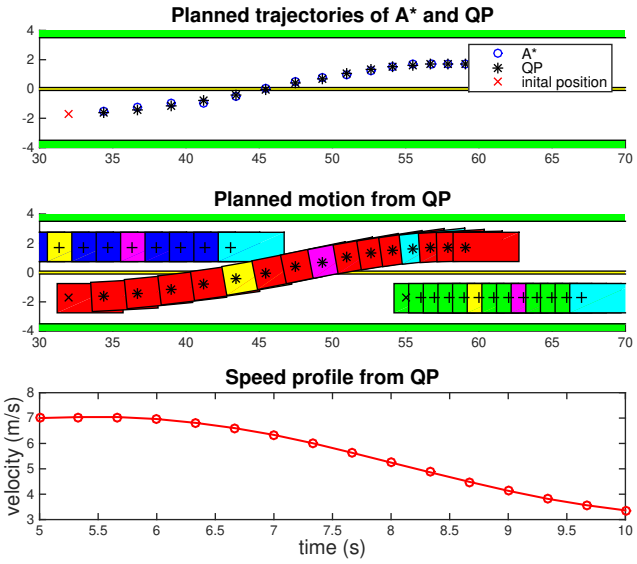


Figure 2.12: Planned trajectories.

Given the long-term speed profile, the A* search obtained a rough trajectory, and QP smoothened it to get the final trajectory. The trajectories obtained by A* and QP at time $t = 5$ s for a short-term horizon $T_s = 5$ s were compared in Fig. 2.12. As is demonstrated by the figure, the rough trajectory generated by A* provided a fairly good reference, and QP locally smoothened the trajectory to make it comfortable. Fig. 2.12 also demonstrated the relationship of the motions of V_0, V_1 and V_2 within the preview horizon. The three vehicles were marked with the same color at three representative time steps (yellow for $t = 6.66$ s, purple for $t = 7.66$, and cyan for $t = 9$ s). During $t = 6.66 - 9$ s, the boundary of V_1 is *active*, which means that V_0 reached the target lane at $t = 6.66$ s, and became the front vehicle of V_1 at $t = 9$ s.

Scenario 2

The second practical scenario considered an emergency maneuver at a two-way-stop intersection as is shown in Fig. 2.13. The host autonomous vehicle V_0 (red) was passing a two-way-stop intersection, following V_1 (green) and holding the right of way against any vehicle from the perpendicular cross street controlled by a stop sign. Deep blue cars, including V_3, were all static obstacles. V_2 (blue) was violating the stop sign with a relatively high

speed. Since the parked cars blocked the view of V_0, it cannot see V_2 until it was too late to apply emergency braking. V_0 had to accelerate to avoid being hit by V_2. However, V_3 temporarily blocked the lane and V_1 had to slow down and come to a stop. Also, there were oncoming vehicles V_4 and V_5. Hence, V_0 had to swerve to the parking lane on the right side to maneuver around. When reaching the parking lane, it had to slow down immediately to come a stop after escaping the conflict with V_2.

The parameter settings were the same as Section 2.5, and acceleration limits were $a_{\max} = 2.5$ m/s^2 and $a_{\min} = -7$ m/s^2.

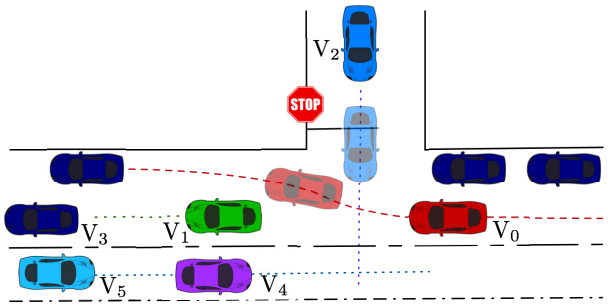


Figure 2.13: Illustration of the second exemplar scenario.

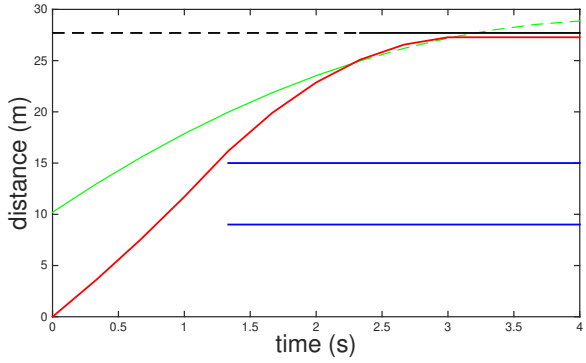


Figure 2.14: Rough speed profile and boundaries.

When applying the method in Section 2.4, no feasible solution can be found within the set of acceleration. Therefore, the rough speed planner resorted to plan a course of collision-free motions with a combination of constant acceleration and deceleration within $a_{\max}$ and $a_{\min}$. Fig. 2.14 reveals that it is impossible to apply emergency stop to make the distance under the lower blue bound created by V_2. Therefore, it has to accelerate to pass the upper blue bound. Fig. 2.15 demonstrates the trajectories generated by A* and QP, which reveals the

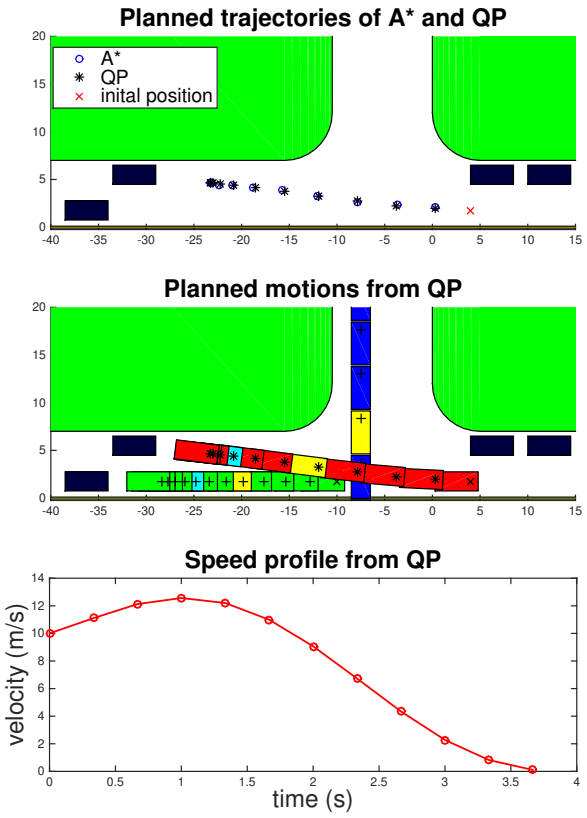


Figure 2.15: Planned trajectories.

performance of the planner to generate safe and feasible trajectories under an emergency situation. The three vehicles were marked with the same color at representative time steps (yellow for $t = 1.33$ s and cyan for $t = 2.33$ s).

2.6 Chapter Summary

A spatially-partitioned environmental representation and planning architecture was proposed for on-road autonomous driving in this chapter, which can generate safe and feasible trajectories with high driving qualities when dealing with a variety of complicated driving scenarios. The planning framework spatially partitioned longitudinal and lateral motions instead of path and speed in the temporally-partitioned architecture. Three topological elements and corresponding longitudinal boundaries were proposed to compose the representation of all types of on-road driving scenarios discussed in this chapter. A* search plus

quadratic programming (QP) planners were designed for long-term longitudinal motions and short-term trajectory to leverage the advantages of each method, and the linearization of safety and feasibility constraints obtained from the proposed methodology. As was demonstrated in the simulation results, the proposed planning framework can generate high-quality and collision-free trajectories in the long term in complicated driving scenarios and emergency situations.

Chapter 3

Imitation Learning with Constraints

3.1 Introduction

Autonomous vehicles need to plan trajectories with desirable driving qualities while avoiding collisions with obstacles on the road. Moreover, the motion of other vehicles should be predicted within a specific horizon. Motions generated by planning and prediction module of an autonomous vehicle should be 1) safe to avoid collisions, 2) feasible according to the vehicle kinematics and dynamics, 3) human-like with high driving quality and 4) real-time with low computational load.

Conventional motion generation methods are usually based on techniques such as search, optimization and optimal control. A representative work is the spatiotemporal nonlinear programming [145] implemented in the demonstration of Mercedes-Benz autonomous car. An optimization problem was formulated by comprehensively considering comfort, speed, road structure, as well as safety and feasibility constraints to achieve smooth and collision-free trajectory planning. Inequality constraints were employed for satisfying safety and feasibility. [5] proposed a planning method based on model predictive control (MPC) near the boundaries of safety and feasibility with a relatively simple dynamic model. For more critical situations in the sense of dynamics feasibility, a new concept of critical instability region for steering maneuver was proposed in [75], which explicitly provided the theoretical upper bound of vehicle stability and promisingly expanded the vehicle stability control region.

Conventional methods are capable of satisfying safety and feasibility with hard constraints and achieving high driving quality with comprehensive cost functions. However, they lack the flexibility to learn the human-like behavior from data. Also, the computational complexity increases significantly when the environment is complicated. For instance, the runtime of the algorithm in [145] is 0.5 s, which is not desirable for replanning of the vehicle, especially in emergency situations. On the contrary, neural-net-based methods have powerful capability to learn driving policies from human. The computational load can be extremely low when the size of the net is small. However, it is hard to make the net learn how to push the vehicle near the safety and feasibility boundaries to achieve high-quality and human-like motions

with satisfied constraints.

A typical neural-net-based method is the end-to-end paradigm utilized in [15], [129], [132] and [35], which learns the driving policy of human drivers via convolutional neural networks (CNN) with front-view images as the input and steering maneuver as the output. The end-to-end paradigm is effective for lateral maneuvers under relatively simple tasks. It becomes challenging for the paradigm in complicated scenarios, especially when combinatorial predictions and planning are involved, and the obstacles or road structures are beyond the field of view. Also, [112] demonstrated that such paradigm requires exponentially larger training samples than those required by modular paradigm with semantically meaningful components.

Instead of providing steering maneuvers, neural-net-based methods were brought forward to generate sequences of motions with better interpretability. [113] proposed a reinforcement learning architecture based on recurrent neural network (RNN) to plan long-term longitudinal motions of the host vehicle via predicting short-term motion of other vehicles to handle challenging scenarios such as roundabout merging. However, safety and feasibility were not addressed for the autonomous vehicle on a 2-dimensional plane.

In order to satisfy safety and feasibility constraints of the vehicle, conventional and learning-based methods were combined to exploit their advantages. [117] proposed an integrated planning and control framework to learn the planning policy from a long-term MPC planner with imitation learning, and a low-level control layer guaranteed short-term safety and feasibility. [111] brought forward an architecture to learn the desire of the autonomous vehicle via reinforcement learning, and safety constraints were satisfied via a subsequent short-term optimization. [21] utilized CNN for the detection of other vehicles and localization of the host vehicle to provide key indicators for a rule-based controller. [134] employed a logistic regression model to learn the intention probabilities of other road users, and proposed an integrated decision-making and planning framework to continuously adjust the motions of the host vehicle according to the probability of the threat, while safety and feasibility constraints were satisfied in the worst case.

The aforementioned works combined conventional and learning-based methods hierarchically to achieve the learning flexibility, and to satisfy safety and feasibility. It is also worth exploring to extend the capabilities of learning-based models to satisfy the safety and feasibility constraints by incorporating the domain knowledge from conventional methods. [56] addressed safety-critical issues for collision avoidance problem of drones with policies based on neural networks. The loss function for the policy training contains not only the Kullback-Leibler (KL) divergence between the expert output and the policy net, but also the cost function of the expert MPC-based planner, so that the number of collisions are significantly reduced. The distance-based cost function could make the robot conservative to keep in the middle of several obstacles and get rid of the safe area near the obstacles for high-quality motions. [91] proposed a constrained CNN which can incorporate linear constraints to the loss function for classification problems. However, the motion generation of autonomous vehicles is a regression problem with nonlinear hard constraints on safety and feasibility.

In this chapter, Constrained Policy Net (CPN) is proposed to incorporate the domain knowledge of an optimization-based expert planner into the training of the net so that inequality constraints on safety and feasibility can be satisfied. The CPN can be directly trained to generate safe and feasible motions for autonomous vehicles. We address the problem of training a policy net to approximate inequality-constrained receding horizon trajectory optimization so that the trajectories can be generated near the boundaries of the constraints to achieve high-quality and human-like motions, while safety and feasibility constraints can be satisfied.

The remainder of the chapter is organized as follows. Section 3.2 provides the concept of the proposed CPN and Section III discusses an exemplar scenario to apply CPN; results are provided in Section IV; and Section V concludes the chapter.

3.2 Concept of Constrained Policy Net (CPN)

Our task is to train a policy net to generate safe and feasible motions which can mimic the output of an optimization-based expert planner in similar situations. The input of the net is a set of features describing the states of the host vehicle and the environment. The output is a sequence of motions of the autonomous vehicle, which can be executed by the lower-level layer. The design of the optimization-based expert planner employs lots of domain knowledge on driving quality as well as safety and feasibility. Our goal is to incorporate the domain knowledge of the expert planner into the policy so that the motions generated by the policy satisfy safety and feasibility constraints.

An intuitive solution is supervised learning (SL) shown in Fig. 4.3(a), to train a policy net via minimizing the loss function with L_2 norm of the difference between expert motions $\hat{q}$ and motions generated by the policy net q. However, SL using L_2 norm as the training loss cannot satisfy safety and feasibility constraints. We can take the motion generation for billiards as examples. [32] used L_2 norm as the loss to train a deep net to predict the motion of billiards. The motion patterns were learned correctly, while oscillation violating laws of physics still existed. In [82], sharp predicted images of billiard motions were generated via training a generative adversarial network (GAN). A possible reason why SL using L_2 norm is not appropriate for motion generation was explained in [82]. If the distribution has two equally likely modes, the average of the two modes minimizes the L_2 loss over the data, even if the average has very low probability. Therefore, using L_2 norm as the loss averages generated motions among several possible modes and the average may be a sequence of motion which is unsafe or infeasible.

Instead of training a discriminator of GAN, we resort to a training framework which can directly incorporate the objective and constraints of the expert planner as the loss function to eliminate unsafe and infeasible motions. The challenge is how we can obtain a policy net generating motions near the boundaries of the hard constraints without violation. Motions violating constraints should be abandoned, while safe and feasible motions are retained.

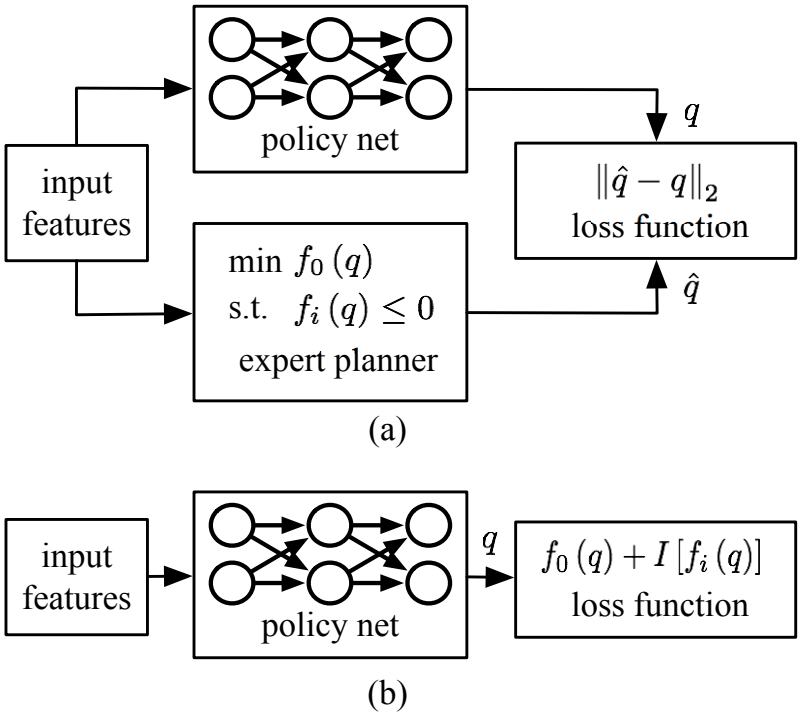


Figure 3.1: Training architecture of typical supervised learning and Constrained Policy Net

The safety and feasibility constraints for the motion generation of autonomous vehicles can be expressed as inequalities on collision avoidance as well as vehicle kinematics and dynamics, such as the curvature constraints used in [145] and G-G diagram used in [139]. Therefore, the expert planner can be written as the following generalized form, which is an inequality-constrained optimization problem.

$$\begin{aligned} \min_{q} \quad & f_0(q), \\ \text{s.t.} \quad & f_i(q) \leq 0, i = 1, ..., n, \end{aligned} \tag{3.1}$$

where q is a sequence of vehicle motions.

In the context of solving an inequality-constrained optimization problem, hard constraints are often transformed into the objective via a barrier function, which grows extremely fast when the corresponding constraint is violated and remains to be zero when the constraint is satisfied. For instance, [135] employed an exponential function to transform constraints into the objective in the iterative linear quadratic regulator for trajectory planning. For the aforementioned inequality-constrained optimization problem, a differentiable barrier function $I[f(x)]$ can be used to construct an unconstrained form of optimization problem, that is,

$$\min_{q} \quad f_0(q) + I[f_i(q)]. \tag{3.2}$$

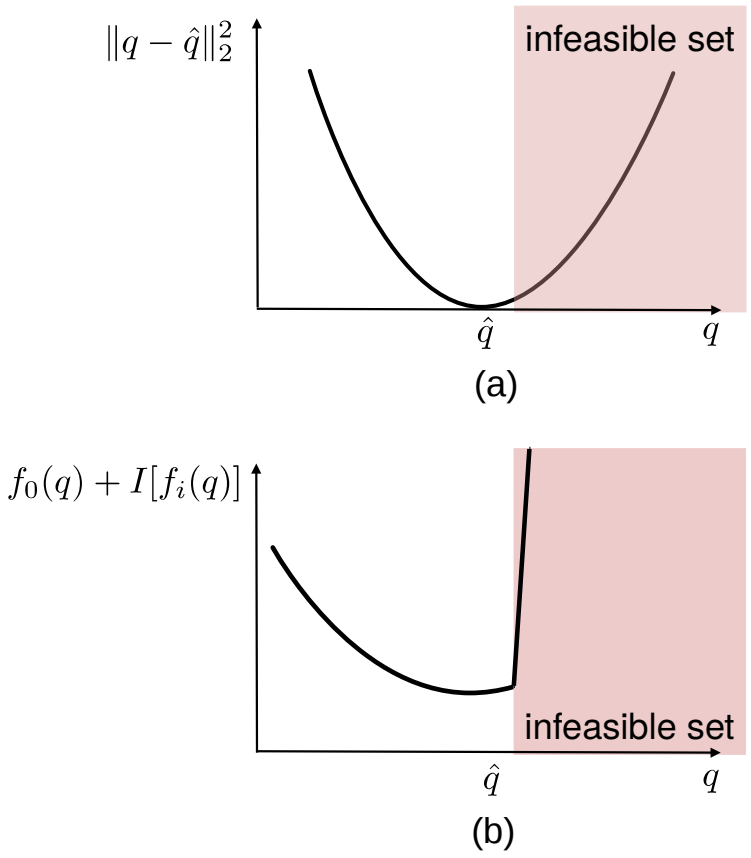


Figure 3.2: Comparison of costs near the boundaries of infeasible sets for different loss functions.

Suppose an infeasible set is composed by motions which are not safe in the sense of collision avoidance, or not feasible according to vehicle kinematics and dynamics. A comparison of costs near the boundaries of infeasible sets is provided in Fig. 3.2 for different loss functions, namely, the L_2 norm (Fig. 3.2 (a)) and the cost function of the unconstrained optimization (Fig. 3.2 (b)). For L_2 norm loss function, the cost just follows a quadratic function regardless of safety and feasibility of the motion. On the contrary, the cost function of the unconstrained optimization generates extremely large penalties in the infeasible set. The penalty decreases rapidly near the boundary of the infeasible set, which forces the trained policy to be safe and feasible.

We can use the cost function with differentiable barrier functions to incorporate the domain knowledge in the expert planner to train our policy net π_θ to approximate the performance of the expert planner and not to output motions which are infeasible or unsafe. Suppose θ is the a set of weight of the policy net, $q(\theta)$ is the output of the policy net π_θ,

then the policy optimization problem to solve is

$$\min_{\theta} \quad \sum_{\mathcal{D}} f_0\left(q(\theta)\right) + I\left[f_i\left(q(\theta)\right)\right], \tag{3.3}$$

where $\mathcal{D}$ is the generated dataset for training.

The training architecture is shown in Figure 4.3(b). The process can be briefly described as follows.

1 Large amounts of cases are generated as training set $\mathcal{D}$. The input features of each case are randomly generated within a specific range so that a safe and feasible trajectory exists in the case.

2 The policy net outputs sequences of motions q corresponding to each case.

3 The motions are evaluated by the loss function we designed according to the unconstrained form of the expert planner.

4 The gradients are calculated via backpropagation to minimize the loss function.

5 Update the parameters in the policy net.

2-5 are repeated to solve the problem (6.5) until the change of the loss is small enough between iterations.

Note that there is no restriction to the form of the inequality-constrained optimization problem. It can be a complicated nonlinear and nonconvex optimization problem as long as all terms are differentiable. Therefore, CPN provides us a powerful tool with super fast online computation capability to learn the policy of a complicated optimization-based expert planner, which can hardly be implemented onboard in real time. The trained policy net can learn the ability of the expert planner to satisfy safety and feasibility constraints with high driving quality. The net also has the flexibility to be trained by data collected from human drivers to learn human-like behaviors.

3.3 An Exemplar Scenario

In this section, we use an exemplar scenario (shown in Fig. 4.2) with two-dimensional motion of the host autonomous vehicle to apply the proposed Constrained Policy Net (CPN). The host autonomous vehicle (red car) needs to plan the path to avoid static lateral obstacles (blue and yellow car).

Fig. 4.4 shows how we represent the motions of the host vehicle in a 2D space. The x-axis is the forward direction along the road and the y-axis is perpendicular to the forward direction. The rectangle represents the host vehicle at each step. The yellow dots are the rear axle centers of the vehicle. h is the sampling distance and ϕ is the yaw angle. We can use the constraints of the lateral obstacles on left (upper) and right (lower) sides of the host

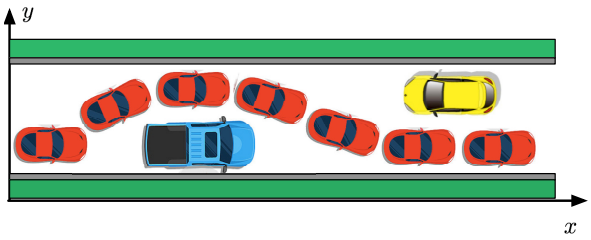


Figure 3.3: An exemplar scenario.

vehicle as the input features to describe the environment. In Fig. 4.4, u_i and l_i are the upper (blue dots) and lower (red dots) bounds at the ith step, respectively, which are used to formulate safety constraints.

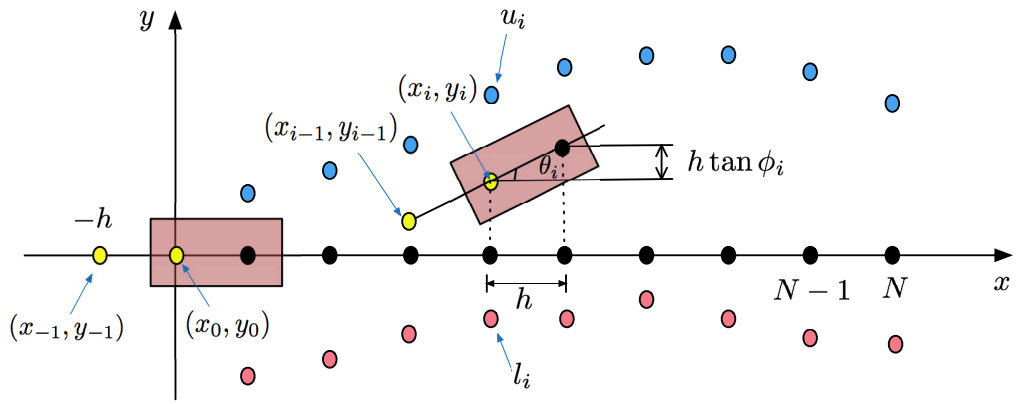


Figure 3.4: Motion in a 2D space.

Note that we predefine the rear axle center positions of the vehicle at the first two steps: (x_{-1}, y_{-1}) and (x_0, y_0) and let $y_{-1} = y_0 = 0$. The reason is that we want to set (x_0, y_0) as our starting point and make sure that the starting position has a curvature equal to zero to represent the normal driving direction of the vehicle. Since the current forward direction of the vehicle is determined by connecting the previous rear axle center with current rear axle center, we need to place the previous rear axle center on the x-axis.

For the exemplar scenario, two aspects need to be considered for the driving quality. On one hand, the vehicle should follow the reference, which is the x-axis. Thus the deviation from the reference is penalized. On the other hand, we penalize curvature to enhance comfort. Suppose $q_i = \begin{bmatrix} x_i & y_i \end{bmatrix}^T$, $q = \left[q_1^T, q_2^T, \ldots, q_N^T\right]^T$, then the cost function can be written as

$$f_0(q) = w_1 \sum_{i=1}^{N} y_i^2 + w_2 \sum_{i=1}^{N} \kappa_i^2, \tag{3.4}$$

where y_i is the deviation from the reference and

$$\kappa_i = \frac{\dot{x}_i \ddot{y}_i - \dot{y}_i \ddot{x}_i}{(\dot{x}_i^2 + \dot{y}_i^2)^{3/2}}$$

is the curvature at the ith step.

For the feasibility constraints, the curvature of the vehicle needs to be bounded at each time step, that is, $|\kappa_i| \leq \kappa_{\max}$. Then the inequality constraints can be written as

$$\begin{aligned} f_1(q_i) &= \kappa_i - \kappa_{\max} \leq 0, \\ f_2(q_i) &= -\kappa_i - \kappa_{\max} \leq 0. \end{aligned}$$

For the safety constraints, the vehicle body should be within the range of upper and lower bounds defined by the lateral obstacles. The inequality constraints can be expressed as

$$\begin{aligned} f_3(q_i) &= y_i + b - u_i \leq 0, \\ f_4(q_i) &= -y_i + b + l_i \leq 0, \\ f_5(q_i) &= y_i + h \tan \phi_i + b - u_{i+1} \leq 0, \\ f_6(q_i) &= -y_i - h \tan \phi_i + b + l_{i+1} \leq 0, \end{aligned}$$

where b is a predefined safety distance composed by a half of the vehicle body width and the safety margin. At the ith step, $f_3(q_i)$ and $f_4(q_i)$ bound the rear axle center, and $f_5(q_i)$ and $f_6(q_i)$ bound the front part of the vehicle body corresponding to the upper and lower bound of the next step. Thus we can formulate an inequality-constrained optimization problem with the form in (6.3).

In order to transform the inequality-constrained form in (6.3) to the unconstrained form in (6.4), we use a rectified linear unit (ReLU) as the barrier function. It is a linear function when there is a violation in f_1-f_6, and is set to be 0 when the constraint is satisfied. Other kinds of barrier functions, such as exponential barrier function, are also implemented, but the final performance is worse. A possible reason is that for the near-limit violations the exponential function decays much slower than ReLU function which can have much larger slope. Thus the violation cannot be sufficiently penalized.

Then the optimization problem of CPN can be written as follows to obtain desirable policy π_θ.

$$\min_\theta \quad \sum_{\mathcal{D}} \left[f_0(q(\theta)) + \sum_{i=1}^{N} \sum_{j=1}^{6} c_j \max\left\{0, f_j(q_i(\theta))\right\} \right],$$

where c_j are the slopes of each constraint in the ReLU function. The input of the CPN is a set of upper and lower bounds u_i and l_i at each step, and the output is a sequence of lateral positions y_i at each step to compose the path.

Note that the path planning problem in the exemplar scenario can also be regarded as a trajectory planning problem if the spatial steps are regarded as temporal steps, and an assumption is made that the host vehicle will keep its speed along the x-axis. Then the lateral obstacles can be static or moving obstacles. For moving obstacles, their predicted motions in the preview horizon of the host vehicle should be provided, which is a reasonable assumption. Then the worst case boundaries at each time step can be obtained, which are the bounds on left (upper) and right (lower) sides of the host vehicle as the input feature to describe the environment.

3.4 Results

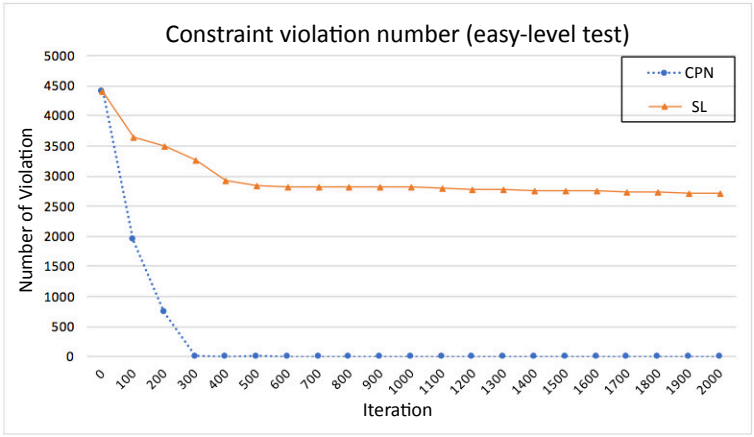
In this section, we apply the proposed approach to train a Constrained Policy Net (CPN) for the exemplar scenario, and test the performance by challenging cases in comparison with the baseline optimization, as well as supervised learning (SL).

In order to train a widely applicable policy net to generate safe and feasible motions in the exemplar scenario, the input features of the training cases should be diversified. To avoid overfitting in the training process, the input lateral bounds should cover different patterns via random generation.

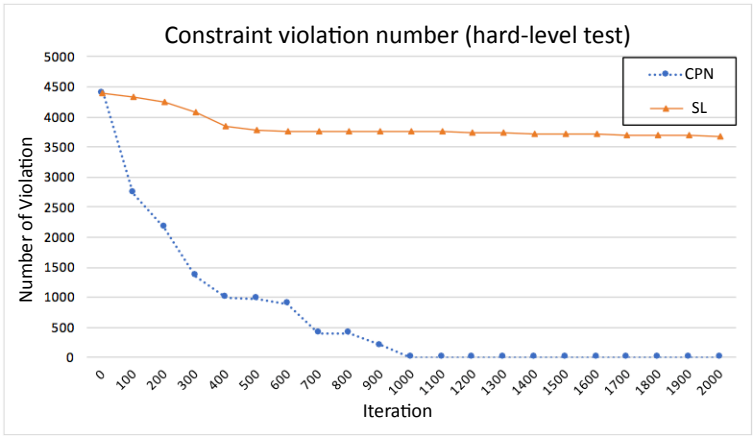
The training dataset contains two parts. The first part was generated based on random feasible paths. We first randomly generated a large number of feasible paths in the 2D space under feasibility constraints on curvature. Then the area covered by the vehicle body according to the feasible paths were used to generate upper and lower bounds. The bounds were randomly generated within proper range above or below the area. For the second part, we generated large amounts of cases with 25 manually designed patterns, which are common in real-world scenarios. Specifically, we generated obstacles of various shapes at different positions of the road, and the sizes of obstacles in each case were randomly generated within a proper range. In the training dataset, the ratio of the cases generated from feasible path and manual design was 1:3, which led to the desirable performance according to our attempts. The total number of cases in the training set was 10000.

The test dataset was generated from manual design, and it was designed with two levels: *easy* and *hard*. The lateral bounds in the *easy* level had similar shapes to those in the *hard* level but with simpler obstacles and smaller curvatures. No same cases can be found in training and test dataset. The total number of cases in the test set is 5000 for each level.

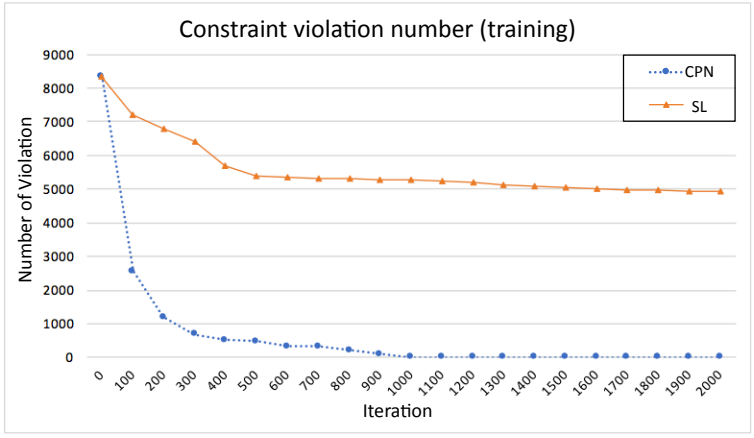
The model of the CPN was a fully connected net with one hidden layer of 600 nodes with ReLU nonlinearity. We set $N = 10$ steps as our horizon. Then the input size was 20 with upper and lower bounds, and the output size was 10 with lateral positions at each step. CPN needed much more effort to enable the backpropagation process since the barrier functions with nonlinear constraints were incorporated into the loss function. The gradients of parameters θ needed to be calculated in sequence with respect to each output node. We trained 2000 iterations to obtain the parameters. The ratio of the weights of the cost function in (6.1) was set as w_1:w_2 = 1:1000.



(a)



(b)



(c)

Figure 3.5: Numbers of rear upper constraint violations over training iterations.

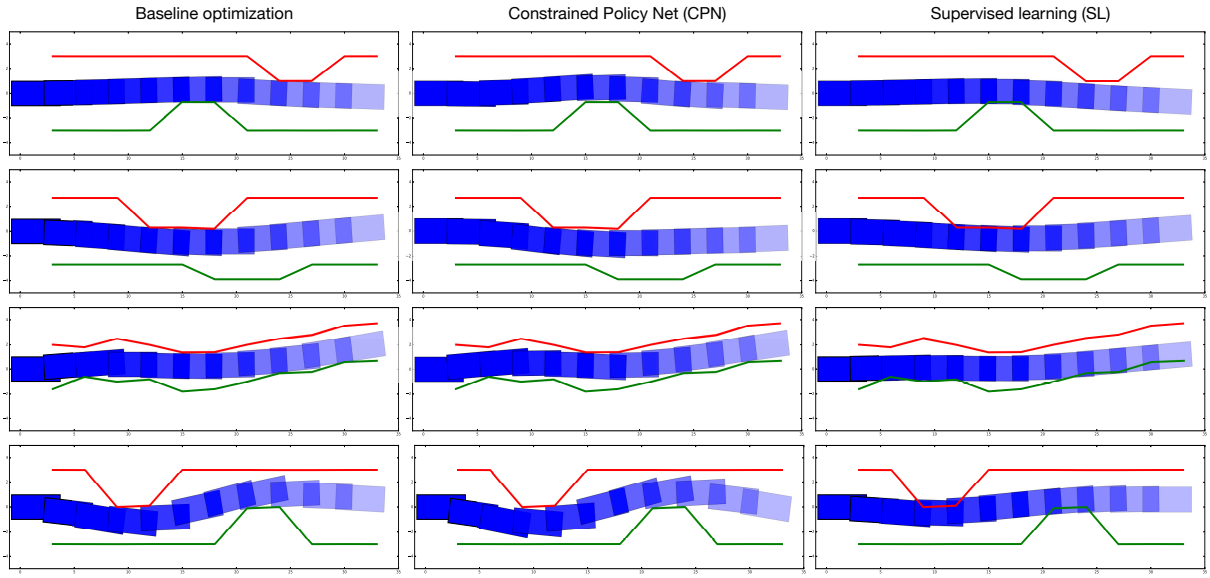


Figure 3.6: Visualized motions.

We recorded the weights θ of the CPN at every 100 iterations of the training process, and tested the performance via the test dataset with the two levels of difficulty. The results of constraint violation numbers over iterations with the test data of *easy* and *hard* level can be found in Fig. 4.5 (a) and (b), respectively. The violation number with training data is shown in Fig. 4.5 (c). Fig. 4.5 (a) demonstrates that the number of violations rapidly decreased to zero, which means that it was relatively easy to train a net to satisfy constraints for such difficulty level of cases. In Fig. 4.5 (b), the number of violations also became zero, but several hundreds of more iterations were needed to achieve such performance due to the more difficult cases to handle. For both *easy* and *hard* test sets, the violation number of SL remained large. An intuitive reason was that SL only tried to minimize the distances to the expert outputs. For cases in which the expert motion was near the boundaries of the constraints, the policy net trained via SL would not receive enough penalty when the output violates the constraints. On the contrary, CPN was sufficiently penalized for such cases with the barrier function.

The visualized generated motions can be found in Fig. 4.6, in which the performance of the policy nets trained by CPN and SL were compared. We also provided the results from the baseline optimization by directly solving the problem in (6.3). Four cases were used to test the performance. The first two cases were *easy*, while the last two were *hard*. As demonstrated in Fig. 4.6, the policy net trained by SL led to violations near the safety boundaries. It tended to generate the "mean" of possible motions of various kinds of cases. By contrast, CPN generated motions which were very similar to the baseline optimization with much less computation time. More importantly, there was no violation to the safety

and feasibility constraints, even in complicated cases such as the third one.

3.5 Chapter Summary

In this chapter, Constrained Policy Net (CPN) was proposed to learn a driving policy to generate safe and feasible motions for autonomous vehicles. An exemplar scenario with lateral obstacles was used to apply the idea of CPN. We first designed an baseline expert planner with receding horizon optimization, which included driving quality as objectives, and safety and feasibility as hard constraints. Then a differentiable barrier function was used to transform the inequality-constrained optimization problem into an unconstrained form to formulate the policy optimization problem, which can incorporate the domain knowledge in the expert planner directly. The test results with challenging cases demonstrated that the CPN was able to learn a driving policy to generate motions which are safe to avoid collisions and feasible according to vehicle kinematics with similar desirable driving quality to the baseline optimization.

Part II

Decision and Planning under Uncertainty

Chapter 4

Non-Conservatively Defensive Strategy (NCDS)

4.1 Introduction

Urban autonomous driving is very challenging since the host vehicle has to handle various kinds of scenarios and the behaviors of other road participants are highly unpredictable. Variations in intentions of others [76][120] and uncertainties in subsequent motions [71] may lead to different decisions and resultant motions taken by the host autonomous vehicles. Typically, the relationship between decision-making and motion planning is hierachical [144]. However, for autonomous vehicles in dynamic environment, taking an immediate action does not mean that it has to immediately make a final decision among possible future actions. Tentative actions are commonly seen in human driving, which is ambiguous and allows the diversification of possible subsequent actions that can be transitioned smoothly in the future. The following challenging urban driving scenarios are provided to illustrate why decision-making for ambiguous intentions of others can be incorporated in motion planning by taking actions tentatively, which enables a non-conservatively defensive strategy (NCDS).

Scenario 1: Avoiding violation vehicles at intersections. Careless drivers may violate stop signs or red lights and cause fatal accidents [107]. In order to be defensive to red light runners, Google programmed a short pause after light turning green before its car proceeds into the intersection [121]. Also, stop sign violation is not an extremely rare behavior in real world traffic. Google self-driving car was crashed by a car violating stop sign in February 2015 [38]. When observing a vehicle that may run the stop sign or red light with high speed and low deceleration, an autonomous vehicle should be prepared to avoid a possible crash. However, it does not mean that the autonomous vehicle should slow down prematurely to make a yielding decision. If the violation probability is relatively low, it can maintain its speed as long as its braking capability can allow it to stop timely before reaching the conflict region.

Scenario 2: Roundabout entering. When an autonomous vehicle is entering a roundabout,

some vehicles in the roundabout may act aggressively to deter the host vehicle from merging but other non-aggressive vehicles may slow down and yield to the host vehicle [113]. It was observed in the test in Parma [18] that the VisLab autonomous car, when entering a roundabout, acted conservatively so that there were long pauses and unnecessary stops before proceeding even though the other vehicle in the roundabout had already started to turn into other road branches. Therefore, in planning to enter the roundabout given the uncertainties above, an autonomous vehicle does not have to decide immediately whether to merge before or after the other vehicle in the roundabout coming to its entrance point. It can just keep a proper speed and further observe the motion of the vehicle in the roundabout, and make final decisions when it has to.

Scenario 3: Four-way-stop intersection entering. At a busy four-way-stop intersection, an autonomous vehicle can hardly move forward if it strictly obeys the rule and behave cautiously, waiting behind the stop bar for its turn. This case was observed in a demonstrated situation that the Google self-driving car faced. Google decided to enable the car to move forward a little to show its determination to go. In fact, the principle behind such human-like behavior can be explained as follows. It is possible that all vehicles from other approaches of the intersection may yield. Hence the host vehicle can start to accelerate and show its intention to go first. However, to stop again and yield to others is still possible in case any of them shows stronger determination so that the host vehicle has to yield.

Scenario 4: Lane change. When an autonomous vehicle is changing lane, other vehicle may speed up to prevent it from cutting in. Drivers of large vehicles on the target lane tend to assume that the autonomous vehicle would not risk to cut in closely, so that they just maintain the speed anyway. That was how the accident happened on the Google car on February 2016 [39]. Moreover, another vehicle may be merging into the same lane from the other side at the same time [78]. Therefore, an autonomous vehicle should be prepared to change back to its original lane when such threats appear. Starting the movement to change lane is also a tentative motion and the final decision does not need to be made at that point.

In all the aforementioned scenarios, autonomous vehicles should execute a defensive driving strategy to avoid possible collisions when potential threats exist. However, the current demonstrated design of autonomous vehicles is often overly cautious and sometimes behaves unhuman-like in real world scenarios in order to guarantee safety. Such conservative behaviors will degrade the driving quality and may still jeopardize safety when the behaviors are not expected by other road participants. To address this problem, we propose that autonomous vehicles need a unified planning framework to handle uncertainties in various kinds of urban driving scenarios when the decision and action are tentative, so that the driving strategy is defensive enough to guarantee safety even when others are violating traffic rules, and yet not too conservative to degrade driving quality.

In literature, partially observable Markov decision process (POMDP) was used extensively in autonomous driving for decision making and planning under uncertainty [16][10]. However, there is no guarantee for the planned trajectories to be collision-free within a specific preview horizon regarding each possible intention of the others. In this chapter, we emphasize on the guarantees of the safety of the autonomous vehicle even when the others

choose the worst case behavior.

The rest of this chapter is organized as follows. Section II models the behavior of others via simplifying the uncertainty quantification and providing the model in an exemplar scenario. A non-conservatively defensive strategy (NCDS) is detailed in Section III. A deterministic planner is designed as a baseline. Then a safe set is defined, and a unified planning framework under uncertainty is proposed. Then Section IV gives illustrative examples, and Section V concludes the chapter.

4.2 Behavioral Modelling

In this section, we model the behavior of other road participants via calculating the probabilities of the cases that lead to passing and yielding decision of the host autonomous vehicle. A logistic regression model is then provided for an exemplar scenario to get the probabilities of other vehicles to violate the stop sign or not.

Simplified uncertainty quantification

In order to define the boundary for moving obstacles at each future time step, the behaviors of other road participants within the preview horizon should be predicted, which are full of uncertainties in urban driving scenarios. In fact, it is beyond the scope of this chapter to model all uncertainties in object detection, intention recognition and motion prediction in various kinds of urban driving scenarios. In the following we will discuss how the uncertainty quantification can be simplified to facilitate decision-making and motion planning.

Although a road participant may exhibit varying intentions and there can be uncertainties in recognizing the object, an autonomous vehicle typically has just two choices in the current preview horizon – passing or yielding. This simplified description is applicable to various kinds of urban driving scenarios, such as merging into a roundabout, passing through an intersection, changing into another lane, or yielding to crossing pedestrians. The significance of solving the problem with two cases is not reduced by the fact that multiple moving objects may exist and possible decisions may not be limited to just two cases. Usually, they can be clustered and two best decisions can be created for the proposed framework to plan motions. Also, the accidents and overly conservative behaviors of autonomous vehicles mentioned in Section I were all caused by inappropriate motions with two ambiguous decisions. Moreover, the framework has the potential to be extended to handle more than two cases.

Uncertainties originating from different causes can be combined. For instance, uncertainties from both perception and intention recognition systems may co-exist on whether an object is a pedestrian and whether the pedestrian intends to cross the street. For the yielding case, the motion planner only considers the probability of the event that the object is a pedestrian and the intention of the pedestrian is to cross the street. The passing case then encompasses the remaining possibilities.

For the potential threats beyond the field of view, prior knowledge is used to obtain the probabilities. The probabilities are constant over time until the view reveals the suspected region. When more information becomes available, then the probability assessment is updated.

In summary, by observing the motions of other road participants with contextual information, the probabilities for an autonomous vehicle to pass and yield are obtained, which are denoted as $P\,(\text{pass})$ and $P\,(\text{yield})$, respectively. Then under each case, the worst possible boundaries generated by the motion are constructed as the predicted motion. Although it is possible for weird behaviors to happen, the bounds we introduced are meaningful as 1) they are reasonable assumptions anticipated by human drivers; and 2) they cover cases with low probability.

Specific models and learning methods based on different approaches can be adopted to quantify the uncertainties and create constraint boundaries for different road participants under various conditions. In the next subsection, a possible method to model the uncertainties in an exemplar scenario is shown as an example.

An exemplar scenario

The exemplar scenario we use is a two-way-stop intersection, which is shown in Fig. 4.1. The autonomous vehicle V1 (red car) holds the right of way, and the other vehicle V2 approaching the stop sign (orange car) is expected to stop. We model the behavior of V2 to calculate the violation probability.

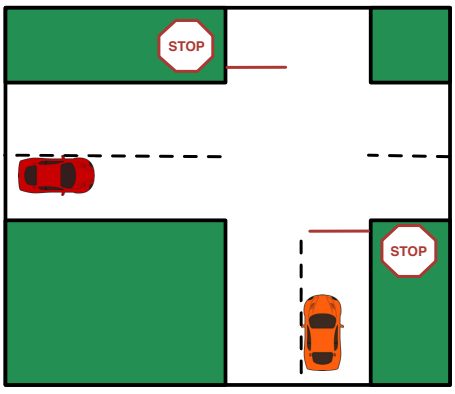


Figure 4.1: A two-way-stop intersection.

We carried out a field observation effort at a real-world intersection to collect motion data of vehicles approaching stop bars. Features including distance to the stop bar d_{stop}, velocity $\dot{d}_{\text{stop}}$ and acceleration $\ddot{d}_{\text{stop}}$ are chosen to represent the motions. Sample data is illustrated in Fig. 4.2.

Among the V2 motions observed from the field data, slight rolling stop motions are frequently observed, which are labeled and grouped together with the full-stop motions as

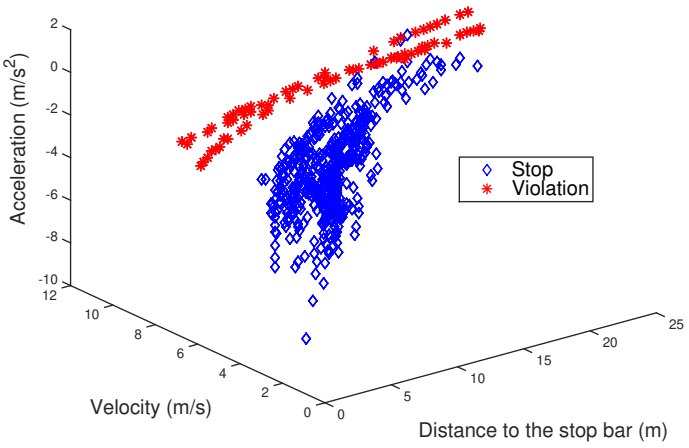


Figure 4.2: Visualized 3D taining data

"stop cases" in Fig. 4.2. The reason we regard slight rollings as cases that V1 can pass is that drivers of those slight rolling vehicles are likely paying attention to the cross traffic and are ready to stop if a vehicle holding the right of way appears [88].

A logistic regression model is employed to obtain the probabilities to quantify the uncertainty based on the motion features of V2, which can be written as

$$P\left(\text{pass}|z\right) = \frac{e^{-\beta^T z}}{1+e^{-\beta^T z}},$$
$$P\left(\text{yield}|z\right) = 1 - P\left(\text{pass}|z\right),$$

where

$$z = \begin{bmatrix} d_{\text{stop}} & \dot{d}_{\text{stop}} & \ddot{d}_{\text{stop}} & 1 \end{bmatrix}^T,$$

the passing case corresponds to stop of V2 and yielding case corresponds to violation of V2. The vector β is obtained by maximizing the log likelihood based on the Newton-Raphson method.

For the passing case, no blocking line is created. For the yielding case, blocking lines are created for V1 in a time period from the earliest possible time for V2 to enter the conflict region to the latest possible time for V2 to leave the conflict region.

4.3 Non-Conservatively Defensive Strategy

In this section, a non-conservatively defensive driving strategy (NCDS) is proposed for urban autonomous driving scenarios, which is defensive to deal with potential threats to guarantee safety, but not overly conservative to degrade driving quality.

Deterministic planner

In this section, a spatiotemporal trajectory planner in deterministic environments is designed by adopting a modified receding horizon optimization framework [145], which is solved by sequential quadratic programming. $q_i = [x_i \quad y_i]^T$ is the position vector of the autonomous vehicle at time step i, where x_i and y_i depicts the position of the center of the vehicle rear axle. Suppose T_p is the whole planning time horizon, t is the current time step. Typically, the planner optimizes the motion within the whole time horizon $q_{t+1:t+T_p} = \left[q_{t+1}^T, q_{t+2}^T, \ldots, q_{t+T_p}^T\right]^T$.

Note that the position vector q_t is not the full state of the vehicle at time step t. The full state s_t of the vehicle should contain at least two more position vectors backward, that is $s_t = \left[q_{t-2}^T, q_{t-1}^T, q_t^T\right]^T$. Then velocity and acceleration can be obtained via backward differences. For the optimization at each time step t, s_t is used as the initial value so that the velocity, acceleration, yaw angle, as well as jerk and yaw rate at $t+1$ can be calculated.

The objectives contain five aspects to enhance driving quality and follow traffic rules. Taking into account factors such as comfort, smoothness and fuel consumption, we penalize accelerations, jerks, and yaw rates in J_1, J_2, and J_3, respectively. The expressions can be found in [145]. Position errors relative to the desirable traffic-free reference path are penalized as

$$J_4\left[q_{t+1:t+T_p}\right] = \sum_{i=t+1}^{t+T_p} d\left(q_i\right)^2,$$

where $d\left(q_i\right)$ is the distance from the position at the ith time step to the desirable traffic-free reference path. In order to enhance time efficiency and avoid overspeed, velocity errors relative to the desirable traffic-free reference velocity are penalized as

$$J_5\left[q_{t+1:t+T_p}\right] = \sum_{i=t+1}^{t+T_p} \left\|v_{\text{limit}} V\left(q_i\right) - \dot{q}_i\right\|^2,$$

where $V\left(q_i\right)$ is the unit tangent vector of the reference path for q_i. After defining the five aspects above, we can express the cost function of the optimization as a weighted sum, that is,

$$J\left[q_{t+1:t+T_p}\right] = \sum_{j=1}^{5} w_j J_j\left[q_{t+1:t+T_p}\right].$$

In order to guarantee the feasibility of the planned trajectory according to the vehicle kinematics and dynamics, we constrain curvatures and accelerations of the vehicle. The curvature constraints can be written as

$$|\kappa_i| \leq \kappa_{\max}, \quad i = t+1, ..., t+T_p. \tag{4.1}$$

For dynamics constraints, [145] used a tire friction circle as the acceleration constraints, which is shown in Fig. 4.3(a). Such constraints miss a key element of vehicle dynamics, which is the limitation of engine traction when accelerating the vehicle. In fact, maximum traction acceleration $a_{\max}^{+}$ is much smaller than the absolute value of maximum brake deceleration $a_{\max}^{-}$ typically. Therefore, a better approximation of the vehicle dynamics constraints is proposed in this chapter, which is shown in Fig. 4.3(b). In the forward-rearward direction, the maximum acceleration is $a_{\max}^{+}$, and the maximum deceleration is $a_{\max}^{-}$. Therefore, the radius of the acceleration circle is $r_a = \left(a_{\max}^{+} + a_{\max}^{-}\right)/2$, and the distance from the origin to the center of the circle is $c_a = \left(a_{\max}^{-} - a_{\max}^{+}\right)/2$. Then the acceleration constraint circles can be written as

$$\left\|\ddot{q}_i + c_a V\left(q_i\right)\right\|^2 \leq r_a^2, \quad i = t+1, ..., t+T_p. \tag{4.2}$$

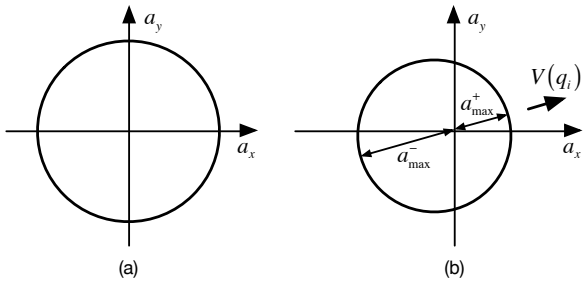


Figure 4.3: Acceleration circles

Collision avoidance for moving and static obstacles is another aspect of hard constraints to consider in order to guarantee driving safety and traffic rule adherence. We will not dive into the details since it is not the core scope of this chapter. Instead, we will only give a brief description here. We use the bounds of moving obstacles obtained in the Section II, as well as information on road structure and static obstacles to create lines for collision avoidance checking. A vehicle body can be represented into several circles, and the created lines are used to bound the centers of the circles. Therefore, the constraints can be expressed as

$$g_k\left[c_j\left(q_i\right)\right] \leq 0, \quad i = t+1, ..., t+T_p, j = 1, ..., n_{\text{cir}}, \tag{4.3}$$

where $c_j\left(q_i\right) = \left[x_{c_j(q_i)} \; y_{c_j(q_i)}\right]^T$ is the center of the jth circle at the ith time step, and $g_k\left[c\right]$ is a linear combination of x_c and y_c with bias representing the kth line created.

After defining all constraints and objectives, the planning problem in deterministic environments can be written as

$$\begin{aligned} \min_{q} \quad & J\left[q_{t+1:t+T_p}\right], \\ \text{s.t.} \quad & \text{Constraints (4.1)-(4.3).} \end{aligned}$$

Safety assessment under uncertainty

Safety is the top priority in autonomous driving, which should be guaranteed. However, conservative definitions of safety may lead to overcautious behaviors. In this chapter, we exploit the feasibility of vehicle kinematics and dynamics to extend the safe set of an autonomous vehicle, so that NCDS can be achieved.

The definition of safety should contain current safety and preview safety. Current safety means that the current state is collision-free. Preview safety means that given the current state as the initial state, a sequence of motion exists, which is feasible and collision-free for a specific length of preview horizon T.

Suppose s_t is the current state of the autonomous vehicle, and $E \in \{\text{pass, yield}\}$ represents the case. We first define current safe set under each case, that is,

$$\mathcal{S}_E^{\text{curr}} = \{s_t |\ q_t \text{ satisfies } (4.3)_E\},$$

where Constraints $(4.3)_E$ are the collision avoidance constraints constructed under the passing or yielding case. Next the preview safe set under each case is defined in terms of preview horizon T, that is,

$$\mathcal{S}_E^{\text{prev}}(T) = \{s_t | \exists\ q_{t+1:t+T} \text{ s.t. } (4.1), (4.2), \text{ and } (4.3)_E\}.$$

Since the collision avoidance constraints do not need to be considered when the probability of the case goes to zero, the overall safe set in terms of T under each case can be written as

$$\mathcal{S}_E(T) = \begin{cases} \mathcal{S}_E^{\text{curr}} \cap \mathcal{S}_E^{\text{prev}}(T), & P(E) > 0 \\ \mathcal{U}, & P(E) = 0 \end{cases}$$

where $\mathcal{U}$ is the whole state space.

Finally, the overall safe set (zone) of the autonomous vehicle in terms of T can be defined as

$$\mathcal{S}(T) = \mathcal{S}_{\text{pass}}(T) \cap \mathcal{S}_{\text{yield}}(T).$$

Then $s_t \in \mathcal{S}(T)$ can be interpreted as that safety is guaranteed within the preview horizon T with state s_t.

Planning framework under uncertainty

Next we will illustrate how the probability of each case, namely $P\left(\text{pass}\right)$ and $P\left(\text{yield}\right)$, can be utilized in a unified trajectory planning framework under uncertainty so that NCDS can be achieved. In the receding horizon optimization framework, although the vehicle trajectory is planned in a relatively long preview horizon, the autonomous vehicle only executes the first (few) motion(s) planned. To be ready to deal with varying cases in the future, different long-term motions should be planned, but the short-term motion should be consistent for different future cases since the motion executed at the next time step should be determined.

Therefore, the following position vector is created, which contains short-term motion with horizon T_1, as well as long-term motions for each case with preview horizon T_p. The vector can be expressed as

$$q = \left[q_{t+1:t+T_1}^T, \left(q_{t+T_1+1:t+T_p}^{\text{pass}} \right)^T, \left(q_{t+T_1+1:t+T_p}^{\text{yield}} \right)^T \right]^T,$$

which is the position vector to be optimized in our planning framework under uncertainty. Then for each case the position vector for the whole preview horizon is

$$q_{t+1:t+T_p}^E = \left[q_{t+1:t+T_1}^T, \left(q_{t+T_1+1:t+T_p}^E \right)^T \right]^T,$$

in which $E \in \{\text{pass, yield}\}$.

Then the optimization problem can be formulated as

$$\begin{aligned} \min_{q} \quad & \sum_{E \in \{\text{pass, yield}\}} P\left(E\right) J\left[q_{t+1:t+T_p}^E \right] \\ \text{s.t.} \quad & \text{(4.1) and (4.2) for } q_{t+1:t+T_p}^E,\ \forall E \in \{\text{pass, yield}\}, \\ & (4.3)_E \text{ for } q_{t+1:t+T_p}^E,\ \forall E \in \{\text{pass, yield}\}. \end{aligned}$$

to minimize the expected cost. The position vector for each case should satisfy feasibility constraints, as well as the collision avoidance constraints for each case, respectively.

It can be easily proved that with preview horizon T_p, by executing the next position vector q_{t+1} obtained, the state of next time step is in the safe set with $T_p - 1$ preview horizon, that is,

$$s_{t+1} \in \mathcal{S}\left(T_p - 1\right).$$

Therefore, the driving strategy based on the proposed planning framework is defensive if potential threats exist.

Moreover, the driving strategy will not overreact to potential threats with low probability since the cost for yielding case $J\left[q_{t+1:t+T_p}^{\text{yield}} \right]$ only minimally influence the total cost. In fact, the long-term motions under each case are voting as part of the cost function to decide the

short-term motion to execute at the next time step. The voting outcome depends on the probability of each case. Decision-making is incorporated in the planning framework, and the final decision is not made immediately until it needs to be. This is the reason why the strategy is not conservative.

4.4 Illustrative Examples

In this section, examples are shown to illustrate the capability of the proposed planning framework to achieve NCDS. We used the two-way-stop scenario in Fig. 4.1 to show how the probabilistic threats were handled. The logistic regression model described in Section II B and trained by the empirical data was used to obtain $P\,(\text{pass})$ and $P\,(\text{yield})$, which was updated at every time step. We also created static obstacles invading the travel lane of the host vehicle to test the collision avoidance capability and smoothness with lateral motions.

The sampling time of the receding horizon optimization is $h = 0.25$ s. The horizon of planning at each time step is $T_p = 4$ s. The short-term horizon is $T_1 = 0.5$ s. Rectangle was used to represent the vehicle body. $v_{\text{limit}} = 10$ m/s, $\kappa_{\max} = 0.2$ m^{-1}, $a^{+}_{\max} = 4$ m/s^2 and $a^{-}_{\max} = 8$ m/s^2.

First a sequence of violation motions in our dataset was used to test the defensive capability of the planning framework and corresponding results are shown in Fig. 4.4. When the violating vehicle was relatively far away from the stop bar, $P\,(\text{pass})$ was still as high as 0.9806. However, as the distance becomes smaller but the velocity was still high and there was hardly any deceleration, $P\,(\text{pass})$ went down rapidly at each sample point as

$$[0.8496, 0.3801, 0.0594, 0.0049, 0.0003, 0, 0, ...]\,.$$

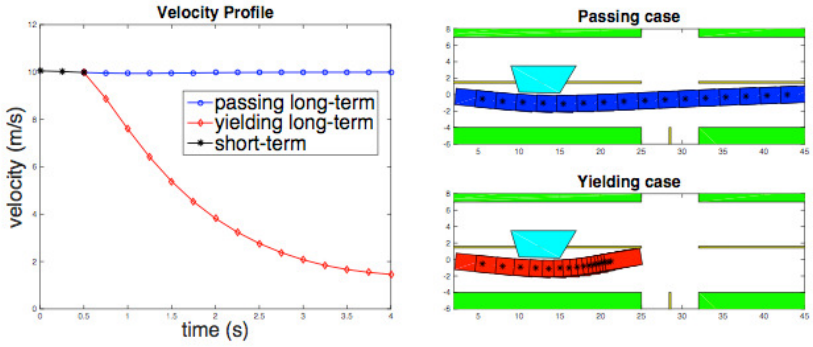
The planned motions and velocity profiles at the first four time steps are shown in Fig. 4.4 with corresponding timestamps and probabilities. The final executed motions and velocity profiles are shown in Fig. 4.6(a).

The results proved several aspects of capabilities of the planning framework. First, the trajectory was very smooth even with lateral motions, and the speed did not exceed the speed limit. Also, collision avoidance and feasibility were guaranteed within the preview horizon. Moreover, the vehicle tried to keep at the center of its lane.

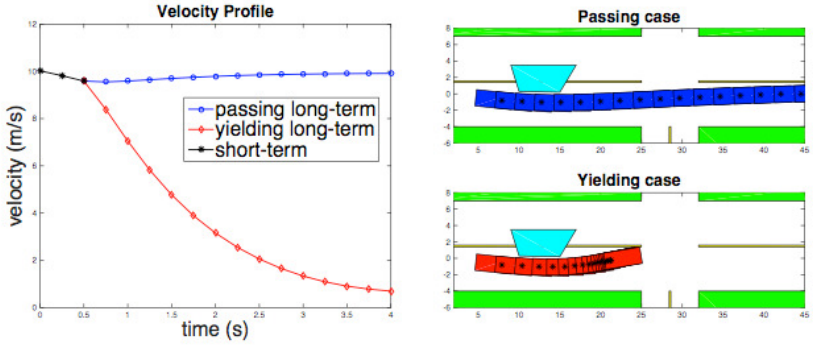
By comparing the planning results with different probabilities, we can explain the principle of our planning framework intuitively. In the velocity profiles, the black lines are the short-term motions which will be executed for the next time step. The blue line represents the long-term motion under the passing case and the red line corresponds to the yielding case. When $P\,(\text{pass})$ is close to 1, the planner tends to keep the current speed which increases the cost under the yielding case. When $P\,(\text{pass})$ becomes smaller, the planner tends to slow down and the deceleration is higher which increases the cost under the passing case.

Next a sequence of stop motions in our dataset was used to test the non-conservative capability of the planning framework. The probabilities at sample points were

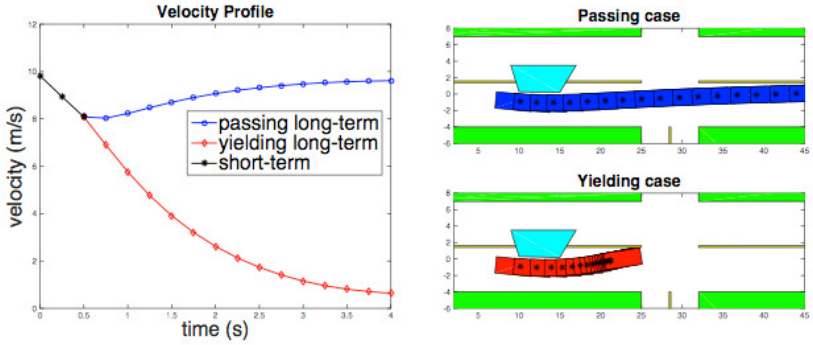
$$[0.9696, 0.9992, 0.9999, 0.9998, 1, 1, ...]\,.$$



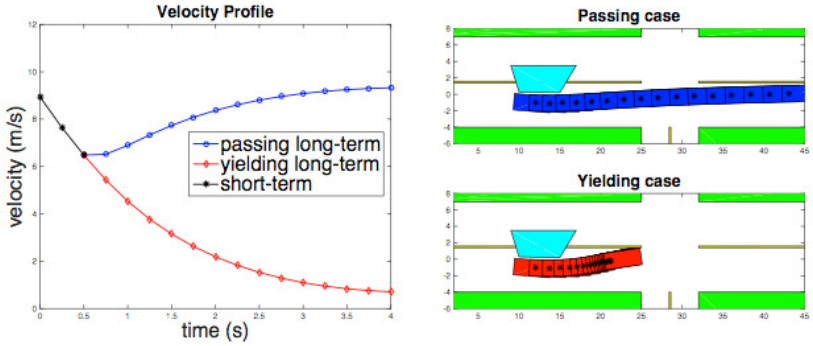
(a) t=0, P(pass)=0.9806



(b) t=0.25 s, P(pass)=0.8496



(c) t=0.5 s, P(pass)=0.3801



(d) t=0.75 s, P(pass)=0.0594

Figure 4.4: Planned motions and velocity profiles at the first four time steps under yield case

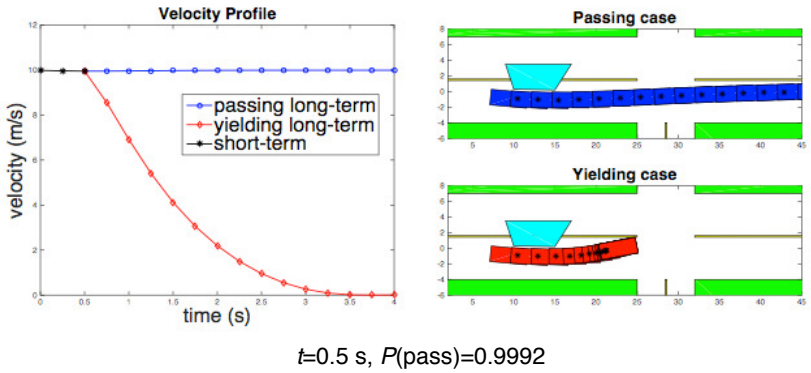


Figure 4.5: Planned motions and velocity profiles at one step under pass case

When the vehicle was relatively far away from the stop bar, the high speed made P (pass) a little smaller than 1, which imitated the threat with low probability. As it started to decelerate, the probability went to 1. The planned motions and velocity profiles at one step are shown in Fig. 4.5. with corresponding timestamp and probability. The final executed motions and velocity profiles are shown in Fig. 4.6(b). In the results we can see that the potential threat with very low probabilities does not influence the speed of the vehicle meaningfully, which makes the strategy non-conservative. However, safety is always guaranteed since the long-term motion under yielding case ensures safety at each time step.

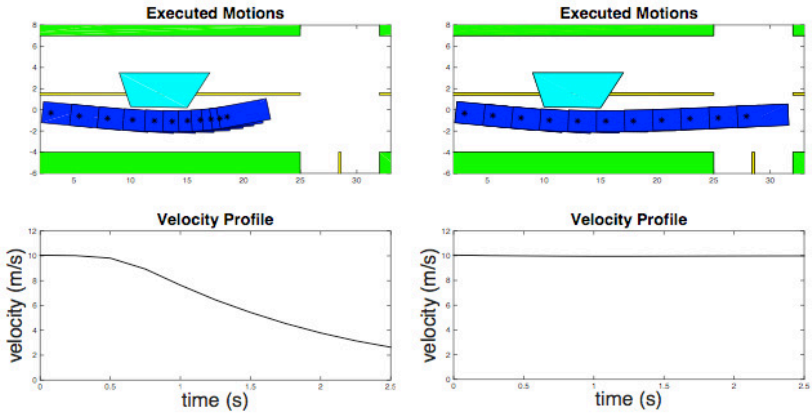


Figure 4.6: Executed motions under each case

4.5 Chapter Summary

A unified planning framework under uncertainty was proposed in this chapter for urban autonomous driving, which can achieve NCDS for various kinds of scenarios. Based on the deterministic planner designed, as well as the probability under each possible case obtained by behavioral modelling, trajectories were planned to avoid overcautious behavior and to guarantee safety via defensive behavior. Two-way-stop intersection was used as an exemplar scenario to show the capabilities of the proposed planning framework. The results demonstrated that based on the proposed planner, the autonomous vehicle can guarantee safety even when others are violating traffic rules, and the host vehicle does not overreact to threats with low probabilities. For future studies, various urban driving scenarios will be used to test the capabilities of the proposed planning framework to achieve a driving strategy which is defensive, but not overly conservative.

Chapter 5

NCDS Intuitions, Extensions and Experiments

5.1 Integrated Decision and Planning Framework

The proposed non-conservatively defensive strategy (NCDS) is an integrated decision and planning framework under uncertainty, instead of a hierarchical architecture making a definite decision first then planning the motion accordingly. As shown in Fig. 5.1, NCDS combines a decision network and a motion planner with hard constraints.

A decision network maximizes the expected utility over the decision a, that is,

$$\max_{a} \sum_{E} P(E|z)U[a,E],$$

where z is the observed feature of the environment, E is the future case, and $U[a,E]$ is the utility function. The conditional probability of a future case given the observed feature, $P(E|z)$, can be obtained from the prediction or perception models.

An arbitrary motion planner with cost function $J[q]$ and feasibility constraints $f(q) \leq 0$ and safety constraints $g(q) \leq 0$, that is,

$$\begin{aligned} \min_{q} \quad & J\left[q\right], \\ \text{s.t.} \quad & f(q) \leq 0 \text{ (feasibility)} \\ & g(q) \leq 0 \text{ (safety).} \end{aligned}$$

can be incorporated into the framework with the decision network. We provided an example of optimization-based motion planner in Chapter 4 and its integration with a decision network. However, such framework can incorporate different motion planners such as search-based or sample-based planners. In this chapter, experimental results are provided by combining decision networks and search-based or sample-based planners.

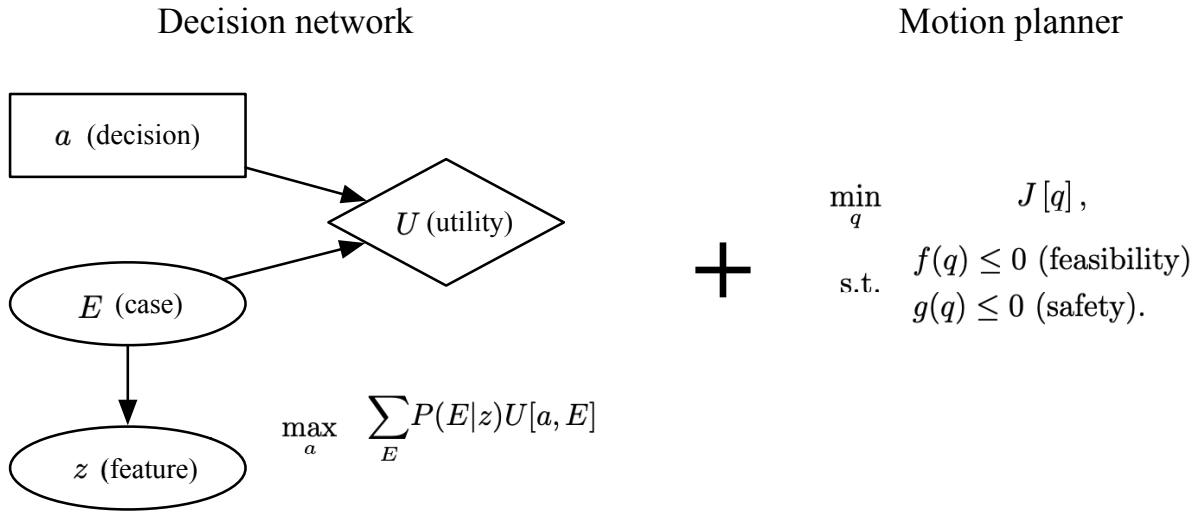


Figure 5.1: An integrated decision and planning framework under uncertainty.

Moreover, a more general formulation of the framework is provided to illustrate the geometrical intuition. Suppose q_s denotes the short-term planned motions for the autonomous vehicle, while q_{ly} and q_{lp} correspond to the long-term planned motions for "yield" and "pass" cases, respectively. Then the optimization problem can be formulated as

$$\begin{aligned} \min_{q_{\text{aug}}} \quad & P\left(\text{pass}|z\right) J\left[q_{\text{pass}}\right] + P\left(\text{yield}|z\right) J\left[q_{\text{yield}}\right] \\ \text{s.t.} \quad & f(q_{\text{pass}}) \leq 0 \\ & f(q_{\text{yield}}) \leq 0 \\ & g_{\text{pass}}(q_{\text{pass}}) \leq 0, \text{ if } P\left(\text{pass}|z\right) > \epsilon \\ & g_{\text{yield}}(q_{\text{yield}}) \leq 0, \text{ if } P\left(\text{yield}|z\right) > \epsilon, \end{aligned}$$

where

$$q_{\text{aug}} = \left[q_s^T, q_{ly}^T, q_{lp}^T\right]^T$$

is the augmented decision variable, and

$$\begin{aligned} q_{\text{pass}} &= \left[q_s^T, q_{lp}^T\right]^T, \\ q_{\text{yield}} &= \left[q_s^T, q_{ly}^T\right]^T. \end{aligned}$$

5.2 Geometrical Intuition

The cost function J can be non-convex, and the inequality constraints $f(q) \leq 0$ and $g(q) \leq 0$ can be highly nonlinear and non-convex according to our design in Chapter 4 to

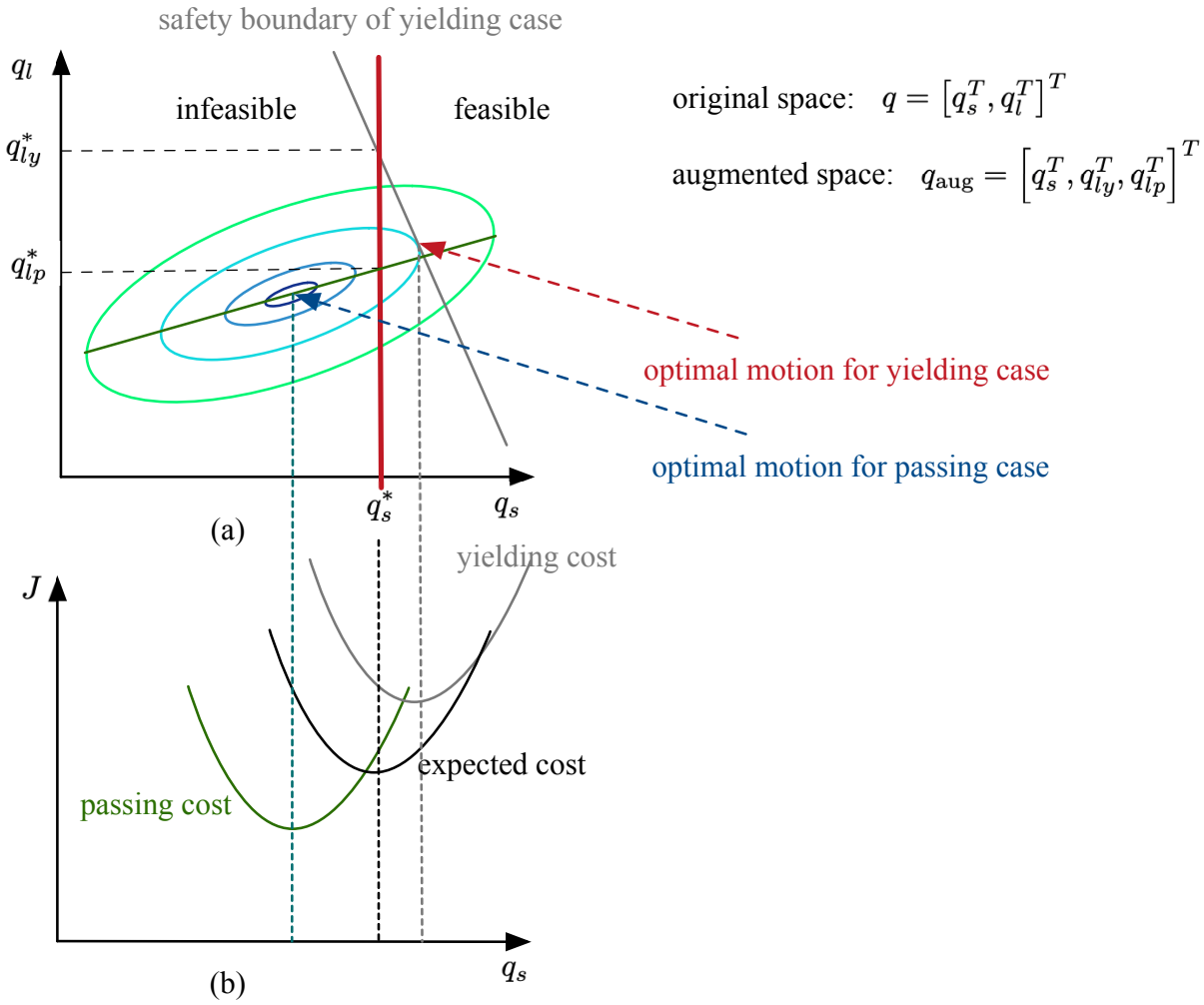


Figure 5.2: Geometrical intuition for NCDS with a unified and convex cost function.

take into account practical factors. However, we would like to provide intuitive explanations from geometrical perspectives on how the probabilities obtained from the perception or prediction modules can reshape the cost function and the short term planned motions to be executed for the next few time steps. Therefore, we simplify the cost function to be convex and the constraints to be linear. Also, the subspaces for q_s, q_{lp} and q_{ly} are one-dimensional so that they can be easily visualized.

In Fig. 5.2 (a), the cost $J[q]$ in (q_s, q_l) space is visualized with contours. The grey line represents the safety boundary for the yielding case, and the feasible set is on the right-hand side. We simplify the cost function to be quadratic and the boundary to be linear for easier representation.

If NCDS is not introduced, the potential threat (infeasible set) will be considered as a deterministic infeasible set. The deterministic motion planner will directly find the local optima as the optimal motion for yielding case in Fig. 5.2 (a), which is a relatively conservative option (slow down immediately).

On the other hand, if the integrated decision and planning framework with NCDS is introduced, an augmented space (q_s, q_{ly}, q_{lp}) is constructed. We assume that both of the subspaces (q_s, q_{ly}) and (q_s, q_{lp}) share the same cost function, which is compatible with what we have illustrated in Chapter 4. The safety boundary is active only for subspace (q_s, q_{ly}), and in subspace (q_s, q_{lp}) there is no such threat.

Then two cost functions can be obtained for cost $J[q_s]$ as shown in Fig. 5.2 (b). We can project the cross section of the cost and safety boundary plane onto the (J, q_s) plane, so that a yielding cost (grey curve) can be formed, as shown in Fig. 5.2 (b). It represents the optimal cost obtained given a specific q_s in the feasible set. Then without considering the safety boundary for the passing case, we can obtain the passing cost (green curve) in Fig. 5.2 (b), which is the optimal cost obtained given a specific q_s.

Finally, the probabilities from the prediction or perception modules are utilized to reshape the cost function into an expected cost (black curve) in Fig. 5.2 (b), which is a weighted sum of the passing and yielding cost. When the probability (weight) of the yielding case is much higher than that of the passing case, the optima in terms of the expected cost will be very close to the optima in terms of the yielding cost, which is relatively conservative. On the other hand, if the risk (probability/weight of the yielding case) is extremely low, the short-term planned motions will be very similar to the planned motions of the passing case to avoid over-cautious behavior. Most importantly, the obtained q_s^* will always guarantee that we can find a q_{ly}^* within the feasible set so that collision can be avoided.

Similar intuition can also be achieve for non-unified and locally convex cost with non-convex feasible set as shown in Fig. 5.3. We also provided several toy examples on how two quadratic functions are reshaped by probabilities (weights) to form an overall (expected) cost in Fig. 5.4.

5.3 Experimental Setup

Vehicle platform

We utilized the test vehicle of Berkeley DeepDrive to conduct experiments. The test vehicle is shown in Fig. 5.5. We implemented an extended Kalman filter to estimate the states of the vehicle by fusing the information from DGPS and IMU. A model predictive controller was designed for the lateral control, and we used the PID controller integrated in the vehicle system for longitudinal control. Then the decision and planning framework we designed can be implemented on the vehicle platform.

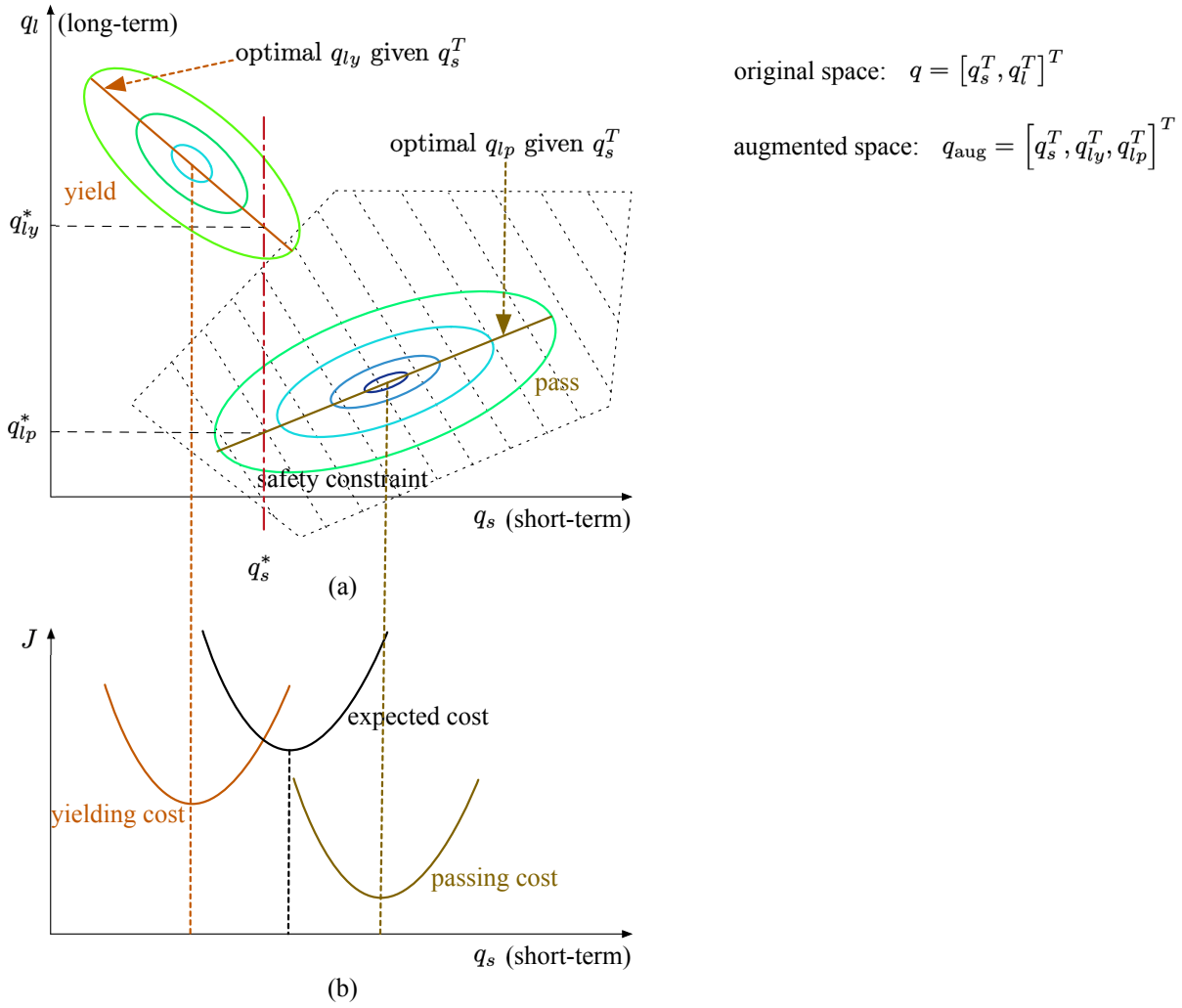


Figure 5.3: Geometrical intuition for NCDS with separated and locally convex cost function.

Predictor

We utilized the predictor proposed in [109]. It is a prediction framework based on dynamic Bayesian network incorporating vehicle kinematics and driver model without relying on data. The model is relatively stable and highly generalizable in different scenarios.

Experiment field and scenario

We used the test field in Richmond Field Station for our experiments. The vehicle ignored the road structure and objects in the real test field and was placed on a virtual map from

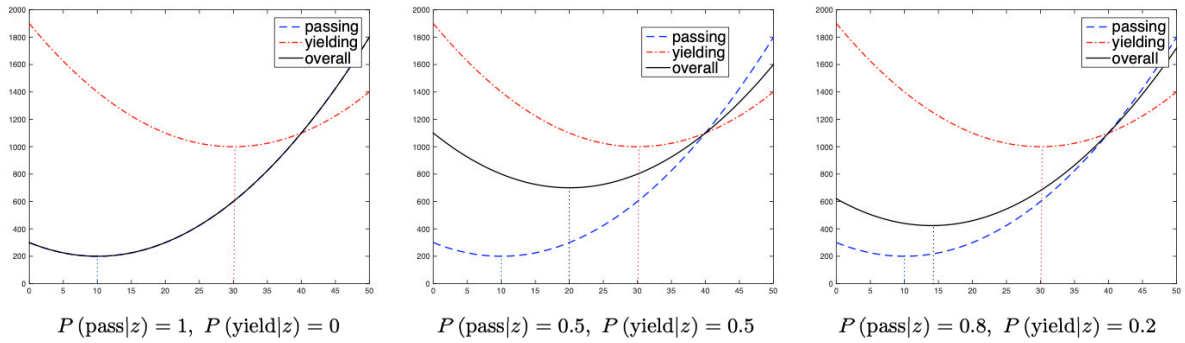


Figure 5.4: How probabilities reshape the cost functions.

Figure 5.5: Test vehicle.

real-world scenario, which is from the INTERACTION dataset [137]. The scenario is shown in Fig. 5.6. It is a roundabout with several entrance. The test vehicle came from the very left branch in the figure, and interact with a vehicle in the roundabout holding the right-of-way.

5.4 Extension to Graph-Search-Based and Sample-based Planners

As stated in Section 5.1, the proposed NCDS can take into account an arbitrary motion planner with cost and constraints. Optimization-based planner used in Chapter 4 is not

Figure 5.6: Real-world roundabout scenario for experimental test.

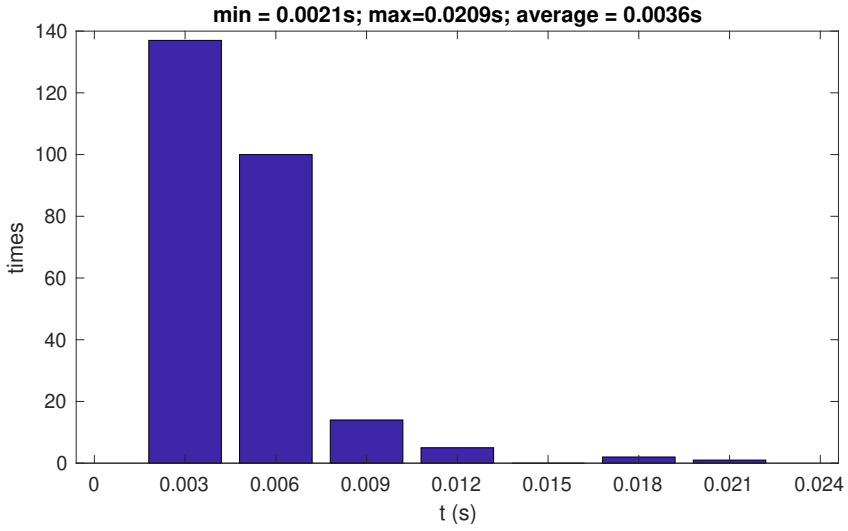


Figure 5.7: Computation time statistics of the NCDS with graph-search-based planner.

a desirable solution to achieve real-time computation for autonomous vehicles. Therefore, we combined a graph-search-based planner proposed in Chapter 2, as well the sample-based planner in [43] with decision network to achieve NCDS in real time.

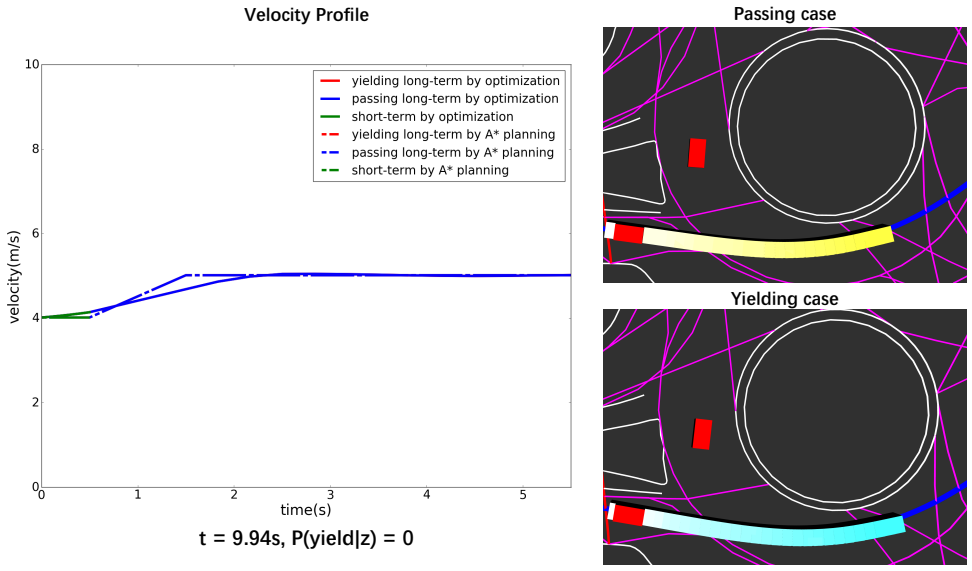


Figure 5.8: Planned motions at one time step when the probability of the case for the autonomous vehicle to yield is 0.

Graph-search-based planner

The planning horizon was 5 s with passing and yielding cases considered. The statistics of computation time is shown in Fig. 5.7. The upper and lower bound of the computation time were 20.9 and 2.1 ms, while the average was 3.6 ms. The proposed integrated decision and planning framework with the A* search planner is highly efficient for online computation.

The planned motions with corresponding probabilities can be found in Fig. 5.8-5.11. In Fig. 5.8 and 5.9, we can see that when the predictor provided deterministic prediction results for future cases, the planned motions were just the same as a deterministic planner designed in Chapter 2.

When the probability was larger than our risk tolerance, but smaller than 1, as shown in Fig. 5.10 and 5.11, there were one short-term trajectory with two long-term trajectories planned. By minimizing the expected cost weighted by the corresponding probabilities, the desirable short-term motion tended to form a long-term motion which is similar to the optimal long-term motion of the case with higher probabilities. When the probability of the yield case is relatively high in Fig. 5.10, the desirable short-term motion with the yielding long-term motion formed a smooth speed profile. When the probability gets lower in Fig. 5.11, the corresponding speed profile became relatively jerky.

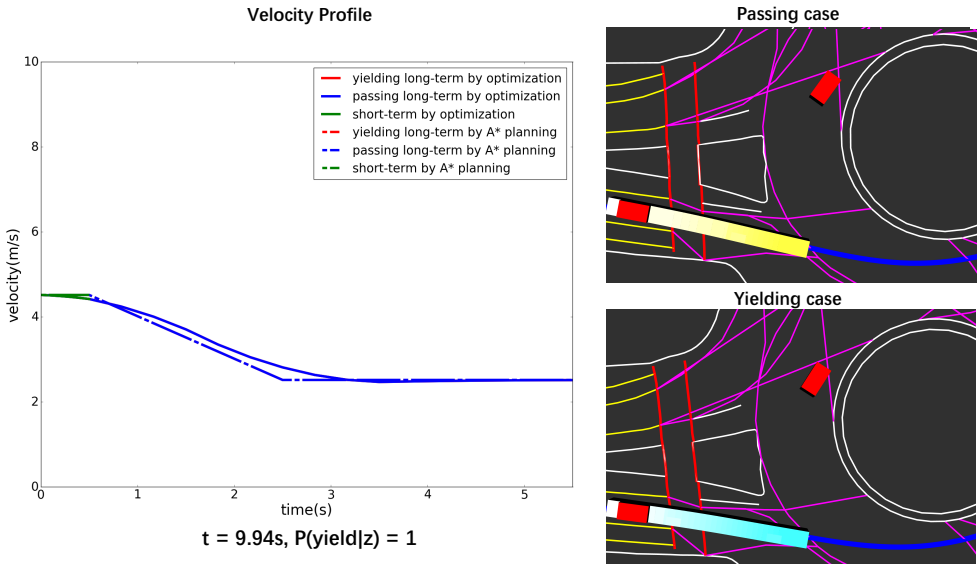


Figure 5.9: Planned motions at one time step when the probability of the case for the autonomous vehicle to yield is 1.

Sample-based planner

The trajectory candidates of a sample-based planner with NCDS is provided in Fig. 5.12. The best long-term candidates for pass and yield cases are selected and the expected cost of the short-term motion is then calculated. Finally, the short-term motion with the most desirable expected cost is selected as the one to be executed for the next time step. The planned motions at one time step with the sample-based planner can be found in Fig. 5.13. The desirable short-term motion was relatively smooth when connecting with the long-term motion of the yielding case since $P(\text{yield}|z)$ was relatively high.

5.5 Chapter Summary

In this Chapter, the intuition of the integrated decision and planning framework was provided. It was combined with graph-search-based and sample-based planners, and implemented on a real vehicle with real-world scenarios and motions of other vehicles with non-conservatively defensive strategy achieved. Highly efficient computation time was achieved based these methods.

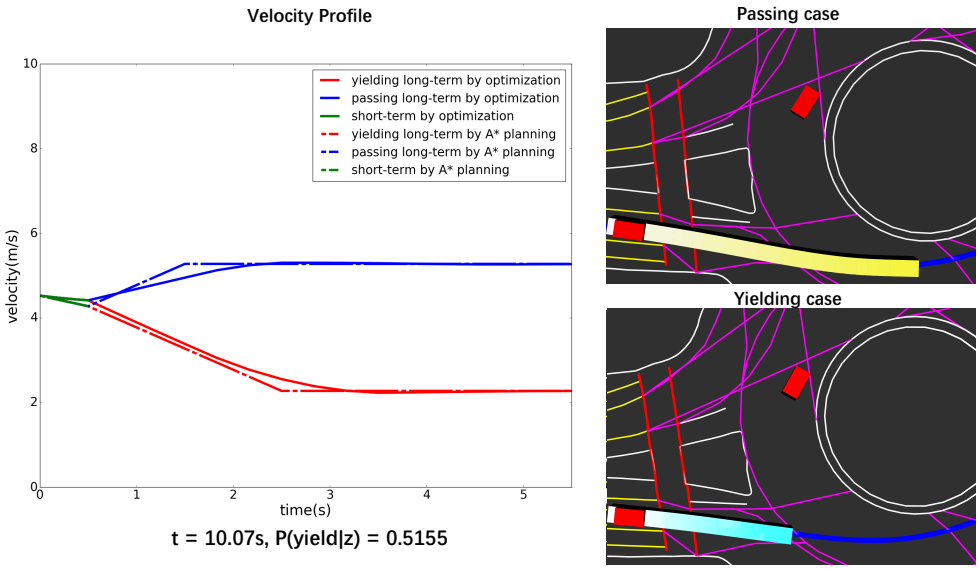


Figure 5.10: Planned motions at one time step when the probability of the case for the autonomous vehicle to yield is 0.5155.

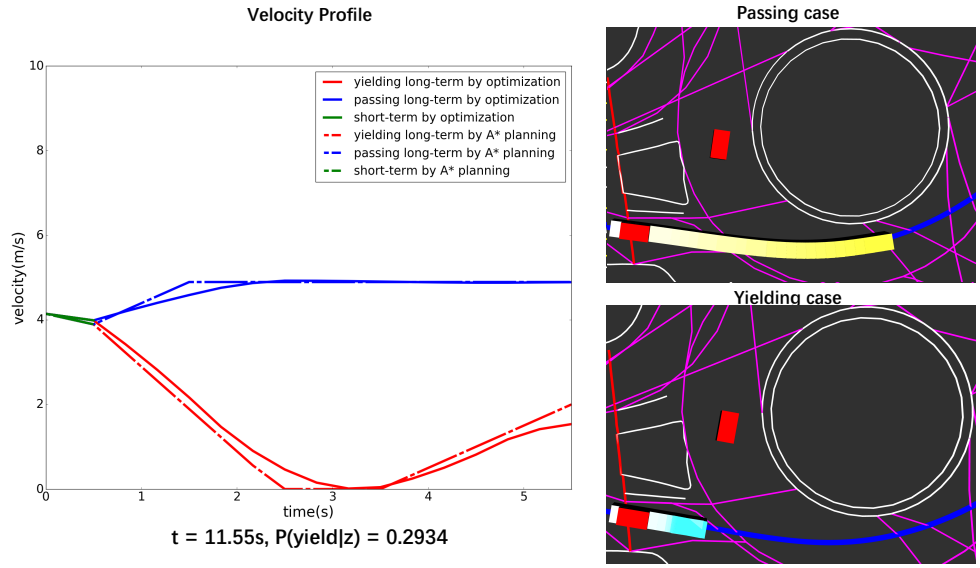


Figure 5.11: Planned motions at one time step when the probability of the case for the autonomous vehicle to yield is 0.2934.

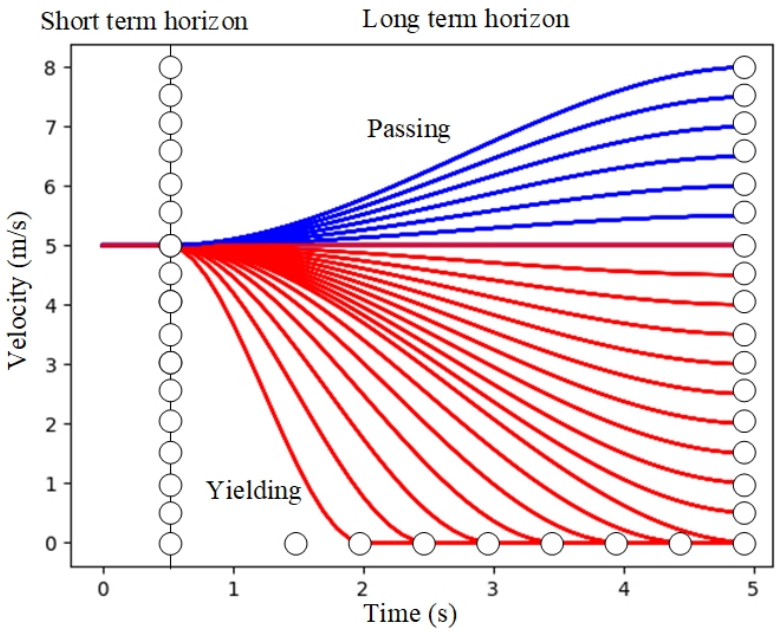


Figure 5.12: Trajectory candidates of a sample-based planner with NCDS.

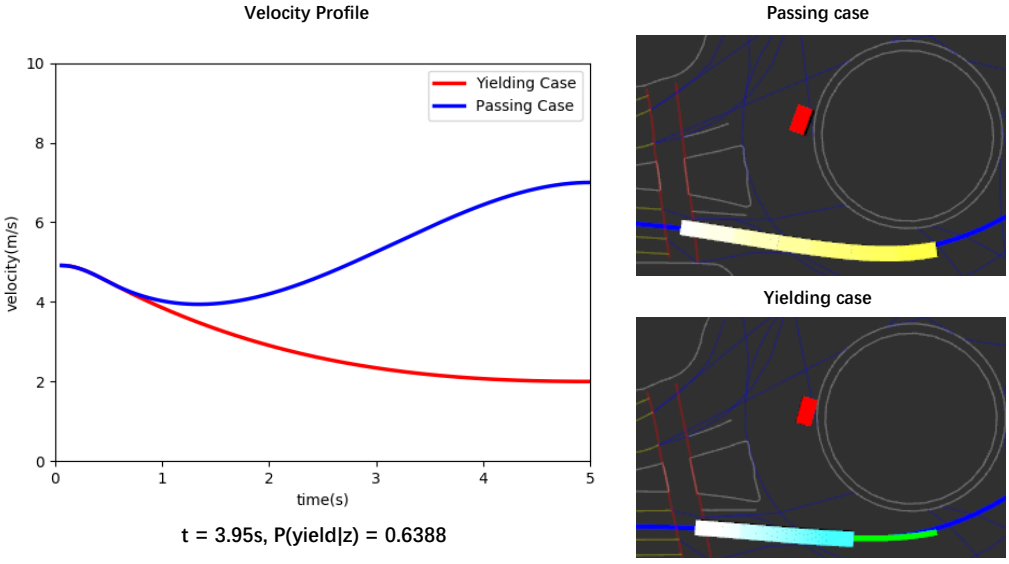


Figure 5.13: Planned motions at one time step with a sample-based planner.

Part III

Interactive Prediction and Planning

Chapter 6

Formulation and Representation for Prediction

6.1 Introduction

The behavior of traffic participants is full of uncertainties in the real world. Autonomous vehicles need to well estimate such uncertainties to increase driving quality (time-efficiency, comfort, etc.) and safety level for the decision-making and motion planning. To drive safely, autonomous vehicles should predict possible intentions and motions of other road participants, and avoid collisions accordingly. To enhance driving quality, autonomous vehicles should take threats of high probability seriously, yet not overreact to threats of low probability. Therefore, probabilistic intention and motion predictions are inevitable for safe and high-quality decision-making and motion planning for autonomous vehicles.

Most of the research efforts on probabilistic prediction [71] were focused on prediction algorithm design for specific scenarios. Some recent works addressed incorporating prior knowledge to construct prediction frameworks which can deal with a variety of scenarios [108][36]. Solutions for problem formulation and motion representation simplification were often arbitrarily adopted, which lacks sufficient investigation and solid foundation.

Problem formulation

The variation of the assumptions and settings for the input and output of the prediction algorithm can significantly change the forms of the input and output of prediction algorithms. It is hard to compare algorithms with different problem formulations due to the viewpoint and interaction-involvement of the input, as well as the number of predicted entities and motion representation of the output. Such variation can also completely change the problem complexity, as well as model practicability for decision-making and planning of the host vehicle.

Many probabilistic prediction methods have been proposed based on neural networks (NN) [93][86][51], as well as probabilistic graphical models (PGM) [28] such as particle filter

[47][73] and Bayes net [108]. All the aforementioned literatures formulated the prediction problem to predict the distribution of future motions of an entity given the historical motions of relevant participants in the scene. However, solving such a problem cannot provide sufficient and accurate predictions for highly interactive driving scenarios. When several entities are closely interacting with each other in a specific scenario, the future motion of the host autonomous vehicle can significantly impact the motion of its surrounding entities. Therefore, like human drivers, autonomous vehicles should always ask "what if I take this action" during interactions. Recently, prediction methods based on inverse reinforcement learning (IRL) [105] were proposed to tackle reaction prediction problems. However, only optimal motions were provided in IRL, which makes it deterministic. Therefore, the main stream paradigms for prediction, such as NN, PGM and IRL, should be modified to solve probabilistic reaction prediction problems for highly interactive driving scenarios. For methods except for IRL, approximating the (situational) joint distribution [74][50] [63][70] of the motions of several entities is typically much easier than approximating the reaction distribution. Therefore, we should also provide the transformation between situation and reaction predictions.

Motion representation simplification

Indicators are commonly used to simplify the motion representation to enable the probabilistic description of the predictions since the original space of trajectories can be extremely high with continuous variables. The representation of future motions should be homogenized so that same indicators are used for comparing the predicted distribution and the ground truth to make the evaluation metric meaningful. Motion patterns such as route [62] and pass-yield [28] patterns, as well as spatiotemporal representations such as prototype trajectory [67] and reachable set [6], were used as indicators, which need to be summarized.

Intention is one of the most common indicators for simplification. However, the meaning of "intention" can be twofold, namely, original desire and executed motion pattern. In highly interactive scenarios, an entity may not be able to achieve its original desire due to the motion of others. In fact, the decision and planning module checks the potential collision with others according to their possible motions to execute, but not the original desire. Moreover, the original desire cannot be labeled as the ground truth in datasets. Instead, only the executed motion patterns can be labeled. Therefore, the difference between pattern and desire should be clarified.

6.2 Survey on Problem Formulation

Suppose x_i is an input observation in Figure **??**, which contains the extracted features from historical motions in the dataset with map context. $\hat{y}_i$ denotes a representation of the future motion of the predicted entity in the prediction algorithm corresponding to x_i. y_i is the ground truth of the future motion from dataset homogenized to the same form of $\hat{y}_i$. X

and Y are the corresponding random variables. Then the problem is to design prediction algorithms so that the predicted distribution $p(Y|X)$ can best approximate the distribution in the dataset $q(Y|X)$. Note that typically $q(Y|X)$ can only be obtained or represented by ground truth data points (x_i, y_i). There are several variations of the original problem by changing the input and output of the model.

Input variation 1: viewpoint

An important variation of the observation input is the viewpoint. The main distinction for different viewpoints lies in occlusions, such as whether the surroundings of the predicted entity is occluded to the prediction module, and whether the predicted entity itself is occluded.

Bird's-eye view (host prediction)

The viewpoint is a *bird's-eye view* when the surroundings of a predicted entity are fully observable. An algorithm designed with such viewpoint is actually predicting the motion of a host vehicle (first-person viewpoint), since all the adjacent entities can be assumed to be fully observable from the viewpoint of the host vehicle. The most representative bird's-eye view vehicle motion dataset is the NGSIM dataset, which was utilized by many recent works on probabilistic prediction [72][93][28][86].

Local view (surrounding prediction)

The viewpoint is a *local view* when the surroundings of a predicted object are partially observable due to occlusions. An algorithm designed with such viewpoint can be used to predict the motion of a surrounding vehicle (third-person viewpoint). Although the data used in many existing works were collected in local view, probabilistic prediction with partial occlusion has not been sufficiently investigated by researchers yet. It is possible to take into account the uncertainty caused by occlusion via using the raw sensor data as the input [69] or constructing noisy measurement input [73].

Blind view (occluded entity prediction)

The predicted entity can also be partially or fully occluded by vehicles or buildings, which leads to a *blind view*. Algorithms designed with such viewpoint can provide prediction for blind corner decision-making [48][85][133] or occluded object tracking [34].

Input variation 2: interaction

Another input variation is interaction. The main distinction lies in whether the behavior of the predicted vehicle is influenced by the surrounding entities. Another distinction is whether there is a decision-maker asking "what if my motion is like this in the future".

Independent prediction

The "maneuver-based" prediction model was defined in [71] as the behavior of the predicted vehicle is independent from others. Since the word "maneuver" was used in some literatures for behaviors influenced by others, we use *independent prediction* to describe the same problem.

Interdependent prediction

The "interaction-aware" prediction model was defined in [71] when the behavior of the predicted entity is influenced by others. In this chapter we emphasize whether the behavior is impacted by historical or future motions of the surroundings. When only historical surrounding motions are used as the input, the problem can be defined as *interdependent prediction*, namely, the motions of the entities are only historically interactive.

Reactive prediction

From the host vehicle decision-making and planning perspective, the most desirable prediction for a highly interactive scenario is an algorithm that can answer the question "what if my motion is like this in the future". Therefore, we define *reactive prediction* as the problem to obtain the algorithm whose input can take into account the future motion of a host vehicle or a pattern (simplification) of the motion, and provide the distribution of the predicted entity accordingly.

Output variation 1: number of predicted entities

The number of predicted entities as the output can change the formulation of the prediction problem.

Single-entity prediction

Most of the recent works are focused on *single-entity prediction*, where the future behavior of only one entity is provided.

Situation prediction

The combinatorial decision and planning [11] of the host vehicle may strongly depend on the possible behavior of several entities as a group in complicated scenarios. However, it is hard to directly combine the predictions of single entities since the future motions may be exclusive. Instead, situation prediction [63][70] can provide the joint distribution of the motions of surroundings.

Output variation 2: motion representation

The original and intuitive representation of predicted motion is to use continuous trajectories. However, due to the high dimension and difficulty in describing the distribution for continuous random variables, the motion representation is often simplified in order to describe the probability distribution for complicated scenarios. Detailed discussion on representation simplification is provided in Section 6.3.

6.3 Survey on Representation Simplification for Prediction

As discussed in Section 6.2, the representation of long-term motions is usually simplified by indicators since it is intractable to directly use continuous trajectories to describe the distribution of predicted motions in complicated scenarios. In this section, the most commonly used indicators are categorized as continuous motions, motion patterns or spatiotemporal indicators. We also categorize motion patterns according to the hierarchy from decision and planning perspective. Moreover, the spatiotemporal indicators are discussed based on whether motion patterns are considered.

Continuous motions

A sequence of positions and yaw angles [130], as well as velocities [28] and accelerations [86][72] are typically used as the continuous motions. Such representation is relatively more applicable in car following [86][128][123] and ramp merging [72] [28][29] scenarios, where only longitudinal motions need to be considered. When the preview horizon is relatively long, it is intractable for a model to directly output the distribution of long-term motions. Instead, the models can output the distribution for one step look ahead, then structures such as Bayesian filtering [73][130][120] and long short-term memory (LSTM) [86][3] can be employed to propagate the motion to the long-term future.

Motion patterns with hierarchical categorization

For decision-making and planning of the host vehicle, the destination to reach is the first to be considered. A corresponding route can be planned offline, which is typically independent from the situation encountered in real time. Then the local decision-making and planning module can deal with the specific situation, such as whether to pass a conflict region before or after another entity. Such a hierarchical architecture can also be applied for prediction, namely, the route pattern and pass-yield pattern are hierarchical. One example for such hierarchy is predicting potential right turn with pedestrian yielding at an intersection [76], in which route patterns were turn right and go straight, and the pass-yield patterns were go and stop for straight, and turn and yield for turn right. Another example is predicting

routes and right-of-way at a four-way-stop intersection [127]. The route patterns were turn left/right and go straight, and the pass-yield pattern was the right-of-way at intersections.

Moreover, there are also subtle motion patterns which cannot be defined as route or pass-yield patterns, such as slow down, go as expected, accelerate, etc. A comprehensive example for the hierarchy of route, pass-yield and subtle patterns is predicting potential left turn with proceeding and oncoming vehicles at an intersection [64]. The route patterns were go straight and turn left. The pass-yield patterns were go in front and yield for the oncoming vehicle when turning left. The subtle patterns were free drive and influenced for the proceeding vehicle, and full stop and slow down for yielding oncoming vehicle when turning left.

Route pattern

Route pattern (intention) denotes the discrete pattern shaped by the spatial road structure, such as road branches, lanes and parking lots. It depicts which parking lot the entity wants to occupy, or which branch of road it wants to take at nodes such as intersections, roundabouts, exit ramp, etc. Dedicated lanes (left/right turn only) at these nodes are decisive in estimating the intention. Route pattern were estimated for vehicle [93] and cyclists [96] with probabilistic models. Also, probabilistic models for route patterns of vehicles at intersections were also proposed along with other motion patterns [76][64][62][127].

Pass-yield pattern

Pass-yield pattern depicts which entity occupies the conflict region first when the potential routes of two entities have an overlap (conflict). Similar concept was also used in cooperative driving as homotopy class [40]. Such pattern was predicted for ramp merging [28], pedestrian yielding [76], oncoming vehicle yielding for left turn [64], four-way-stop right-of-way [127], gap selection for lane change [51] etc. The patterns such as pass or stop during yellow light, red-light violation or stop [54], and stop or violate the stop sign [26][134] can also be categorized as pass-yield pattern.

Subtle pattern

Subtle pattern denotes the motion patterns which cannot be explicitly categorized via spatial road structure or conflict region for two entities. For example, there can be several motion patterns for vehicles near stop signs such as conservative/normal stop and rolling/moderate/severe violation [26], which can be clustered via unsupervised learning. The longitudinal motions of vehicles can be simplified as acceleration, deceleration and expected behavior pattern [63]. In [64], the motion of the predicted vehicle impacted by the proceeding one was simplified as free drive and influenced, and the left-turn motion yielding the oncoming vehicle was simplified as full stop and slow down.

Spatiotemporal indicators

Motion patterns can be incorporated in the spatiotemporal domain by two kinds of indicators, namely, prototype trajectory and reachable set. The spatiotemporal domain can also be represented without considering any motion patterns and semantic meaning by using occupancy grid.

Prototype trajectory

The terminology "prototype trajectory" was defined in [71] for one or a set of trajectories which can represent a motion pattern. It was employed in [127][55][67] to represent possible motions.

Reachable set

Reachable set is a widely used motion representation describing the area which may be occupied by the body of an entity. Stochastic reachable set [6] can represent the probabilistic prediction and take into account information from motion patterns. It is also more direct to indicate drivable area for the planning module.

Occupancy grid

Occupancy grid divides the spatial domain [60][41], or other domains of motion variables such as velocity [41][62], into discrete grids evenly. It can provide a uniform representation of the environment regardless of the number of entities in the scene, which is favorable for learning models such as deep neural networks [60][41], and for complex scenarios with a number of entities interacting with each other.

Finally, we provide Fig. 6.1 as a summary of the survey on problem formulation and representation simplification.

6.4 Representation Simplification for Prediction

It is intractable to approximate the distribution of long-term trajectories, which are continuous random vectors of motions with high dimensions. Therefore, the representation of future motions is typically simplified by using discrete indicators. In this section, we first clarify the distinction between desires in human mind and executed motion patterns. The clarification explains why "intention" is not suitable to be the simplification indicator. Then we illustrate how to construct the spatiotemporal representation of potential motion patterns.

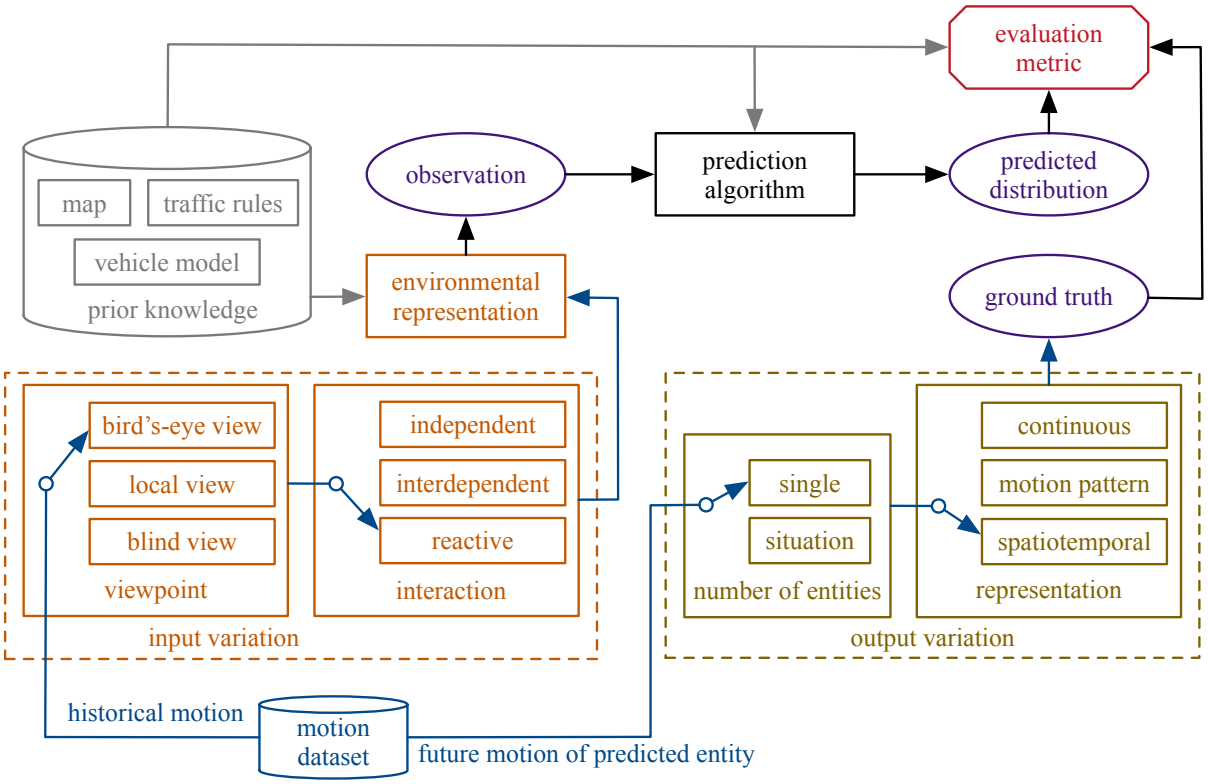


Figure 6.1: Brief summary of the formulation and representation survey.

Clarification of desire and pattern

In literatures, terminologies such as "intention", "maneuver" and "behavior" are often used for motion clusters. However, the meaning of motion clusters can be twofold. One is "motion pattern", which clusters the motions executed by the entities, and it can be observed or labeled in the motion data to serve as the ground truth. The other is "motion desire", which is the motivation inside the mind of humans, and it cannot be fully observed or labeled in the motion data. In fact, the desire in the human minds can rapidly change from time to time with the dynamic behavior of surroundings.

In previous works, using the terminologies such as "intention" is ambiguous to express the distinctions between pattern and desire. It is acceptable to mix pattern and desire for route since the final desire of the entity related to route is typically not influenced by surroundings so that the desire in the data can be time-invariant. However, for pass-yield and subtle patterns, the corresponding desires may jump from one to another rapidly, and the surroundings may restrict the predicted entity to achieve its desire, especially in highly interactive scenarios. Therefore, the ground truth cannot be labeled for such desires in the data, and only motion patterns can be labeled. We will use the following example of ramp

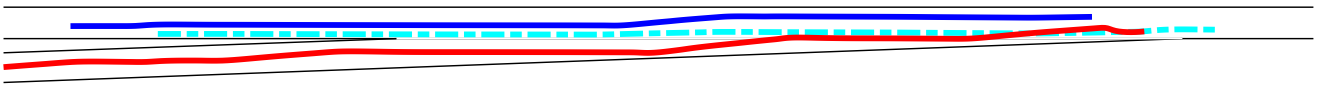


Figure 6.2: Trajectories of the merging (red), target (blue) and front (cyan) vehicles in a ramp merging scenario.

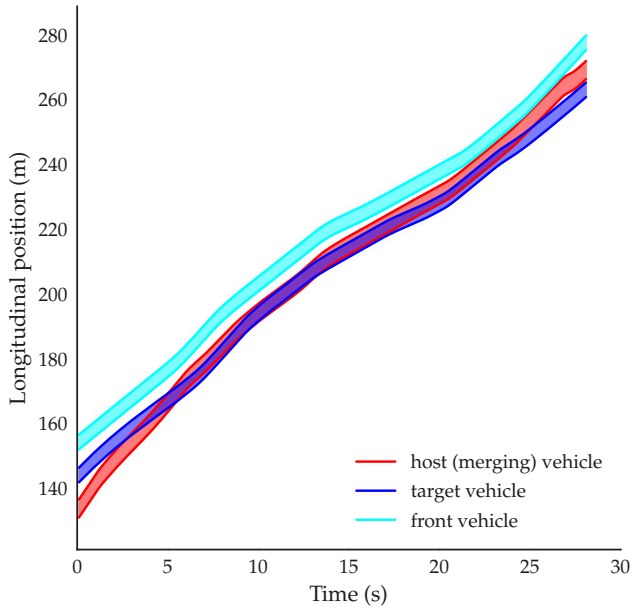


Figure 6.3: longitudinal positions over time of the merging (red), target (blue) and front (cyan) vehicles in a ramp merging scenario.

merging on I-80 in NGSIM dataset [61] to explain.

Figure 6.2 illustrates the trajectories of a ramp merging scene with a preceding and a following target vehicle on the target lane, as well as a merging vehicle on the ramp. The merging vehicle tried to merge into the gap between the preceding and target vehicle, but the target vehicle refused to yield to create a gap. The merging and target vehicles were driven side by side in parallel for a long period. After an adversarial interaction procedure for around 20 seconds, the target vehicle enlarged the gap to enable the merging vehicle to cut in successfully. The procedure in spatiotemporal domain can be found in Figure 6.3 with the longitudinal positions of each vehicle at each time step.

In fact, it is impossible to obtain the ground truth of the desires in the human mind at each time step. For instance, the gap became larger during 5-7 s and went back to a small one during 7-9 s. The desire in the mind of the target vehicle driver on whether to yield

the merging vehicle can change from time to time. Also, it is unclear whether the driver of the merging vehicle hesitated or even decided to give up and merge into the gap behind the target vehicle around 20-22 s since it was approaching the end of the ramp.

In conclusion, we can only observe the executed motion (pattern) in the dataset as the ground truth for highly interactive scenarios. "Intention" representing desires in human mind is not an appropriate indicator to simplify the representation since there is no ground truth to compare. Therefore, we use executed motion patterns in this chapter to simplify the representation.

Motion pattern in a spatiotemporal domain

As was reviewed and summarized in [138], motion patterns can be categorized hierarchically into route, pass-yield and subtle patterns in various kinds of scenarios. In different driving scenarios with different situations of the entities, the choice of the patterns at each level can be completely different. In general, we suppose there are M possible motion patterns for the predicted entity in a specific highly interactive driving scenario. $\hat{s}_j^i$ represents the jth future motion pattern of the ith entity. Then the distribution to be approximated can be written as

$$P(\hat{s}_j^1 | q^{0:N}, \hat{s}^0),\ j = 1, 2, ..., M. \tag{6.1}$$

In [138], two spatiotemporal representations were summarized, namely, prototype trajectory and reachable set, which can incorporate the generated motion patterns. In this chapter, we adopt prototype trajectory as the spatiotemporal representation. The main purpose is to make it convenient for the methods and algorithms in Section 7.1 to obtain the corresponding probabilities since the outputs of those algorithms are trajectories. Suppose the generated prototype trajectory corresponding to motion pattern $\hat{s}_j^i$ is expressed as $\hat{q}_j^i$. Then we can use the normalized probabilities of

$$p(\hat{q}_j^1 | q^{0:N}, \hat{q}^0),\ j = 1, 2, ..., M. \tag{6.2}$$

obtained from the distribution generated by learning methods to calculate (6.1).

For a given motion pattern, we modify the motion planning approach proposed in Chapter 2 to generate trajectories. The framework proposed in Chapter 2 can deal with various kinds of driving scenarios, and generate smooth, feasible and collision-free (if necessary) trajectories. The computation is extremely fast since only simple A* search and quadratic programming (QP) are involved. The approach is suitable for both online prediction generation, as well as offline evaluation of trajectories for which a large set of predicted trajectories need to be generated.

6.5 Unified Problem Formulation for Interactive Prediction

In this section, the problem to be solved by probabilistic reaction prediction is formulated. Suppose q^i and $\hat{q}^i$ represent the historical and (predicted) future motions of the ith entity, respectively. The host vehicle corresponds to $i = 0$, and the predicted entity corresponds to $i = 1$. Suppose there are N entities to be considered around the host vehicle in the scene at the current time step. Then the original problem which is typically tackled in existing literatures is to obtain desirable models to approximate the conditional probability density function (PDF)

$$p(\hat{q}^1|q^{0:N}). \tag{6.3}$$

However, what is required by the prediction in highly interactive scenarios is far beyond perfectly solving the original problem. When the driving scenarios are highly interactive, the host autonomous vehicle cares about not just the future motion of the predicted entity condition on the historical motions of all relevant entities. The reaction of the predicted entity given different future motions of the host vehicle should also be taken into account. Then the conditional PDF to be approximated for reaction prediction can be written as

$$p(\hat{q}^1|q^{0:N}, \hat{q}^0), \tag{6.4}$$

where $\hat{q}^0$ is the future motion of the host autonomous vehicle.

From the perspective of designing learning algorithms, approximating the reaction distribution (6.4) is much harder than approximating a situational joint distribution of the interaction pair. A situation distribution to be approximated can be written as

$$p(\hat{q}^0, \hat{q}^1|q^{0:N}), \tag{6.5}$$

and it is easy to transform (6.5) to (6.4) since

$$p(\hat{q}^1|q^{0:N}, \hat{q}^0) = \frac{p(\hat{q}^0, \hat{q}^1, q^{0:N})}{p(q^{0:N})p(\hat{q}^0|q^{0:N})} = \frac{p(\hat{q}^0, \hat{q}^1|q^{0:N})}{\int_{\hat{q}^1} p(\hat{q}^0, \hat{q}^1|q^{0:N})}.$$

It means that we can design learning algorithms to learn how to predict the joint distribution of the motions of the predicted entity and the host vehicle. Then we can transform it into reaction predictions.

6.6 Chapter Summary

In this chapter, problem formulation and motion representation of probabilistic prediction methods were reviewed. Based on the review, it was found that different paradigms of probabilistic prediction methods can be more suitable for the formulation of either reaction prediction or situation prediction. Then the mathematical transformation between the two formulations are provided so that a unified framework can be provided for the formulation of various paradigms of probabilistic prediction algorithms.

Chapter 7

Interactive Prediction and Planning Methods and Interface

With the unified framework of formulation and representation constructed in Chapter 6, different paradigms of probabilistic prediction methods are utilized to provide the predicted distribution for the design of interface for interactive prediction and planning in this chapter, as well as a fatality-aware benchmark in Chapter 8.

7.1 Methodologies

In this section, the methodologies to generate probabilistic reaction prediction results are briefly introduced, including hidden Markov model (HMM) as probabilistic graphical model, mixture density network (MDN) as deep neural network, and inverse reinforcement learning (IRL) as planning-based prediction.

Hidden Markov model (HMM)

A hierarchical motion prediction framework is employed. It is essentially a cascade of a situation inference module based on a group of hidden Markov model (HMM), and a motion prediction module based on a group of Gaussian mixture model (GMM) corresponding to each interaction outcome. The labeled trajectories of each potential situation are used to train a HMM individually with the Baum-Welch algorithm, which is a variant of Expectation-Maximization (EM) algorithm. The GMM is used to obtain the conditional distribution of the actions of multiple interactive entities given the current state information.

At the inference stage, given a sequence of historical motion, we can obtain the likelihood of the observation sequences for each HMM by the forward algorithm. Then the likelihood values are normalized to obtain the posterior probability of each situation, which can be written as $p^{k'}$, where $k' = 1, ..., K$, and K is the number of possible situations according to the combination of pass-yield motion patterns [138]. For each GMM, we can obtain the

probabilistic density for the jth prototype trajectory, denoted as f_j^k. Then the probability of the jth prototype trajectory can be obtained by

$$P(\hat{s}_j^1|q^{0:N},\hat{s}^0)=\frac{\sum_{k'=1}^{K}p^{k'}f_j^{k'}}{\sum_{j'=1}^{M}\sum_{k'=1}^{K}p^{k'}f_{j'}^{k'}}. \tag{7.1}$$

Mixture density network (MDN)

We use the mixture density network (MDN) [13] to obtain the joint PDF of the host vehicle and other entities. Instead of learning a single output value using neural networks, MDN is capable of predicting an entire probability distribution for the output using a Gaussian Mixture Model (GMM). Given a set of input states and output actions of multiple traffic participants, MDN can generate necessary parameters to formulate the conditional probability of actions given states.

Given a sequence of prototype trajectory, the performed actions by the vehicle between each state can be obtained. Then at each time step, we can forward the current state into the MDN network and use the conditional distribution to get the likelihood of the action at the given state. For each motion pattern and its corresponding trajectory sequence, we can then multiply the obtained likelihood over the entire horizon and perform a normalization to get the posterior probability for each situation.

Then the approximated distribution for the jth future motion pattern can be formulated as:

$$P(\hat{s}_j^1|q^{0:N},\hat{s}^0)=\frac{\sum_{n=1}^{N_m}w_n\phi_n(\hat{q}_j^1|q^{0:N},\hat{q}^0)}{\sum_{j'=1}^{M}\sum_{n=1}^{N_m}w_n\phi_n(\hat{q}_{j'}^1|q^{0:N},\hat{q}^0)}, \tag{7.2}$$

where N_m denotes the total number of mixture components, w_n is the mixing coefficient and $\phi(\hat{q}|q)$ is the kernel function.

Inverse reinforcement learning (IRL)

Inverse reinforcement learning allows us to learn the cost functions of human by observing their behavior. We assume that all predicted agents are rational, and their cost function along a motion trajectory q can be linearly parametrized as $C(\theta,q)=\theta^T\mathbf{f}(q)$ where $\mathbf{f}(q)$ are features. We also assume that trajectories with lower cost are exponentially more probable based on the principle of maximum entropy [143]:

$$P(q|\theta)\propto e^{-C(\theta,q)} \tag{7.3}$$

In the training phase, the goal of IRL is to find the optimal θ^* that best explains the observed demonstrations in terms of a set of selected features $\mathbf{f}(q)$. Mathematically, we need to solve the following optimization problem:

$$\theta^*(\mathcal{Q}) = \arg\max_{\theta} \prod_{\tilde{q}\in\mathcal{Q}} \frac{e^{-C(\theta,q)}}{\int e^{-C(\theta,\tilde{q})} d\tilde{q}}. \tag{7.4}$$

where $\mathcal{Q}$ represents the set of demonstrated trajectories.

With θ^*, an exponential distribution family is established to approximate the distribution of future trajectories. Different from approaches based on probabilistic graphical models and neural networks, IRL directly generates the conditional probability defined in (6.4) instead of generating the joint distribution over trajectories of the two interacting entities.

In the test phase, given a set of sampled motion patterns $\mathcal{S}$, we can evaluate the normalized probability of each motion pattern $\hat{s}_j^1 \in \mathcal{S}$ via:

$$P(\hat{s}_j^1 | q^{0:N}, \hat{s}^0) = \frac{e^{-C(\theta, \hat{s}_j^1 | q^{0:N}, \hat{s}^0)}}{\sum_{\tilde{s}^1 \in \mathcal{S}} e^{-C(\theta, \tilde{s}^1 | q^{0:N}, \hat{s}^0)}}. \tag{7.5}$$

More details on IRL-based probabilistic prediction can be found in [116].

7.2 Interface for Interactive Prediction and Planning

In Section 7.1, unified output of three paradigms of interactive prediction algorithms, namely $P(\hat{s}_j^1 | q^{0:N}, \hat{s}^0)$ can be obtained, which is the distribution of reaction prediction. It means that given the historical motions of all the relevant entities, as well as the future motion of the host autonomous vehicle, we can obtain the probabilities of possible future motions of the predicted entity.

In order to utilize such powerful tools for interactive prediction, we also need to design the decision and planning framework correspondingly. The integrated decision and planning framework with non-conservatively defensive strategy (NCDS) proposed in Part II has been proved to be effective to deal with uncertainties from prediction modules. In this Section, we will combine the three aforementioned paradigms of interactive prediction algorithms with our decision and planning framework via a properly designed interface.

First, we need to choose the appropriate deterministic planner for the interface. The future motion of the host autonomous vehicle should be provided as an input for the aforementioned prediction algorithms. Then sample-based planner is the most proper choice since the future motions of the host vehicle are already trajectory candidates generated.

We first generated several candidates for the short-term future motion of the host vehicle. The k-th short-term motion candidate $\hat{s}_k^{0,\text{short}}$ has many future candidates. Then the future motion candidates of the predicted entity are also generated, where the j-th candidate is represented by $\hat{s}_j^1$.

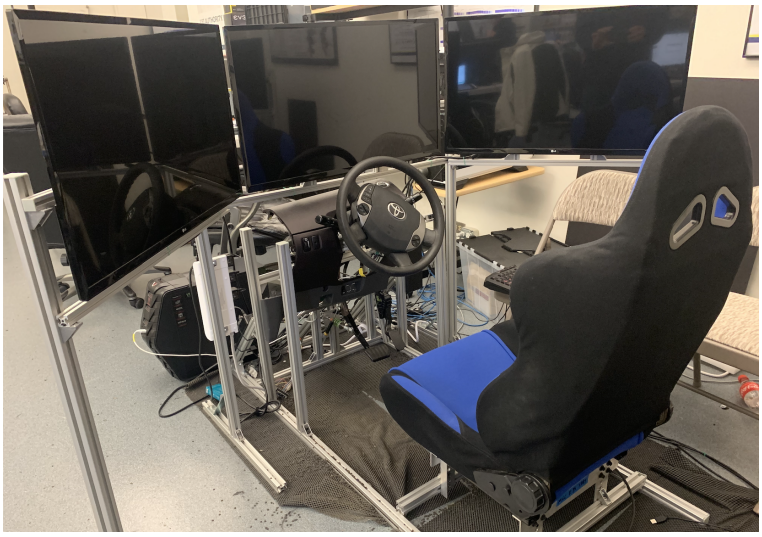


Figure 7.1: Driving simulator for human-in-loop test.

By calculating all the pairs of long-term motion candidates for a specific short-term motion candidate with proper normalization, we can obtain $P(\hat{s}_j^1|q^{0:N}, \hat{s}_k^{0,\text{short}})$. These probabilities can serve as the weights for in the expected costs of $\hat{s}_k^{0,\text{short}}$. Then we can choose the best $\hat{s}^{0,\text{short}}$ according to the comparison of the expected cost, and execute it at the next time step.

7.3 Experiments

We utilized a driving simulator shown in Fig. 7.1 to include a human driver in the loop so that an interactive experiment can be conducted. The roundabout scenario in Fig. 5.6 in the INTERACTION dataset [137] was used to construct the test scenario. The host autonomous vehicle came from the very left branch in Fig. 5.6 and interactive with a vehicle driven by the human driver using the simulator. Interaction pairs from the dataset were manually selected as the training dataset for the aforementioned three prediction algorithms, namely, MDN, HMM and IRL.

A screenshot of one time step of the interaction is shown in Fig. 7.2. The host autonomous vehicle was entering the roundabout in the simulator running the interactive prediction and planning algorithms. In this figure, MDN was used to generate predicted distribution. The corresponding sampled speed profile is shown in Fig. 7.3. For the ego autonomous vehicle, one desirable short-term future trajectory candidate corresponded to five long-term candidates generated. There were also five long-term candidates generated for the obstacle vehicle in the roundabout. Since the obstacle vehicle in the roundabout decelerated, the ego

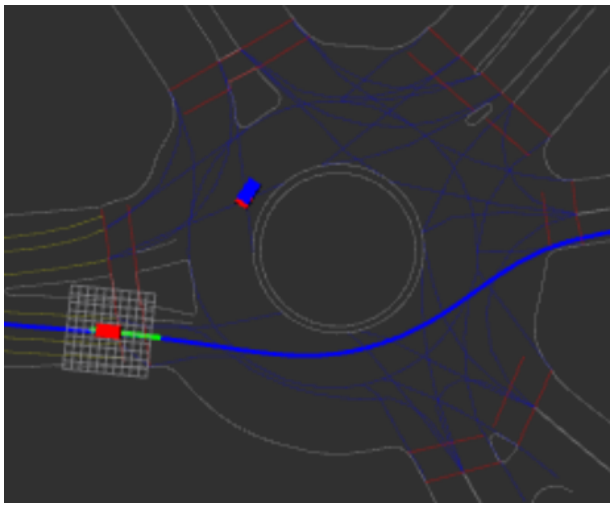


Figure 7.2: Scenarios to test interactive prediction and planning.

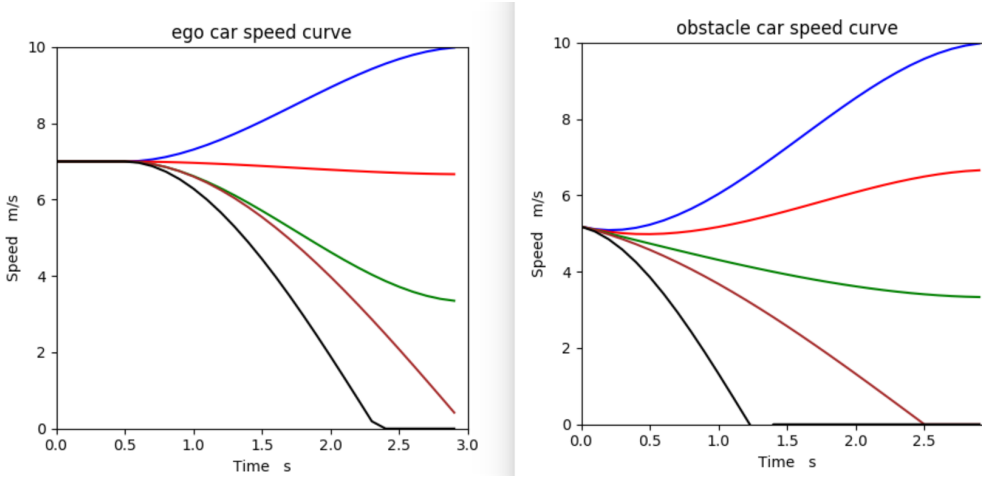


Figure 7.3: Sampled speed profile.

autonomous vehicle tended to keep the current speed to achieve efficient motion.

Another case is shown in Fig. 7.4 and 7.5 with an IRL predictor. The ego vehicle accelerated since the obstacle vehicle decelerated. The long-term sample of the ego car in Fig. 7.5 colored black served as a safety guarantee to make sure that the ego vehicle can avoid a collision in case that the obstacle vehicle finally chose to accelerate.

The computational efficiency of the whole pipeline was relatively high. We were able to guarantee 5 Hz with MDN and 10 Hz with HMM and IRL in the experiments.

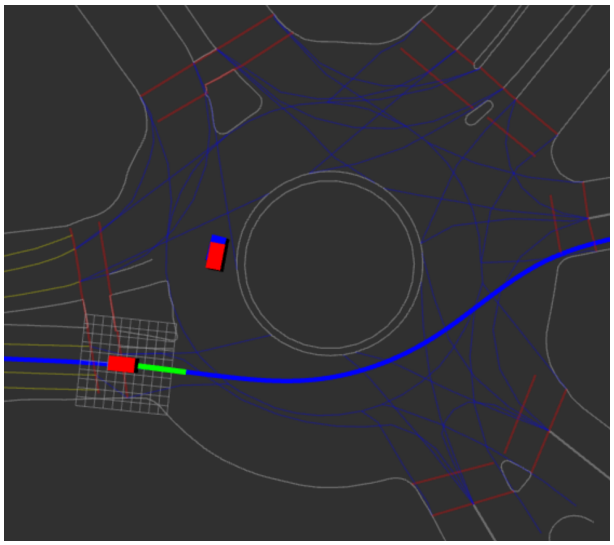


Figure 7.4: Scenarios to test interactive prediction and planning.

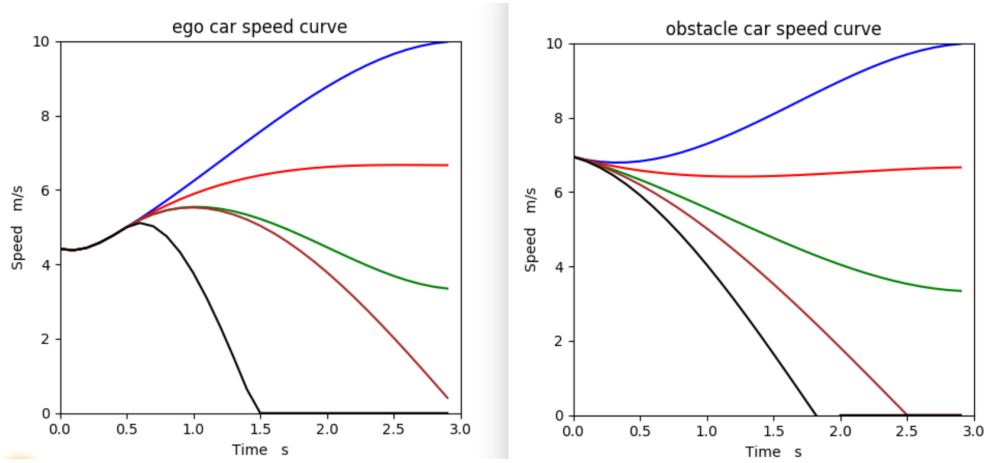


Figure 7.5: Sampled speed profile.

7.4 Chapter Summary

In this chapter, three paradigms of interactive prediction algorithms were introduced with unified output for reaction prediction by utilizing the problem formulation transformation proposed in Chapter 6. Then an interface was designed for interactive prediction and planning, which can incorporate all the different paradigms of predictors, as well as an integrated decision and planning framework with non-conservatively defensive strategy. Human-in-loop

experiments were conducted using a real-world scenario. The predictors were trained using motion data from real world in the same scenario. Results showed the effectiveness of the proposed interface.

Part IV

Prediction Benchmark and Dataset

Chapter 8

Fatality-Aware Prediction Benchmark

8.1 Introduction

Accurate prediction of the future motions of surrounding entities is a prerequisite for autonomous vehicles to make decisions and plan motions under uncertainties [53][110][22] that are safe with high driving quality. Probabilistic prediction is necessary since the human behavior is full of uncertainties. The accuracy of generated prediction probabilities can significantly impact the safety and driving quality of autonomous vehicles. As was stated in [134], a desirable driving strategy of autonomous vehicles should be defensive to real threats, but not conservative to threats of low probability. A fatal accident may happen if the prediction algorithm ignores a real threat by mistaking its probability as zero, which makes the driving strategy non-defensive. On the other hand, the decisions and motions of autonomous vehicles can be very conservative if the prediction algorithm overestimates the probability of a threat and generates false alarms. Therefore, accurate probabilistic prediction is a key building block for safe and high-quality autonomous driving.

Evaluation metrics are required to measure the performance of predictions. Distance-based trajectory similarity metrics were investigated and employed in [8][45][142][25]. Comprehensive reviews on distance-based metrics were provided and novel measures were proposed in [12][98]. Distance-based metrics are well applicable to evaluate deterministic predictions. However, the evaluation of probabilistic prediction cannot be provided by using distance-based metrics directly.

A variety of learning metrics were used to evaluate the performance of probabilistic prediction models on whether the distribution in the dataset is well approximated, such as area under the curve (AUC) [64], root mean square error (RMSE) [72], likelihood [93], and Kullback-Leibler (KL) divergence [126]. Each of the metrics has its own limitations which may lead to difficulties or misinterpretations to measure the performance of probabilistic prediction. We need to choose a proper metric to measure the performance of prediction

algorithms to approximate the data distribution.

The executed motion of a vehicle typically satisfies planning constraints on feasibility (vehicle model) and safety (collision avoidance). However, small perturbations making a trajectory infeasible or unsafe, can hardly make a large difference for learning and distance-based metrics. The aforementioned problem has not been sufficiently addressed for evaluation. It is also a concern for the decision and planning module whether accurate prediction is provided timely with a sufficient preview horizon. Neither the learning nor distance-based metrics can take into account these aspects.

Moreover, the purpose of probabilistic prediction is not limited to approximate the data distribution. The prediction results are used online for decision-making and planning modules. Safe and high-quality motions are expected by adopting the prediction outputs. Therefore, the evaluation metric should reveal the decision consequence due to the inaccurate prediction, such as how non-defensive or conservative the planned motion would be, and what is the fatality of the consequence. A fatality-aware metric for prediction is expected, which has not been addressed in existing works. Also, when evaluating predicted motions of surrounding entities, we should also take into account prior knowledge such as vehicle kinematics and dynamics, as well as rare collision in the real world. In fact, satisfying feasibility and safety requirements is hard to achieve for many existing methods.

The main contribution of this chapter is to propose a fatality-aware evaluation metric for probabilistic reaction prediction in extremely challenging driving scenarios with interaction. The proposed metric can reveal the fatality of prediction errors by considering the criticality of the corresponding motion pair. Moreover, we utilize the unified framework with homogenized problem formulation and motion representation proposed in Chapter 6, which can facilitate the evaluation of different types of methods such as PGM, NN and IRL. We implement these methods in highly interactive ramp merging scenarios, and evaluate the prediction performances with the proposed metrics with analysis. We also provide a summary of learning metrics, and address the crucial but omitted aspects on prediction evaluation from the perspective of decision-making and planning, so that novel metrics can be inspired.

8.2 Survey on Evaluation Metrics

In many recent works, probabilistic prediction was regarded as a pure machine learning problem, that is, to find a learning model which can best represent the distribution in the dataset in the sense of some training metrics. However, distribution learned from data alone for prediction is not sufficient to support high-quality and safe decision-making and planning. In this section, the most commonly used learning metrics are summarized. Also, we discuss potential metrics based on prior knowledge on planning constraints, and aspects of metric construction which are crucial for decision-making and planning.

Learning metrics

Typical classification and regression metrics are widely used to evaluate the performance of learning models for probabilistic prediction. Despite the most intuitive metric, accuracy, the following metrics are also widely used for evaluation of probabilistic predictions.

Receiver operating characteristic (ROC) curve

ROC curve is typically used to illustrate the performance of a binary classifier with different thresholds of discrimination. The area under the curve (AUC) is often used as a quantitive metric. ROC and AUC were employed to evaluate the classification performance of intention estimation, such lane change [9] and intersection maneuvers [64][76], as well as situation prediction [63].

Root mean square error (RMSE)

RMSE is one of the most widely used metric for regression evaluation. Some variations in different literatures evaluate the performance from similar perspective, such as root-weighted square error (RWSE) [86][72][126], mean absolute error (MAE) [9][60], Mean Error (ME) [96]. RMSE is mostly used to represent the errors of continuous motions between the (sampled) prediction and ground truth, such as acceleration [128][1], velocity [86][72][126][62], as well as position and distance [9][126][62]. A special implementation of RMSE is for evaluating the error of discretized probability weighted by the corresponding occupancy grid distance [60].

Likelihood

Likelihood is commonly used metric for training probabilistic models [86][72][60][28][68][120], although it is not used as widely to evaluate the performance of the trained models since the likelihood values are not well interpretable. Likelihood is used as an evaluation metric in [96][126][93].

Kullback-Leibler (KL) divergence

KL divergence is a widely-used metric in machine learning field to measure discrepancies between two probability distributions. It is used as an evaluation metric for probabilistic prediction in [86][126]. KL divergence is also employed in [122] to assess the similarity of data observation and provide the value of additional motion data.

Prior metrics

The decision-making and planning module of the host vehicle need to take into account feasibility constraints according to vehicle kinematics and dynamics, as well as safety constraints for collision avoidance and hard traffic rules. The predicted motions have similar

requirements as planned motions, although the requirements are not as strict. A small perturbation of a trajectory may change it from a safe and feasible one to an unsafe or infeasible one, especially when the trajectory is near the constraint boundary. However, such perturbation may not make a large difference on learning metrics or other distance-based metrics. Therefore, in order to achieve a comprehensive evaluation of predicted motion distribution, prior metrics based on planning constraints should also be considered, which is rarely mentioned in existing works.

Feasibility violation

Some prediction methods or frameworks can inherently guarantee feasibility. For instance, [7] incorporated Rapidly-exploring Random Tree (RRT) into the prediction frameworks so that the generated motions can be dynamically feasible. [81] incorporated kinematic bicycle model into the recurrent network structure, which guarantees that the output of the predictor satisfies hard constraints on vehicle kinematics. It is an ideal case if the generated motions are sampled from obtained distributions via approaches with feasibility guarantee. Otherwise, violation verification and evaluation are necessary, except for rare cases when one can recognize a driver completely losing control of the vehicle.

Safety violation

The requirements on safety are different for prediction and planning. The planned motions for the host vehicle need to consider safety as the top priority, and try its best to guarantee safety if possible. On the contrary, the most important aspect for prediction is to be human-like. Violations to hard traffic rule (red light, stop sign, etc.) and collisions between road users are not common in realistic human driving, but is not extremely rare. Therefore, it is not appropriate to forbid all unsafe motions in predictions, but some obviously unrealistic collisions or violations should not be generated. Values of negative distance headway and negative speed were employed in [86] to evaluate unrealistic motions generated. Collision rate and average distances between the host and the closet merging cars are used as the metric for safety violation and margin of the predicted motions in [28].

Decision-related metrics

The learning and prior metrics can help us evaluate how human-like the generated prediction is. There are still several aspects to be considered to evaluate predictions according to the consequences when the prediction is adopted by decision-making and planning modules.

Fatality

The consequential fatalities of different inaccurate or incorrect predictions can be completely different when they are adopted by the decision and planning module. The inaccuracy

and incorrectness should not be treated equally in the metric. Fatality and criticality differences should be revealed. It is also related to the safety concerns in Section 8.2, but in a more comprehensive way.

It is crucial for the safety and driving quality of the host autonomous vehicle to survive in the worst case (defensive driving strategy), yet not overreact to threats of low or zero probability (non-conservative strategy) [134]. In order to achieve such a non-conservatively defensive driving strategy, the decision and planning module requires the prediction module to acquire the following two capabilities. One is to enhance *defensiveness*, namely, to provide all possible future motions (completeness), including possible violations to the traffic rules, and possible careless or dangerous behaviors, so that the host vehicle can drive defensively to potential threats. On the other hand, *conservatism* should be reduced, that is, to provide zero probability (or a safe to pass indicator) when the corresponding motion is impossible, so that the host vehicle can proceed without hesitation. Similar evaluations were also [70] as *miss detection rate* corresponding to *defensiveness* and *false alarm rate* corresponding to *conservatism*.

Timeliness and preview horizon

Decision and planning module expect correct and accurate predictions to be provided timely, especially for those causing potential fatal accidents. Correct and accurate predictions are meaningless if such motion has already been completed or an accident is inevitable. Therefore, time-to-event (TTE) [96] and time-to-intersections (TTI) [64] variables were used as the horizontal axis with learning metrics as the vertical axis, so that the timeliness of accurate predictions can be revealed. Distance-to-event variables were also employed in [127], in which specific quantile, mean and standard deviation of distance until correct classification (DCC) were used as metrics. [70] also addressed timeliness from threat estimation perspective as time-to-collision when dangerous situation is correctly classified.

The length of the preview horizon is crucial for a motion planning module. Long-term horizon is preferred for safety and driving quality. Accordingly, the predicted motion should have the same horizon as required by the planning module. Therefore, the ideal case is to provide the distribution of the predicted motion within a long-term horizon, and compare the accuracy over the horizon length, as was illustrated in [36][9].

Computational cost

When a probabilistic method serves as an online prediction module, the host vehicle requires it to provide results in real time. The computational cost should be considered as a metric, especially when the size of the model is relatively large. Computational time was provided in a few recent works [65] for comparison.

8.3 Fatality-Aware Evaluation Metric

In this section, we address three aspects to obtain appropriate evaluation metrics. The first is on whether the predicted motions satisfy safety and feasibility requirements based on prior knowledge. Next is to select an appropriate baseline metric. Finally, we propose the fatality-aware metric based on the baseline.

Prior knowledge

Based on our prior knowledge, the predicted motion of vehicles should at least satisfy a simple kinematic model, and collisions should be extremely rare according to the statistics of real-world driving. In other words, safety and feasibility should also be checked when evaluating the prediction performance. However, it is a difficult task for algorithms based on pure neural networks or probabilistic graphical models to satisfy feasibility and safety constraints.

Since we are using the prototype trajectories generated in Section 6.4 to represent possible motion patterns, it is relatively easy to make the generated trajectories satisfy the requirements on safety and feasibility. It can alleviate the requirements on safety and feasibility for learning-based models, and additional verifications are not necessary.

Baseline metric

Existing works typically use metrics such as area under the curve (AUC), likelihood, root mean square error (RMSE), and Kullback-Leibler (KL) divergence to evaluate probabilistic predictions. In this subsection, we briefly discuss the deficiencies of each metric when it is employed for evaluation of probabilistic predictions.

AUC is a metric which is typically used for binary classification, which is not inherently designed to reveal the accuracy of the probabilistic distributions. Likelihood sufficiently measures accuracy of the predicted probabilities or distribution for the ground truth data points. However, it is not possible to indicate how bad the prediction is if high density is generated for motions with low or zero probability.

RMSE measures the error between the ground truth and sampled trajectories from the predicted distribution in Euclidean space. However, the Euclidean distance can be very small between collision-free, critical and colliding trajectories, or between feasible and infeasible trajectories. A small perturbation of the trajectory in Euclidean space can make it completely different on whether or not the trajectory is collision-free or feasible. Also, RSME fails to reveal the approximation performance for multimodal data [101].

KL divergence requires the description of the ground truth distribution, which is extremely hard to obtain in a high dimensional space. Estimation or approximation of the distribution of the test data may not be executable with limited data points in the high dimensional space of the motions.

Therefore, we need an appropriate metric without the aforementioned deficiencies. Brier score [17] is a metric measuring the accuracy of probabilistic predictions, which is widely used in research fields requiring evaluation of probabilistic predictions, such as weather forecast. It can evaluate the prediction performance directly from the ground truth data points without estimated distribution of the test set. Also, the score can both reward high probability for ground truth patterns and penalize overestimation of other patterns. By properly generating motion patterns, the score can also avoid the problem introduced by using Euclidean space.

Suppose $\hat{s}_g^0(k, T_h)$ and $\hat{s}_g^1(k, T_h)$ are the ground truth motion pattern of the host and predicted vehicle future motion with preview horizon T_h for the kth sample in the test set of data. The total number of samples is N_s. We define

$$P_j(k, T_h) = P(\hat{s}_j^1(k, T_h)|q^{0:N}, \hat{s}_g^0(k, T_h)).$$

We also define

$$O_j(k, T_h) = O(\hat{s}_j^1(k, T_h)|q^{0:N}, \hat{s}_g^0(k, T_h))$$

to represent the actual outcome (0 or 1) on whether the jth motion pattern corresponds to the ground truth. According to the definition of Brier score, the baseline metric for probabilistic prediction can be written as

$$\mathcal{B}(T_h) = \frac{1}{N_s M} \sum_{k=1}^{N_s} \sum_{j=1}^{M} [P_j(k, T_h) - O_j(k, T_h)]^2. \tag{8.1}$$

Fatality-aware metric

The baseline metric (8.1) equally weights each probability error $(P - O)^2$. However, each error may have different impact to the prediction accuracy due to the difference of the criticality of each motion pairs.

Suppose $\mathrm{Cr}_{j_1,j_0}(k, T_h)$ denotes a score of the criticality for the motion pair of the predicted entity $\hat{s}_{j_1}^1(k, T_h)$ and host vehicle $\hat{s}_{j_0}^0(k, T_h)$. It can be the inverse of time-to-collision (TTC) or other scores which represents how critical motion pair is for a potential collision. Since only the ground truth motion pattern of the host vehicle is used for evaluation, we use $\mathrm{Cr}_j(k, T_h)$ for the motion pair of the predicted entity $\hat{s}_j^1(k, T_h)$ and host vehicle ground truth $\hat{s}_g^0(k, T_h)$

Then we can sort the predicted motion patterns according to the criticality from low to high as $\left\{\hat{s}_{1:m}^1, \hat{s}_g^1, \hat{s}_{m+2:M}^1\right\}$. Motion patterns $\hat{s}_{1:m}^1$ are less aggressive than the ground truth, while $\hat{s}_{m+2:M}^1$ are more aggressive. $\mathrm{Cr}_g(k, T_h)$ is the criticality of motion pair $\hat{s}_g^0(k, T_h)$ and $\hat{s}_g^1(k, T_h)$.

An intuitive illustration is provided in Figure 8.1 to explain how we can use the values of criticality to achieve better evaluation of probabilistic predictions. There are two predicted distributions with similar Brier scores and similar probability outputs for ground truth (similar likelihood). The distribution colored blue assigns high probabilities to motions which

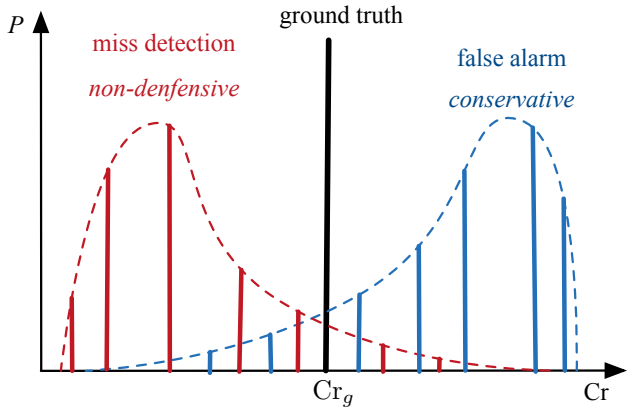


Figure 8.1: An intuitive illustration of conservatism and non-defensiveness by using two distributions with similar probability outputs for ground truth and similar Brier scores.

are much more critical than the ground truth. Such inaccurate predictions overestimate the threats and generate false alarms. It can lead to conservative decisions and motions of the host autonomous vehicle. By contrast, the distribution colored red provides high probabilities to motions which are much less critical than the ground truth. Such inaccurate predictions underestimate the real threat (ground truth) and generate miss detections. It can lead to non-defensive decisions and motions of the host autonomous vehicle, and may cause fatal accidents.

Therefore, the inherent distinction of the consequences (conservative and non-defensive) by adopting the predicted distribution should be revealed in the evaluation metric. Also, the more the criticality deviates from the ground truth, the more penalty should be received by the corresponding predicted probability. We propose to separate the baseline metric (8.1) into three parts to achieve such purpose.

The first part contains the prediction errors of ground truth reaction. which can be written as

$$\mathcal{G}(T_h) = \sum_{k=1}^{N_s} \frac{1}{N_s M} [P_j(k, T_h) - 1]^2. \tag{8.2}$$

The weights remain the same as the baseline metric (8.1).

For all motion patterns which are more aggressive than the ground truth, the probability errors $(P_{m+2:M} - 0)^2$ correspond to false alarms on more dangerous reactions than what really happened. In other words, the prediction algorithm overestimates the aggressiveness of the reaction of the others. It makes the decision of the host autonomous vehicle conservative to false threats. Therefore, we denote a score to measure *conservatism* of the prediction

algorithm based on the baseline metric (8.1), that is

$$\mathcal{C}(T_h) = \sum_{k=1}^{N_s} \sum_{j=m_k+2}^{M} \frac{\mathrm{Cr}_j(k, T_h) - \mathrm{Cr}_g(k, T_h)}{S} P_j(k, T_h)^2, \tag{8.3}$$

in which S is the summation of all weights other than those for the ground truth. We use S to normalize the weights.

For all motion patterns which are less aggressive than the ground truth, the probability errors $(P_{1:m} - 0)^2$ correspond to miss detections. In other words, the prediction algorithm fails to predict a more aggressive reaction, which is the ground truth. It makes the decision of the host autonomous vehicle less defensive to real threats. Therefore, we denote a score to measure *non-defensiveness* of the prediction algorithm based on the baseline metric (8.1), that is

$$\mathcal{D}(T_h) = \sum_{k=1}^{N_s} \sum_{j=1}^{m_k} \frac{\mathrm{Cr}_g(k, T_h) - \mathrm{Cr}_j(k, T_h)}{S} P_j(k, T_h)^2, \tag{8.4}$$

Then the fatality-aware weighted metric can be written as follows, which contains the aforementioned three aspects.

$$\mathcal{B}_c(T_h) = \mathcal{D}(T_h) + \mathcal{G}(T_h) + \mathcal{C}(T_h). \tag{8.5}$$

8.4 Case Study

In this section, the metrics in Section 8.3 are employed to evaluate the performances of the algorithms in Section 7.1. We use an exemplar scenario to provide a mini benchmark for the three methodologies. Highway ramp merging is a highly interactive driving scenario. It can be extremely challenging when the traffic flow is relatively slow, where the merging vehicles have to nudge into a gap on the target lane. The gaps are often very small so that the merging vehicle have to interact with a target vehicle to force it to enlarge the gap. The merging may fail and the merging vehicle have to resort to the next gap behind the target vehicle.

The ramp merging cases in NGSIM dataset were used for the training and evaluation of the algorithms. The cases were manually selected to be highly interactive ones. In each case, we only chose the frames in which the merging vehicle has not merge into the target lane successfully and it is still interacting with the target vehicle. 7102 data samples were used for training, and 708 data samples were used for evaluation.

The merging and target vehicle was treated as the host and predicted vehicle, respectively. The preview horizon $T_h = 3$ s. The number of motion patterns of the reaction of the target vehicle was set as $M = 4$. An illustrative example of prototype trajectories represented by longitudinal positions over time was provided in Figure 8.2. It was a segment in the ramp

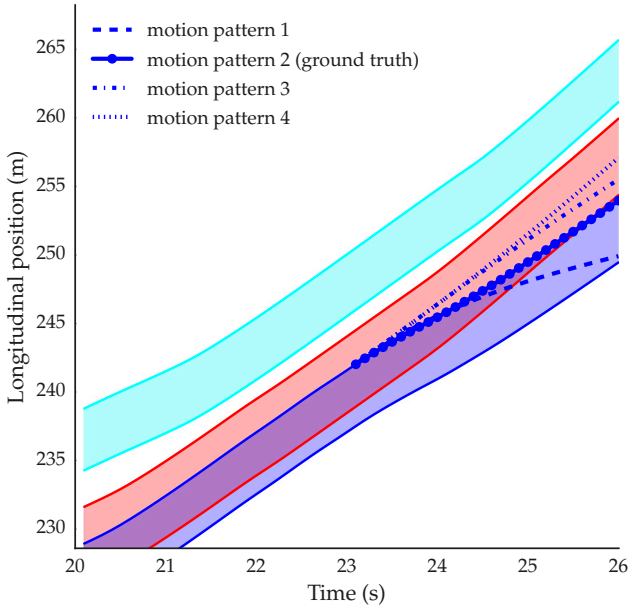


Figure 8.2: An illustrative example of prototype trajectories represented by longitudinal positions over time.

merging case shown in Figure 6.3. The current time step is 23 s. The target actually started to yield the merging vehicle. Motion pattern 1 was less aggressive than the ground truth motion pattern. Motion patterns 3 and 4 were trying to keep the small gap, which were more aggressive than the ground truth.

The criticality score Cr_j was defined as the inverse of time-to-collision (TTC) for the front end of the target vehicle to hit the (potential) merging point of the merging vehicle [141]. Merging point was defined as the longitudinal position of the rear end of the merging vehicle when its (potential) vehicle body overlaps with the potential path of the target vehicle.

The performance scores according to the aforementioned metrics are shown in Table 9.2 for hidden Markov model (HMM), mixture density networks (MDN) and inverse reinforcement learning (IRL). If we use the baseline Brier score as the metric, HMM and MDN outperformed IRL since the score of IRL was much higher than those of the other two. However, if we partition the score $\mathcal{G}$ for the error at the ground truth data points, we can find that $\mathcal{G}$ had great impact to raise $\mathcal{B}$. It revealed that methods approximating the distribution directly trained with log likelihood, such as HMM and MDN, can easily outperform IRL if we only care about assigning high probabilities at ground truth data points.

For *conservatism* $\mathcal{C}$ and *non-defensiveness* $\mathcal{D}$, the scores of IRL were much lower than those of the other two methods. It demonstrated that IRL tended to produce relatively high probabilities for prototype trajectories with similar criticality as the ground truth. The reason may come from the nature of IRL to approximate the reward/cost function other than

Table 8.1: Performance scores of each method

	HMM	MDN	IRL
$\mathcal{B}$	0.1403	0.1099	0.1821
$\mathcal{G}$	0.0710	0.0564	0.1117
$\mathcal{C}$	0.0476	0.0493	0.0178
$\mathcal{D}$	0.1365	0.1303	0.0698
$\mathcal{B}_c$	0.2551	0.2361	0.2053

the data distribution, which makes the motions generated by IRL more interpretable. Such properties can help the host autonomous vehicle to avoid conservative behaviors by reducing the probabilities of potential reactions which are much more critical than the ground truth. More importantly, the autonomous vehicle can behave defensively to avoid underestimating real threat. The advantage of such property was also reflected in the criticality-aware metric $\mathcal{B}_c$. The score of IRL became lower than those of the other two.

Note that what we discussed in this Section analyzed the specific methods we implemented. It does not necessarily conclude that one paradigm or one type of methodologies is better than others in those aspects. By properly modifying the methods, tuning parameters or redesigning the framework, better performances can be achieved.

8.5 Chapter Summary

In this chapter we proposed a fatality-aware metric to evaluate the performance of probabilistic reaction prediction in highly interactive driving scenarios. We utilized the three methods based on probabilistic graphical model (PGM), neural network (NN) and inverse reinforcement learning (IRL) mentioned in Chapter 7 with unified homogenized problem formulation and representation simplification proposed in Chapter 6. By using prototype trajectories with designated motion patterns as the simplified representation, the requirements on collision avoidance and feasibility can be satisfied. We employed Brier score as the baseline metric to overcome the deficiencies of the existing metrics. We proposed a weighted Brier score based on the criticality of the interactive motion pairs. The proposed evaluation metric emphasized the fatality of the consequences when corresponding predictions are adopted. Conservatism and non-defensiveness were also defined based on the proposed metric for analyzing the performance of the prediction algorithms. Analysis on the implemented methods was provided by comparing the baseline and proposed metric scores of each method.

Chapter 9

INTERACTION Dataset

9.1 Introduction

In order to enable fully autonomous driving in complex scenarios, comprehensive understanding and accurate prediction of the behavior and motion of other road users are required. Moreover, autonomous vehicles need to behave like vehicles with human drivers to make themselves more predictable to others and thus, facilitate cooperation. These are two of the major challenges in the field of autonomous driving. To overcome these challenges, considerable amount of research efforts have been devoted to: i) predicting the future intention and motion of other road users [71, 104, 138], ii) modeling and analyzing driving behavior [89, 30], iii) clustering the motion and finding representation of the motion primitives [77, 124], iv) cloning and imitating human and expert behavior [59, 102], and v) generating human-like and social behavior and motion [118, 44, 87].

All the aforementioned research areas require interactive vehicle motion data from real-world driving scenarios, which is the most fundamental and indispensable asset. NGSIM dataset [2] is the most popular one used in the aforementioned areas, such as prediction [116, 140, 3], behavior modeling [30], social behavior generation and planning [118], and representation learning [77], since it is publicly available with decent scale and quality. The recently released highD dataset [66] also greatly assists behavior-related research such as prediction [84]. Public motion datasets such as NGSIM and highD facilitated, but also restricted behavior-related research due to limited diversity, complexity and criticality of the scenarios and behavior. Also, the importance of map information and completeness of interaction entities were under-addressed in most of the existing datasets. However, these missing points are crucial for behavior-related research, which will be discussed in the following.

1) Diversity of interactive driving scenarios: Recent behavior-related research using public datasets was mostly restricted to highway scenarios due to the data availability. There are many more highly interactive driving scenarios to explore, such as roundabouts with yield/stop signs, (unsignalized) one/two/all-way stop intersections (shown in Fig. 9.1), signalized intersections with unprotected left turn, zipper merge in cities, etc.

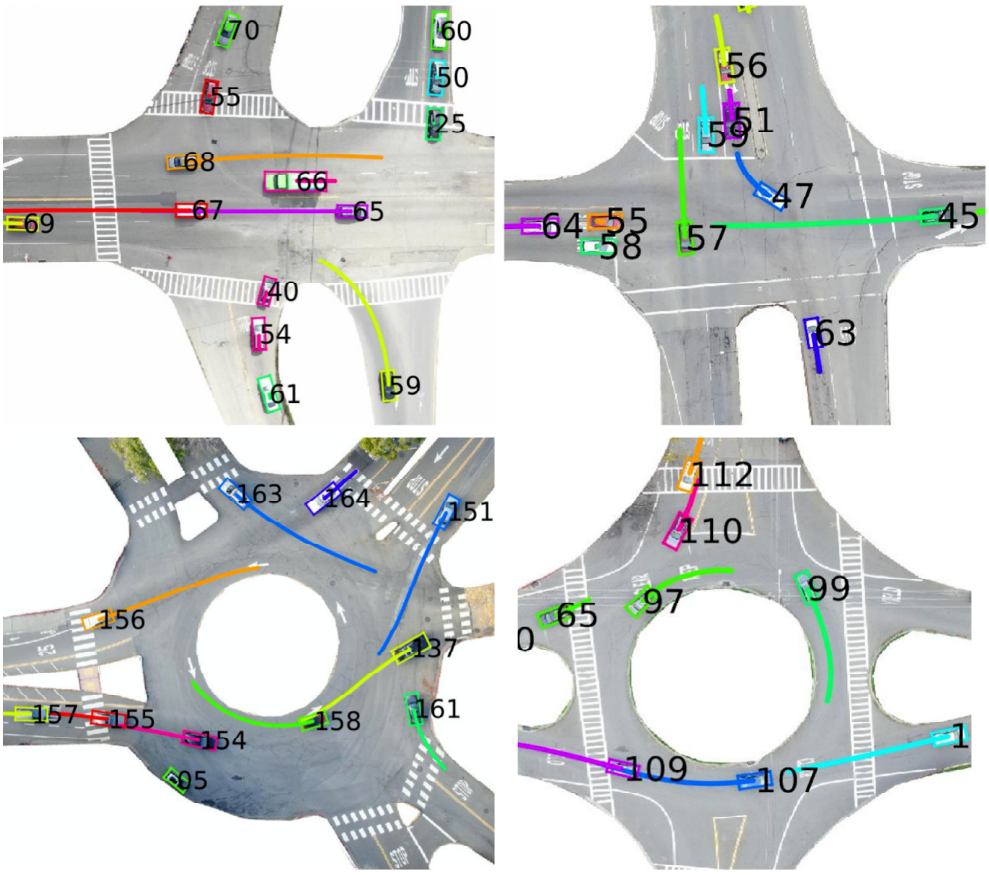


Figure 9.1: Examples of the detection and tracking results in highly interactive driving scenarios in the dataset.

2) International diving cultures: Most of the existing datasets only contain driving data in one specific country. However, driving cultures in different countries and different continents can be distinct for very similar scenarios. Without motion data in similar scenarios from different countries, it is not possible to incorporate the impact of driving cultures in different countries, such as driving styles, preferences, risk tolerance, understanding of traffic rules, etc., for behavior modeling and analysis as well as the design of adaptive prediction and planning algorithms in different countries.

3) Complexity of the scenarios and behavior: Most of the scenarios in the existing public datasets are relatively simple and structured with explicit right-of-way. The behavior of the drivers is only occasionally impacted by others. There is very little social pressure (such as several vehicles waiting behind and even honking) on the drivers, so that their behavior is cautious without aggressive and irrational decisions. A motion dataset with much more complex and interactive behavior and scenarios is expected to facilitate the research tackling

real and challenging problems.

4) Criticality of the situations: Critical situations (such as near-collision cases) are much more challenging and valuable than others for behavior-related research areas. For instance, [140] proposed a fatality-aware prediction benchmark emphasizing prediction inaccuracies in critical situations. However, critical situations are too sparse in existing motion datasets, and can hardly be identified. Therefore, a motion dataset with denser critical situations is necessary to facilitate the research efforts on those difficult problems.

5) Map information: Map information with references and semantics such as lanelet connections and traffic rules, are crucial for behavior-related research areas such as motion planning and prediction. It provides key information on input (features), such as route and goal point [102], distance to the merging point [116, 140], lateral position within the lane [118], etc., and makes the algorithms generalizable to other scenarios. Such semantic maps are currently missing for most of the existing public motion datasets.

6) Completeness of interaction entities: In order to accurately model, predict and imitate the interactive vehicle behavior, it is crucial to provide motions of all surrounding entities which may impact their behavior in the dataset. This requirement was often overlooked when using motion data collected by onboard sensors due to occlusions and limited field of view of the sensors. Although existing motion datasets collected from onboard sensors contain data collected from a wide range of areas for long time periods, complete and meaningful interaction pairs are relatively sparse.

In this chapter, we will emphasize all the aforementioned aspects to construct an international motion dataset collected by drones and traffic cameras.

- *Diverse and international:* It contains a variety of highly interactive driving scenarios from different countries, such as roundabouts, signalized/unsignalized intersections, as well as highway/urban merging and lane change.

- *Complex and critical:* Part of the scenarios are relatively unstructured with inexplicit right-of-way. The driving behavior in the dataset are highly impacted by other drivers, whose behavior can be aggressive or irrational due to the social pressure. Near-collision or slight-collision scenes are contained in the dataset to facilitate the research for critical situations.

- *Semantic map and complete information:* HD maps with semantics are provided to generated key features in the context. Motions of all entities which may influence the driving behavior are included in the dataset.

The proposed dataset can significantly facilitate behavior-related research such as motion prediction, imitation learning, decision-making and planning, representation learning, interaction extraction and social behavior generation. Results from exemplar methods in all these areas are provided utilizing the proposed dataset.

Table 9.1: Scenario comparison with existing motion datasets

	highly interactive scenarios	complexity of scenarios
NGSIM [2]	ramp merging, (double) lane change	structured roads, explicit right-of-way
highD [66]	lane change	structured roads, explicit right-of-way
Argoverse [20]	unsignalized intersections, pedestrian crossing	unstructured roads, inexplicit right-of-way
INTERACTION	roundabouts, ramp merging, double lane change unsignalized intersections	unstructured roads, inexplicit right-of-way

Table 9.2: Comparison with existing motion datasets

	density of aggressive behavior	near-collision situations and collisions	HD maps with semantics	completeness of interaction entities & viewpoint
NGSIM [2]	low	very few near-collision	no	yes, bird's-eye-view from a building
highD [66]	low	very few near-collision	no	yes, bird's-eye-view from a drone
Argoverse [20]	low	no	yes, but partially	only for the ego data-collection vehicle
INTERACTION	high	yes	yes	yes, bird's-eye-view from a drone

9.2 Related Work

Datasets from Bird's Eye View

As mentioned in Section 9.1, NGSIM dataset [2] is the most popular vehicle motion dataset among the behavior-related research communities. The raw data was collected by cameras mounted on buildings and processed automatically [61]. The accuracy of the dataset is mostly acceptable. However, there may be steady errors, and the image projection can significantly enlarge the size of the vehicles. Researchers proposed methods [24] to rectify the errors, but it can only improve the quality of a small part of the dataset. In view of the problems in NGSIM, highD dataset [66] was constructed by using a drone with more accurate vehicle motions and larger amount of high way driving data than NGSIM. Other

datasets [131, 103] from bird's eye view are more focused on pedestrian behavior without strong vehicle interactions.

The driving scenarios presented in NGSIM and highD are quite limited. NGSIM contains highway driving (including ramp merging and double lane change) and signalized intersection scenarios. In fact, signalized intersections are mostly controlled by the traffic lights and interactions are very rare and slight. A small amount of lane changes are interactive, but most of them are neither interactive nor critical. Ramp merging and double lane change can be highly interactive when the traffic is relatively dense, but the amount of interaction is still relatively limited in NGSIM. HighD only contains highway driving scenarios with car following and lane change. Urban scenarios which contain densely and highly interactive behavior, such as roundabouts and unsignalized intersections are not included in either of the two public datasets of vehicle motions.

Datasets from Onboard Sensors

In addition to the bird's-eye-view motion datasets, two types of onboard-sensor-based ones are also publicly available. One includes motion data of surrounding entities from onboard LiDARs and front-view cameras, such as Argoverse [20] and HDD dataset [99]. The other only contains motions of many data-collection vehicles from onboard GPS, such as 100-car study [88].

There are two major advantages for datasets from onboard sensors. One is that a variety of driving scenarios with relatively long data recording time are usually included in those datasets, such as urban driving at signalized/unsignalized intersections and highway driving with ramp merging, etc. The other is that the occlusions of LiDARs and cameras are recorded so that the actual occlusions from perspective of the ego vehicle can be partially recovered.

Completeness of interaction entities is a major problem when using datasets from onboard sensors for behavior-related research. For motion datasets with GPS-based fleets, it is hard to determine whether the vehicles in an "interactive" motion segment was actually interacting with each other since there is no motion recording of other surrounding vehicles (or even pedestrians) without GPS devices installed. For motion datasets constructed from onboard LiDARs and cameras, it is hard to guarantee that all the surrounding objects impacting the behavior of other vehicles are included in the dataset when predicting the motions of others. Therefore, complete interactions are relatively sparse in such kind of datasets. If the sensors cannot cover the full field of view, it will be even impossible to guarantee the completeness of information for the surrounding entities of the ego data collection vehicle.

Also, the data collected in a large area may lead to very few repetitions at the same location. It is hard to learn multi-modal driving behavior for prediction or planning since only one sequence of motions can be found with similar features at the same location.

Map information is also missing in most of the motion datasets. To the best of our knowledge, Argoverse is the only motion dataset providing relatively rich map information. Physical layer (locations of curbs, road markings, etc.) is contained and semantic information

(lane bounds and turn directions, etc.) required by prediction and planning is partially included.

Table 9.2 provides a comparison of the three most useful public vehicle motion datasets as well as the one presented in this article. The proposed dataset contains much more diverse, complex and critical scenarios and vehicle motions comparing to the other three. In addition, HD maps with full semantic information are provided, and the completeness of interaction entities is superior to datasets from onboard sensors.

9.3 Features of the Dataset

In this section, we will illustrate the features of the proposed dataset by highlighting the diversity, internationality, complexity, criticality, and semantic map.

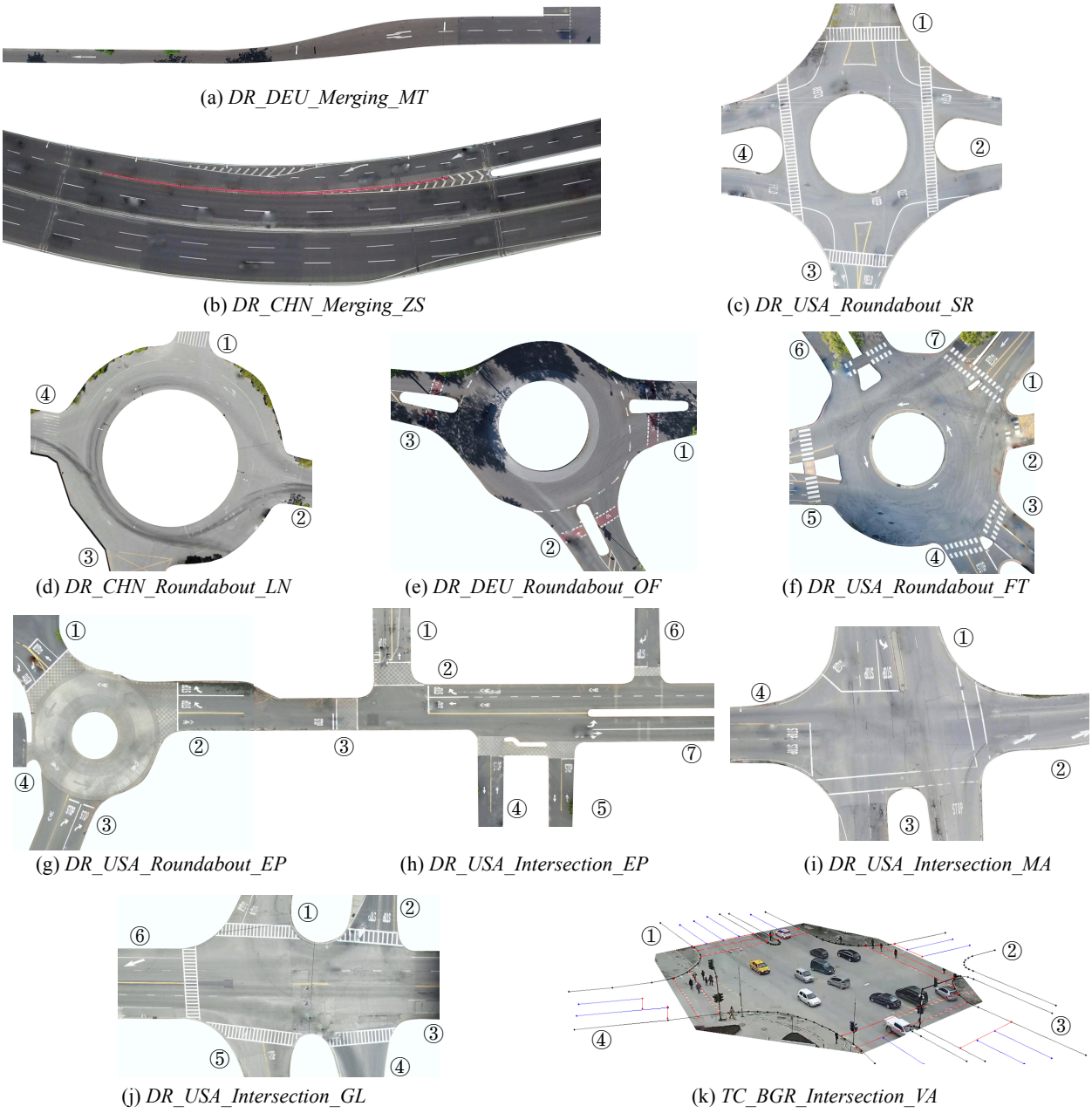
Diversity

Fig. 9.2 illustrates a variety of highly interactive driving scenarios from traffic cameras and drones in our dataset, including zipper merging in a city (Fig. 9.2 (a)), ramp merging and lane change on a highway (Fig. 9.2 (b)), five roundabouts with yield and stop signs (Fig. 9.2 (c) - (g)), several unsignalized intersections with one/two/all-way stops (Fig. 9.2 (h) - (j)), and unprotected left turn at a signalized intersection (Fig. 9.2 (k)). In Fig. 9.2, the first two letters of the names represent the sources of the data (drone as *DR* and traffic camera as *TC*), while next three letters represent the corresponding country and the last two represent the scenario code in the dataset. The numbers in circles denote the branch ID for each scenario.

Fig. 9.2 (b) contains several subscenarios. The subscenario with the upper two lanes (that merge into one finally) is a zipper merging which is similar to the urban counterpart in Fig. 9.2 (a), where vehicles strongly interact with each other. It is also a ramp for the middle two lanes. The subscenario with the lower three lanes (that merge into two finally) is a forced merging and vehicles have to change their lanes.

The roundabout in Fig. 9.2 (f) is an extremely busy 7-way roundabout with one "yield" branch and six "stop" branches. Lots of vehicles enter the roundabout at the same time with intensive interactions and relatively high speeds. The branches of the roundabouts in Fig. 9.2 (c)-(e) are controlled by yield signs, while all branches of the roundabout in Fig. 9.2 (g) are controlled by stop signs.

Figure 9.2 (i) shows an extremely busy all-way-stop intersection with 9 lanes controlled by stop signs. Multiple vehicles are interactively inching to compete. The scenario shown in Fig. 9.2 (j) contains three branches (Branch 1, 2, 5) controlled by stop signs, while vehicles from Branch 3 and 6 have the right-of-way (RoW). Lots of vehicles are entering the intersections from all branches (except Branch 4), and vehicles holding RoW on the straight road are with relatively high speed. A busy all-way-stop T-intersection is shown in Fig. 9.2 (h), while three other branches (Branch 4-6) are also controlled by stop signs.



(a) *DR_DEU_Merging_MT*

(b) *DR_CHN_Merging_ZS*

(c) *DR_USA_Roundabout_SR*

(d) *DR_CHN_Roundabout_LN*

(e) *DR_DEU_Roundabout_OF*

(f) *DR_USA_Roundabout_FT*

(g) *DR_USA_Roundabout_EP*

(h) *DR_USA_Intersection_EP*

(i) *DR_USA_Intersection_MA*

(j) *DR_USA_Intersection_GL*

(k) *TC_BGR_Intersection_VA*

Figure 9.2: A variety of highly interactive driving scenarios recorded by drones in the dataset, including: (a) urban merging, (b) highway ramp merging and lane change, (c)-(g) five roundabouts, and (h)-(j) unsignalized intersections, and (k) unprotected left turn at a signalized intersection.

Internationality

The motion data was collected from three continents (North America, Asia and Europe). Motion data collected by drones are from four countries, namely, the US, China, Germany and Bulgaria, as indicated in the names of the scenarios (*USA/CHN/DEU/BGR*). Vehicles in all these countries are driven on the right-hand side of the road. However, driving culture in these countries is with remarkable distinctions.

We provide motion data from three roundabouts with similar traffic rules, namely, *SR* from the US, *OF* from Germany and *LN* from China. All the three roundabouts do not have stop signs, and the nominal traffic rule is that the vehicles entering the roundabout should yield the ones which is already in the roundabout.

We also provide motion data from two zipper merging scenarios, those are, *MT* from Germany and *ZS* from China (the upper two lanes in Fig. 9.2 (b)). Although *MT* is urban road and *ZS* is the entrance of highway, the "zipper" rule remains the same, and the speeds are similar when the traffic is heavy.

Complexity

In addition to regular driving behavior such as car-following, lane change, stop and left/right/U-turn, our dataset emphasizes highly interactive and complex driving behavior with cooperative and adversarial motions of the vehicles. By carefully choosing the locations and corresponding rush hours for the data collection, we were able to gather large amounts of strong interactions within relative short period of time. Strongly interactive pairs of vehicles can even appear every few seconds from time to time for scenarios such as the ramp in *ZS*, the entrance branches in *FT*, the all-way-stop intersections in *EP* and *MA* as well as the two-way-stop intersection in *GL*.

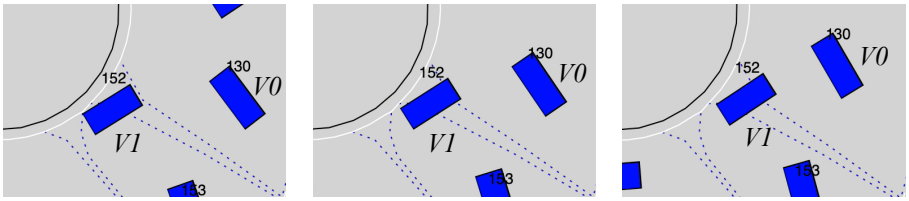


Figure 9.3: A sequence of images of a violation for the right-of-way in a roundabout in the proposed dataset.

Moreover, aggressive or irrational behavior can often be found due to inexplicit nominal or practical RoW. Vehicles may arrive at the stop bars almost at the same time and drivers may negotiate with each other by inching or even accelerating in *MA* and *EP*. The traffic in *FT* and *GL* can be very busy and it may take even minutes for the vehicle without nominal RoW to enter and pass, making the driver impatient. Also, there may be a queue of vehicles

waiting behind and even honking to put social pressures to the one in the front of queue. Although there are explicit traffic rules on who goes first for roundabouts or 2-way-stop intersections, vehicles without nominal RoW may be aggressive, and vehicles with nominal RoW are mostly aware of such potential violations and are ready to react. For example, *V0* in Fig. 9.3 was entering the roundabout in *FT* from Branch 3, while *V1* was in the roundabout holding the RoW. However, *V0* violated the rule and forced *V1* to stop and yield.

Those factors significantly increase the complexity of the motions in the dataset and bring forward lots of challenging but valuable research topics for the community.

Criticality

As discussed in Section 9.3, vehicles holding the nominal RoW (in the roundabout of *FT* or on the straight road of *GL*) may often encounter slight violations from vehicles without nominal RoW (entering the roundabout or intersection from branches controlled by stop signs). Moreover, the vehicles holding the RoW may have relatively high speed (40 km/h or even higher). Therefore, critical situations can be observed in the dataset where time-to-collision-point (TTCP) can be extremely low. A slight collision can even be found in the dataset.

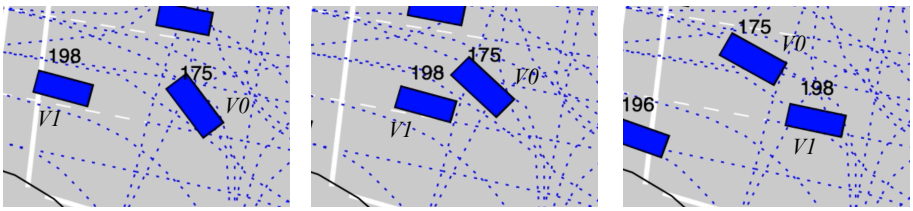


Figure 9.4: A sequence of images of a near-collision case in the proposed dataset.

Fig. 9.4 shows a near-collision case in *GL*. *V0* was making a left turn from Branch 5 (with a stop sign) to Branch 6, while *V1* (with the RoW) was going straight forward from Branch 6 to Branch 3 with a relatively high speed. *V1* had to execute emergency swerve to avoid the collision with *V0*, which was very dangerous.

Semantic Map

Map information is crucial for behavior-related research areas. The information required is twofold. The basic requirement is the physical layer containing a set of points or curves representing curbs, road markings (lane markings, stop bars, etc.) and other key features. In addition to the physical layer, semantic information is also necessary, which includes but is not limited to, 1) reference paths, 2) lanelets as well as their connections and turn directions,

3) traffic rules and RoW associated, etc. Moreover, such information needs to be organized with consistent format and toolkit to facilitate the users when utilizing the map. All the aforementioned requirements are met in our dataset, and more detailed information on map construction can be found in Section 9.4.

9.4 Construction of Motion Data and Maps

In this section, we will discuss the pipeline for constructing the motion data from both drones and traffic cameras, as well as the corresponding semantic maps.

Motions from Drone Data

We used drones such as DJI Mavic 2 and DJI Phantom 4 to collect the raw video data. The raw videos were 4K (3840x2160) by 30 Hz. We downsampled the video to 10 Hz and process the data. The processed results are partially illustrated in Fig. 9.1. The bounding boxes are very accurate and the paths are smooth after going through out processing pipeline with the following three steps.

- *Video stabilization and alignment:* Due to gradual or sudden drift and rotation of drones, the collected videos need to be stabilized via video stabilization algorithms with transformation estimator. Also, similarity transformation is applied to project all the frames to the first one and aligned with the map.
- *Detection:* In order to obtain accurate bounding boxes of the moving obstacles, Faster R-CNN [100] is applied. The boxes are highly accurate, and very few inaccurate detections are rectified manually.
- *Data association, tracking and smoothing:* Kalman filter is applied for data association and tracking. To obtain smooth motions of the vehicles, a Rauch-Tung-Striebel (RTS) smoother [115] is also incorporated.

Motions from Traffic Camera Data

The data processing pipeline for motions from traffic camera data mainly contains the following steps, and more details, including the camera parameter estimation, can be found in [23].

- *Detection:* To detect vehicles and pedestrians in each frame, we use a state-of-the-art object detector [46], which provides detections with 2D bounding box, instance mask and instance type.

Table 9.3: Summary of the dataset.

Scenarios	Locations	Video length (min)	number of vehicles	Total video length (min)	Total number of vehicles
roundabout	USA_Roundabout_SR	40.90	965	365.1	10479
	CHN_Roundabout_LN	24.24	227		
	DEU_Roundabout_OF	55.04	1083		
	USA_Roundabout_FT	207.62	7496		
	USA_Roundabout_EP	37.30	708		
unsignalized intersection	USA_Intersection_EP	66.53	1367	433.33	14867
	USA_Intersection_MA	107.37	2982		
	USA_Intersection_GL	259.43	10518		
merging and lane change	DEU_Merging_MT	37.93	574	132.55	10933
	CHN_Merging_ZS	94.62	10359		
signalized intersection	TC_Intersection_VA	60	3775	60	3775

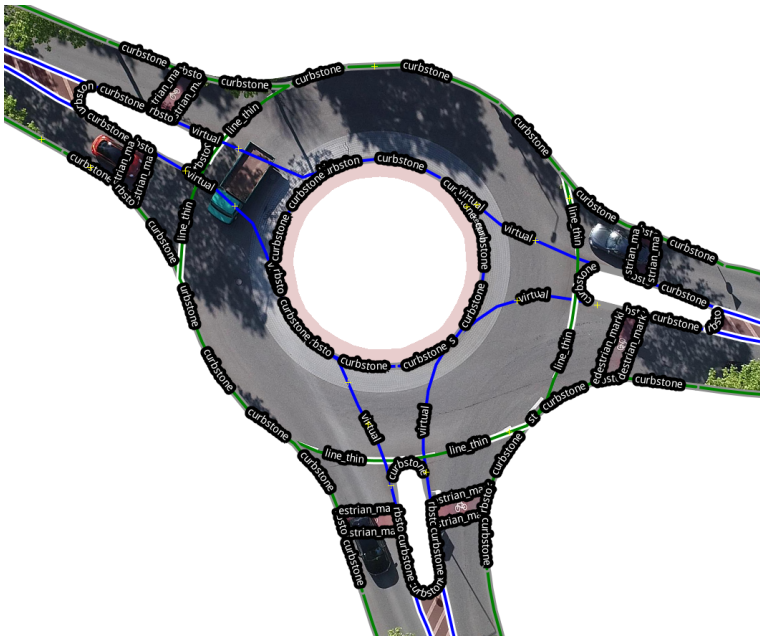


Figure 9.5: An exemplary physical layer of a lanelet2 map [94].

- *Data association:* Detections are grouped into tracks using a combination of an Intersection-over-Union [14] tracker which associates detections with high mask overlap in successive frames, and a visual tracker [80] to compensate for miss detections.
- *Tracking and smoothing:* Once detections are grouped into tracks, trajectories on the ground plane are estimated using a RTS smoother. For the observation model, we use a pin-hole camera model [19]. This allows to incorporate measurements and uncertainty directly in pixels, capturing the uncertainty due to the resolution, position and orientation of the camera. For vehicles, the RTS smoother uses a bicycle model [95] as process model, allowing to capture the kinematics constraints of vehicles.

Construction of the High Definition Maps

As public roads are structured environments, the particular road layout of a certain area strongly affects the motion of all traffic participants. The structure for vehicles mostly starts by subdividing the road into lanes, and later combining them to create junctions, roundabouts, on ramps and so on. Further, movement within this structured area is guided by traffic rules, such as speed limits or prioritizing one road over another. In order to model such coherence, simply mapping center-lines of all lanes is not sufficient anymore.

Thus, in order to allow for a thorough analysis of the recorded trajectories, we provide centimeter-accurate high definition maps in the lanelet2 format [94]. Within lanelet2, the physical layer of the road network, such as road borders, lane markings and traffic signs is

stored. An exemplary physical layer is visualized in Figure 9.5. From this layer, atomic lane elements, called *lanelets*, are created. They describe the course of the lane and form the basis for so called regulatory elements, which determine traffic regulations such as the right of way or the speed limit.

When used alongside the recorded trajectories, these lanelet2 maps facilitate the reasoning about why some vehicles decelerate while approaching a junction, or why others do not, depending on the right of way but also on the presence of other traffic participants that potentially interact.

9.5 Statistics of the Dataset

Scenarios and Vehicle Density

The dataset contains motion data collected in four categories of scenarios: roundabout, unsignalized intersection, signalized intersection, merging and lane change, as shown in Fig. 9.2. A detailed summary of the dataset is listed in Table 9.3. In the roundabout scenarios, 10479 trajectories of vehicles from five different locations were recorded for around 365 minutes. Similarly, in the unsignalized intersection scenarios, three locations were included and 14867 trajectories were collected for around 433 minutes. In the merging and lane change scenarios, 10933 trajectories were recorded at two locations for around 133 minutes. Finally, one location was selected for the signalized intersection, which provided 3775 trajectories for around 60 minutes.

Metrics for Interactive Behavior Identification

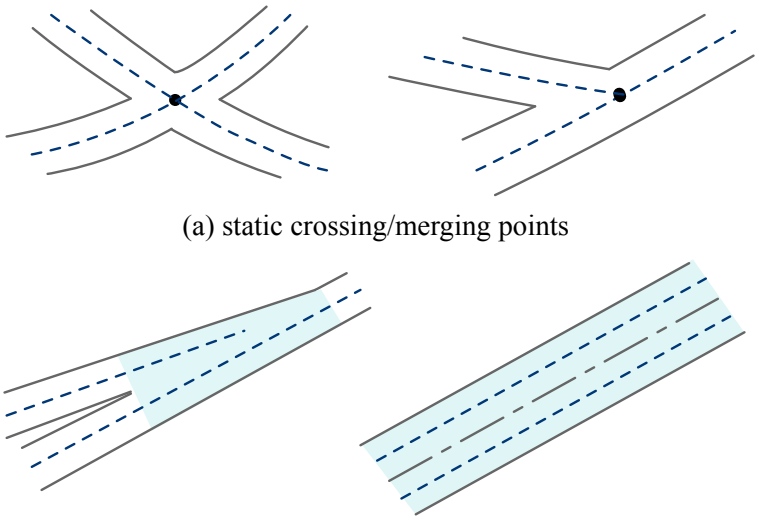
To represent the density of the interactive behavior of the proposed dataset, we use the metric - number of interaction pairs per vehicle (IPV) as in proposed in [136]. To calculate the IPV, a set of rules were proposed in [136] to extract the interactive behavior under different spatial representations of vehicle paths. The set of rules and metric are briefly reviewed below.

1. Minimum time-to-conflict-point difference ($\triangle TTCP_{\min}$): $\triangle TTCP_{\min}$ is a metric to describe the relative states of two moving vehicles in a scenario where the paths of the two vehicles share a conflict point but without any forced stop. As shown in Fig. 9.6, such vehicle paths include two categories: (1) paths with static crossing or merging points such as intersections (Fig. 9.6 (a)-(b)), and (2) paths with dynamic crossing or merging points such as ramping and lane-changing, as shown in Fig. 9.6 (c)-(d). In such scenarios, merging can happen anywhere in the shaded area. We define $\triangle TTCP_{\min}$ as

$$\begin{aligned} \triangle TTCP_{\min} &= \min_{t\in[T_{\text{start}},T_{\text{end}}]} \triangle TTCP^t \\ &= \min_{t\in[T_{\text{start}},T_{\text{end}}]} (TTCP_1^t - TTCP_2^t) \end{aligned} \tag{9.1}$$

where $TTCP_i^t = \triangle d_i^t / v_i^t, i = 1, 2$ is the traveling time to the conflict point of each vehicle in the interactive pairs. v_i^t and $\triangle d_i^t$ are, respectively, the speed of the i-th vehicle and its distance to the conflict point along the path at time t. For the scenarios with dynamic merging points, we use the actual merging points of the vehicle trajectories as the conflict points. In (9.1), T_{start} and T_{end} are set to be long enough to cover the interaction period between vehicles. If $TTCP_{\min} \leq 3\,\text{s}$, then it is defined that interaction exists.

2. Waiting Period (WP): WP is a metric for vehicles with forced stops along their paths. In [136], the default waiting period at stops was set as 3 s, and the behavior deviation from the default one was used as an indicator of the interactivity, i.e., interaction exists when WP >3 s.



(a) static crossing/merging points

(b) dynamic crossing/merging points

Figure 9.6: Geometry of different interactive paths. In (a), the crossing/merging points between two paths are static and fixed, while in (b), the crossing/merging points are dynamic.

Distribution of Interactivity

Based on the set of rules, there are 13375 interactive pairs of vehicles in the proposed dataset. We compare the interactivity among three datasets: the proposed INTERACTION dataset, the highD dataset, and the NGSIM dataset. Results are shown in Fig. 9.7, where the x-axis represents the length of $\triangle TTCP_{\min}$ in seconds, and the y-axis are the number of

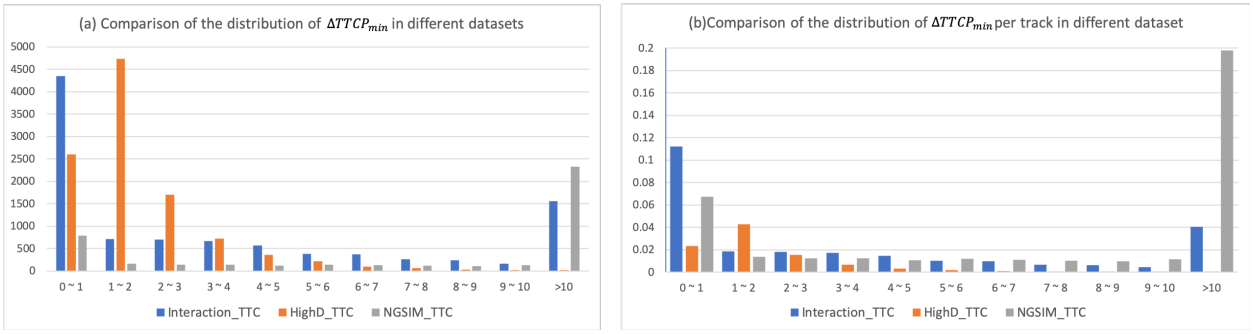


Figure 9.7: Distribution of the $\triangle TTCP_{\text{min}}$ in three vehicle motion datasets: the proposed INTERACTION dataset, the HighD dataset and the NGSIM dataset.

vehicles (Fig. 9.7 (a)) and the density of vehicles[1] (Fig. 9.7 (b)), respectively. We can see that the INTERACTION dataset contains more intensive interactions with $\triangle TTCP_{\text{min}} \leq 1$s.

We also summarized the distributions of $\triangle TTCP_{\text{min}}$ and WP of all vehicles in the dataset over different driving scenarios. The results are shown in Fig. 9.8. Similarly, the x-axis represents the length of $\triangle TTCP_{\text{min}}$ and WP in seconds, and the y-axis is the density of vehicles in each scenario. We can see that the dataset contains highly interactive trajectories with a high density of $\triangle TTCP_{\text{min}} \leq 1$ s, and WP greater than 3 s.

9.6 Utilization Examples

The proposed dataset is intended to facilitate researches related to driving behavior, as mentioned in Section 9.1. In this section, we provide several utilization examples of the proposed dataset, including motion/trajectory prediction, imitation learning, motion planning and validation, motion clustering and representation, interaction extraction and human-like behavior generation.

Motion Prediction and Behavior Analysis

Motion/trajectory prediction is of vital importance for autonomous vehicles, particularly in situations where intensive interaction happens. To obtain an accurate probabilistic prediction model of vehicle motion, both learning- and planning-based approaches have been extensively explored. By providing high-density interactive trajectories along with HD semantic maps, the proposed dataset can be used for both approaches.

For instance, [81] proposed a deep latent variable model based on Wasserstein auto-encoder (WAE) to improve the interpretability. It incorporated the structure of recurrent

[1]The density is given by:
$$\text{density} = \frac{\text{number of vehicles with particular } \triangle TTCP_{\text{min}}}{\text{total number of vehicles in the dataset}}.$$

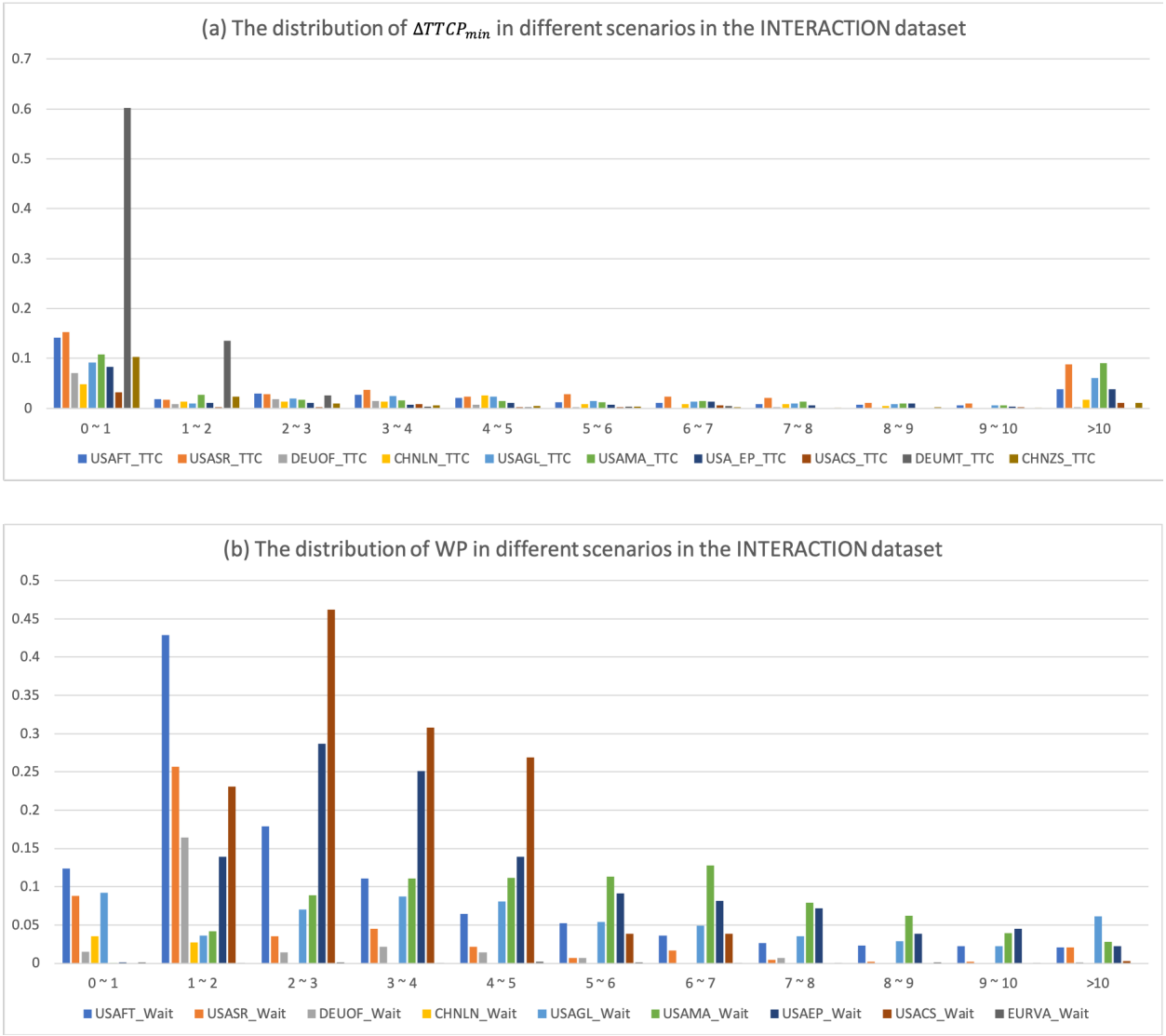


Figure 9.8: Distribution of the $\triangle TTCP_{\text{min}}$, and WP across different locations and scenarios in the dataset.

Table 9.4: Comparisons of prediction accuracy from [81].

Methods	**features**	**RMSE**	**MAE**
WAE-based approach	x	**0.013/0.011**	**0.046/0.035**
	y	0.006/0.014	**0.019/0.041**
	ψ	0.006/**0.008**	**0.018/0.042**
VAE	x	0.018/0.016	0.25/0.22
	y	**0.006/0.003**	0.14/0.22
	ψ	0.006/**0.008**	0.13/0.21
Auto-encoder	x	0.315/0.044	1.026/0.315
	y	0.057/0.141	0.182/0.479
	ψ	0.011/0.066	0.037/0.078
GAN	x	0.024/0.020	0.324/0.273
	y	0.007/0.017	0.188/0.241
	ψ	**0.005**/0.048	0.107/0.286

neural network with vehicle kinematic model such that the output can be constrained. The motion data in FT was utilized to train and test the model in comparison with other state-of-the-art models such as variational auto-encoder (VAE), auto-encoder, and generative adversarial network (GAN). Quantitative results shown in Section 9.6 demonstrated that the proposed WAE-based method can outperform other state-of-the-art models, when comparing the root mean square error (RMSE) and mean absolute error (MAE) of the prediction for position and yaw angle.

On the other hand, [49] took advantage of the HD semantic maps and combined the learning-based and the planning-based prediction methods. A deep learning model based on conditional variational auto-encoder (CVAE) and an optimal planning framework based on inverse reinforcement learning are dynamically combined to predict both irrational and rational behavior of the vehicles. Benefiting from the the HD semantic information, features for the deep learning model were defined in Frenet frame, which generated much better prediction performance in terms of generalization. Some exemplar results are given in Fig. 9.9.

Imitation Learning

The driving behavior in the proposed dataset can also be used for imitation learning which directly imitates how human drive in complicated scenarios. We extended the fast integrated learning and control framework proposed in [117] in the FT roundabout scenario. As shown in Fig. 9.10, both the semantic HD map information and the states of surrounding

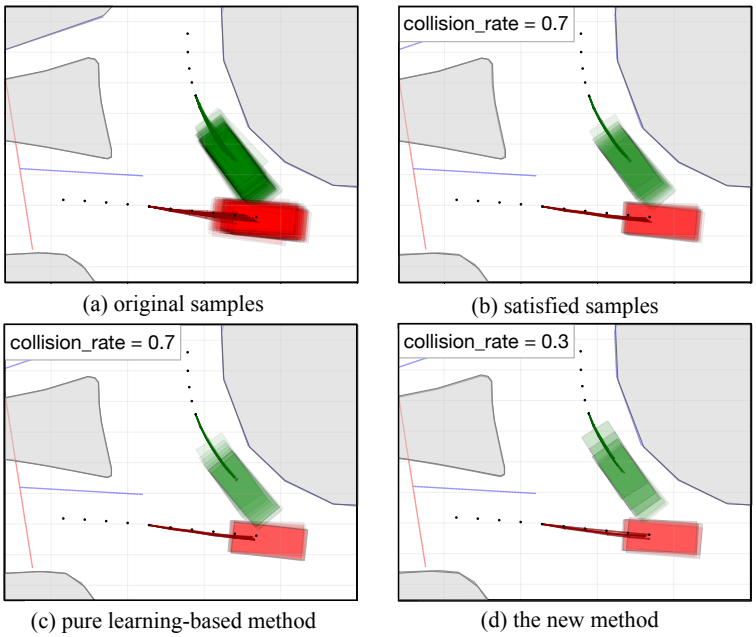


Figure 9.9: Some exemplar prediction results from [49].

vehicles (the red boxes) were included as the features. The grey box represents the current position of the ego vehicle. The green boxes and blue boxes, respectively, are the ground truth future positions and generated future positions of the ego vehicle via the imitation network.

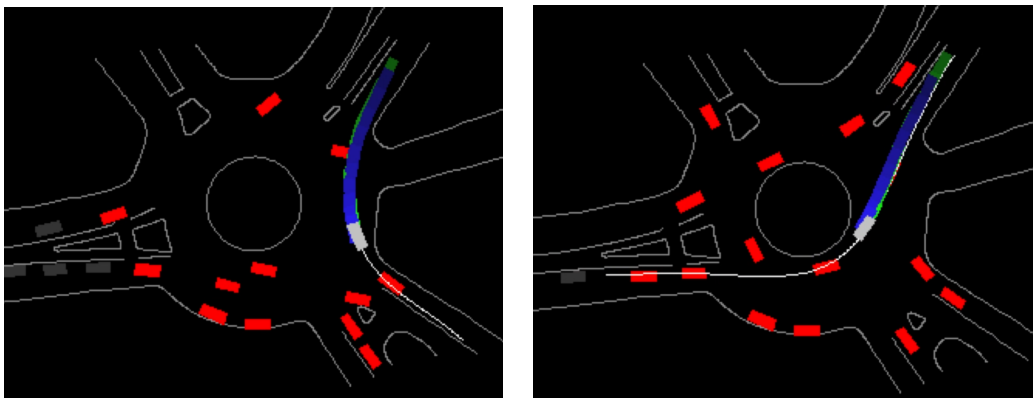


Figure 9.10: Two examples of the imitation learning results by employing the method in [117].

Validation of Decision and Planning

Besides motion prediction and imitation, the motion data and maps in the dataset can also be used for testing different decision making and motion planning algorithms. The data-replay motions in the dataset are more suitable to test the performances of the decision-maker and planner when the motions of surrounding entities are independent of the ego motions. For example, the motion of the ego vehicle may not effect others when it does not have the RoW, or it has the RoW but others violate the rules or ignore the ego motion. We have shown several examples of decision and planning validation in Chapter 5 and Chapter 7.

Motion Clustering and Representation Learning

The X-means algorithm [92] was employed to cluster the trajectories and obtain motion patterns with results shown in Fig. 9.11. We constructed a feature space with vehicle motions in Frenét Frame based on map information. Fig. 9.11 (a) shows the clustered trajectory segments in different colors with the map. Fig. 9.11 (b) and (d) demonstrate the cluster results with longitudinal positions and speeds of the two interacting vehicles as the coordinates. The clustering results with the first and second components of principle component analysis (PCA) for the feature space are shown in Fig. 9.11 (c). In the figures we can see that different interactive motions are separated and similar ones are clustered, which are desirable results to obtain motion patterns.

Extraction of Interactive Agents and Trajectories.

The proposed dataset can also be used to learn the interaction relationships between agents. We implemented the learning method and network structure proposed in [114] to extract the interaction frames of two agents. Some example results are given in Fig. 9.12, where Fig. 9.12 (a) and (b) provide one exemplar pair of interacting cars in the FT scenario, while Fig. 9.12 (c) and (d) represent another pair. In Fig. 9.12 (a) and (c), the paths of both of the interacting cars are provided, and in Fig. 9.12 (b) and (d), the trajectories along longitudinal directions are shown. We can see that the extracted interaction frames (purple circle) align quite well with the ground truth frames (blue star).

Human-like Decision and Behavior Generation

We can also learn decision-making models that generate human-like decisions and behaviors with the proposed dataset. In [119], an interpretable human behavior model was proposed based on the cumulative prospect theory (CPT). As a non-expected utility theory, CPT can well explain some systematically biased or "irrational" behavior/decisions of human that cannot be explained by the expected utility theory. Parameters of three different models were learned and tested using the data in the FT roundabout scenario: a predefined model based on time-to-collision-point (TTCP), a learning-based model based on

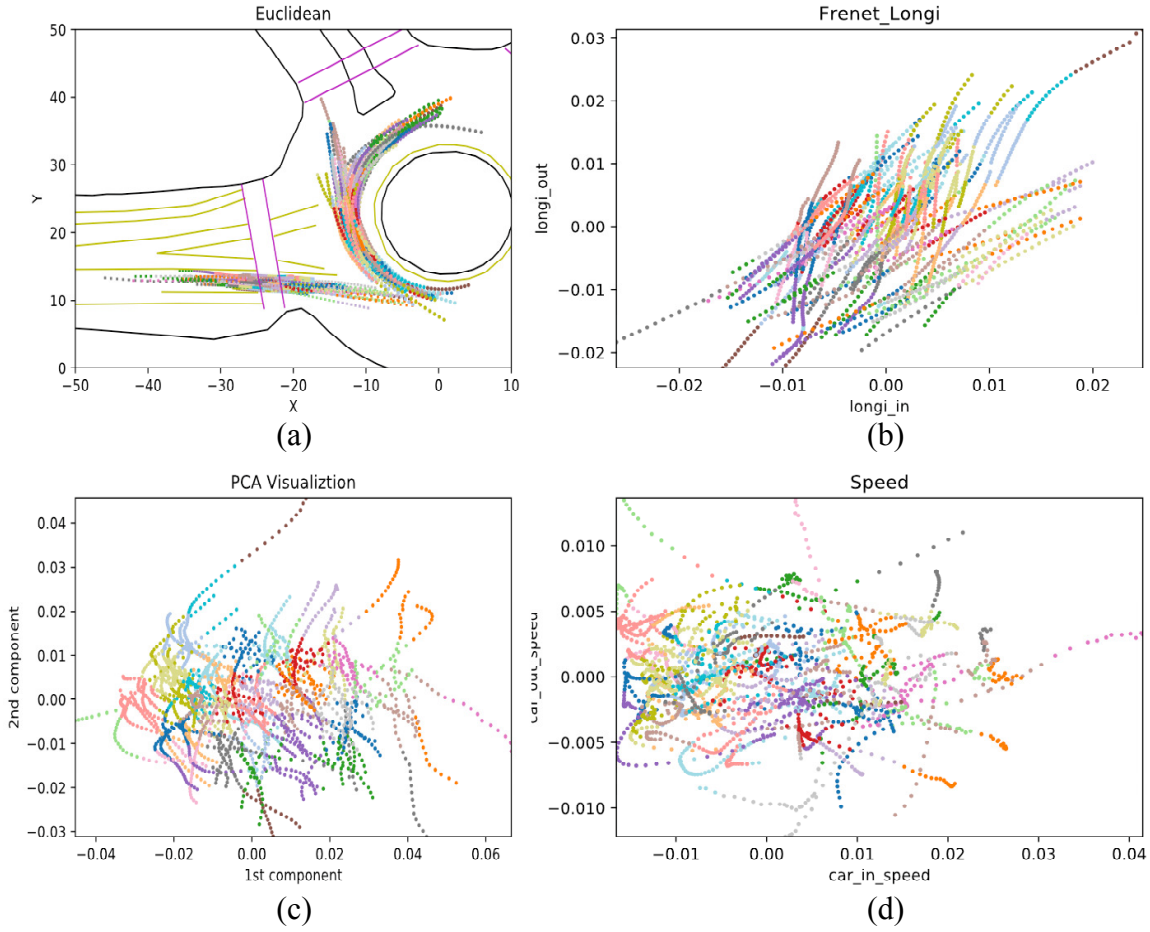
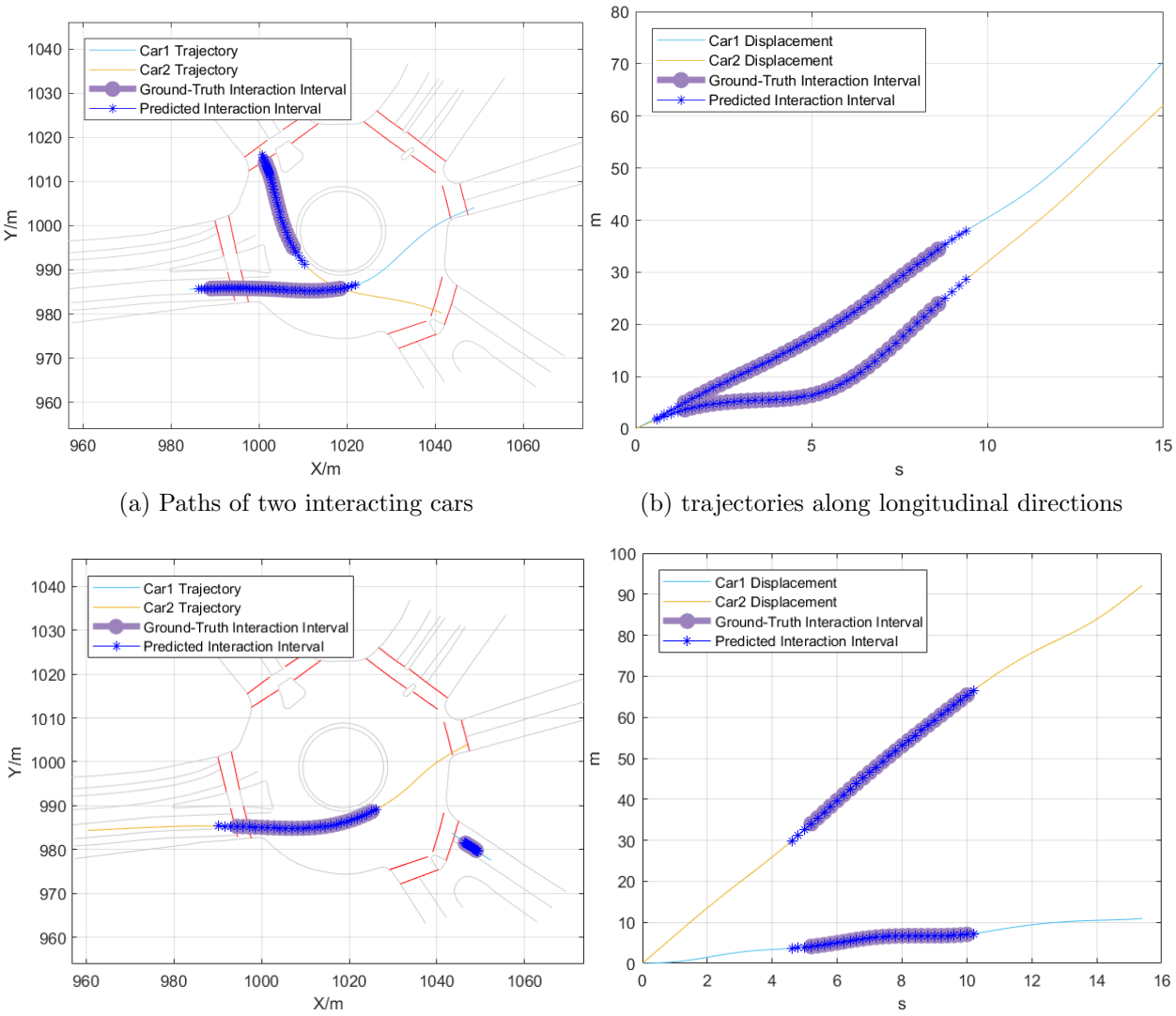


Figure 9.11: Results of X-means [92] motion clustering using the proposed dataset.

neural networks, and the proposed CPT-based model. The results (Fig. 9.13) showed that the CPT-based model outperformed the TTCP model and achieved similar performance as the learning-based model with much less training data and better interpretability.

9.7 Chapter Summary

In this chapter, we presented a motion dataset in a variety of highly interactive driving scenarios from the US, Germany, China and other countries, including signalized/unsignalized intersections, roundabouts, ramp merging and lane change from cities and highway. Complex interactive motions were captured, featuring inexplicit right-of-way, relatively unstructured roads, as well as aggressive and irrational behavior caused by impatience and social pressure.



(a) Paths of two interacting cars

(b) trajectories along longitudinal directions

(c) Paths of two interacting cars

(d) trajectories along longitudinal directions

Figure 9.12: Two examples of the extracted interaction pairs by implementing the learning method and network structure in [114].

Critical (near-collision and slight-collision) situations can be found in the dataset. We also included high-definition (HD) maps with semantic information for all scenarios in our dataset. The data was recorded from drones and traffic cameras and the data processing pipeline was briefly described. Our map-aided dataset with diversity, internationality, complexity and criticality of scenarios and behavior can significantly facilitate driving-behavior-related research such as motion prediction, imitation learning, decision-making and planning, representation learning, interaction extraction, and human-like behavior generation, etc. Results

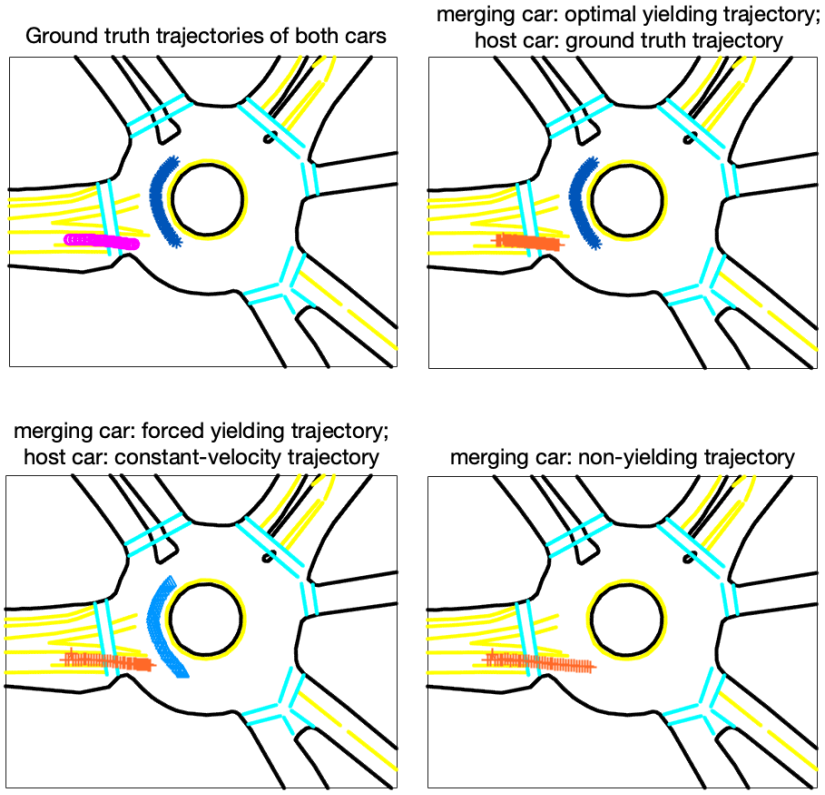


Figure 9.13: Results of interpretable human behavior model based on the cumulative prospect theory (CPT) [119] using the proposed dataset.

from various kinds of methods of these research areas were demonstrated utilizing the proposed dataset.

Chapter 10

Conclusions

This dissertation addressed the fundamental aspects as well as corresponding algorithms for interactive prediction and planning, including deterministic planning algorithms (Part I), design of decision and planning under uncertainty (Part II), unified formulation and representation of various interactive prediction algorithms and their interface with the decision and planning algorithms designed (Part III), and fatality-aware prediction evaluation as well as the motion dataset with highly interactive scenarios and behavior (Part IV).

- Generic environmental representation for various scenarios was constructed by spatially partitioning the spatiotemporal domain and decomposing them into three topological elements. Then the corresponding planning algorithm was designed by combining A* search and quadratic programming given deterministic predictions of a number of other road participants (Chapter 2). Deterministic planners based on imitation learning were also designed by incorporating the hard constraints on safety and feasibility of an optimization-based planner into the training loss of the policy. The policy was able to generate safe and feasible trajectories in highly constrained scenarios (Chapter 3).

- An integrated decision and planning framework under uncertainty from prediction and perception modules was proposed by combining decision network and an arbitrary planner with cost and constraints. A non-conservatively defensive driving strategy (NCDS) was achieved with an optimization-based planner, which is defensive to even potential violations of others, but not overcautious to threats of low probabilities (Chapter 4). The geometrical intuition of the proposed NCDS was provided. Experiments were conducted on real vehicles with real-world scenarios and motions for the proposed NCDS with graph-search-based and sample-based planners in real time (Chapter 5).

- A unified framework for the formulation and representation of interactive prediction was proposed, which can take into account different paradigms of prediction methods such as planning-based prediction (e.g. inverse reinforcement learning), as well as prediction based on probabilistic graphical model and deep neural networks (Chapter 6). Then the aforementioned three paradigms of interactive prediction methods were

introduced, and integrated with a proposed interface for interactive prediction and planning, with NCDS and sample-based planner (Chapter 7).

- By utilizing the proposed unified framework for interactive prediction, a fatality-aware benchmark for the aforementioned three paradigms of predictors was proposed by considering the consequence of using the prediction results. Predictor based on neural network outperformed others in terms of data approximation metrics, while planning-based predictor achieved the best performance when fatality was introduced into the metrics (Chapter 8). Finally, a motion dataset with highly interactive driving scenarios and behavior from several countries was constructed, and density of interaction was defined and calculated to compare different datasets. The dataset has been utilized by various behavior-related research fields such as prediction, planning, imitation learning and behavior modeling, and is inspiring new research fields such as representation learning, interaction extraction, scenario generation, and so on (Chapter 9).

Bibliography

[1] Kazi Iftekhar Ahmed. "Modeling drivers' acceleration and lane changing behavior". PhD thesis. Massachusetts Institute of Technology, 1999.

[2] Vassili Alexiadis et al. "The Next Generation Simulation Program". English. In: *Institute of Transportation Engineers. ITE Journal; Washington* 74.8 (Aug. 2004), pp. 22–26. ISSN: 01628178.

[3] Florent Altché and Arnaud de La Fortelle. "An LSTM network for highway trajectory prediction". In: *Intelligent Transportation Systems (ITSC), 2017 IEEE 20th International Conference on*. IEEE. 2017, pp. 353–359.

[4] Florent Altché, Philip Polack, and Arnaud de La Fortelle. "A Simple Dynamic Model for Aggressive, Near-Limits Trajectory Planning". In: *2017 IEEE Intelligent Vehicles Symposium (IV)*. June 2017.

[5] Florent Altché, Philip Polack, and Arnaud de La Fortelle. "High-Speed Trajectory Planning for Autonomous Vehicles Using a Simple Dynamic Model". In: *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. Oct. 2017.

[6] M. Althoff, O. Stursberg, and M. Buss. "Model-Based Probabilistic Collision Detection in Autonomous Driving". In: *IEEE Transactions on Intelligent Transportation Systems* 10.2 (June 2009), pp. 299–310. ISSN: 1524-9050. DOI: 10.1109/TITS.2009.2018966.

[7] Georges Aoude et al. "Mobile Agent Trajectory Prediction using Bayesian Nonparametric Reachability Trees". In: American Institute of Aeronautics and Astronautics, Mar. 2011.

[8] S. Atev, G. Miller, and N. P. Papanikolopoulos. "Clustering of Vehicle Trajectories". In: *IEEE Transactions on Intelligent Transportation Systems* 11.3 (Sept. 2010), pp. 647–657. ISSN: 1524-9050. DOI: 10.1109/TITS.2010.2048101.

[9] M. Bahram et al. "A Combined Model- and Learning-Based Framework for Interaction-Aware Maneuver Prediction". In: *IEEE Transactions on Intelligent Transportation Systems* 17.6 (June 2016), pp. 1538–1550. ISSN: 1524-9050. DOI: 10.1109/TITS.2015.2506642.

[10] Haoyu Bai et al. "Intention-aware online POMDP planning for autonomous driving in a crowd". In: *2015 ieee international conference on robotics and automation (icra)*. IEEE. 2015, pp. 454–460.

[11] P. Bender et al. "The combinatorial aspect of motion planning: Maneuver variants in structured environments". In: *2015 IEEE Intelligent Vehicles Symposium (IV)*. June 2015, pp. 1386–1392.

[12] P. C. Besse et al. "Review and Perspective for Distance-Based Clustering of Vehicle Trajectories". In: *IEEE Transactions on Intelligent Transportation Systems* 17.11 (Nov. 2016), pp. 3306–3317. ISSN: 1524-9050. DOI: 10.1109/TITS.2016.2547641.

[13] Christopher M Bishop. "Mixture density networks". In: (1994).

[14] Erik Bochinski, Volker Eiselein, and Thomas Sikora. "High-speed tracking-by-detection without using image information". In: *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE. 2017, pp. 1–6.

[15] Mariusz Bojarski et al. "End to End Learning for Self-Driving Cars". In: *arXiv preprint arXiv:1604.07316* (2016).

[16] Sebastian Brechtel, Tobias Gindele, and Rüdiger Dillmann. "Probabilistic decision-making under uncertainty for autonomous driving using continuous POMDPs". In: *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. IEEE. 2014, pp. 392–399.

[17] Glenn W Brier. "Verification of forecasts expressed in terms of probability". In: *Monthey Weather Review* 78.1 (1950), pp. 1–3.

[18] A Broggi et al. "PROUD-Public road urban driverless test: Architecture and results". In: *2014 IEEE Intelligent Vehicles Symposium Proceedings*. June 2014, pp. 648–654.

[19] Duane C. Brown. "Close-range camera calibration". In: *PHOTOGRAMMETRIC ENGINEERING* 37.8 (1971), pp. 855–866.

[20] Ming-Fang Chang et al. "Argoverse: 3D Tracking and Forecasting With Rich Maps". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 8748–8757.

[21] Chenyi Chen et al. "DeepDriving: Learning Affordance for Direct Perception in Autonomous Driving". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, pp. 2722–2730.

[22] Jianyu Chen et al. "Continuous Decision Making for On-Road Autonomous Driving under Uncertain and Interactive Environments". In: *IEEE Intelligent Vehicles Symposium (IV)*. 2018, pp. 1651–1658.

[23] Aubrey Clausse, Salma Benslimane, and Arnaud De La Fortelle. "Large-Scale Extraction of Accurate Vehicle Trajectories for Driving Behavior Learning". In: *30th IEEE Intelligent Vehicles Symposium (IV)* (2019).

[24] Benjamin Coifman and Lizhe Li. "A critical evaluation of the Next Generation Simulation (NGSIM) vehicle trajectory dataset". In: *Transportation Research Part B: Methodological* 105 (Nov. 2017), pp. 362–377.

[25] F. Damerow, S. Klingelschmitt, and J. Eggert. "Spatio-temporal trajectory similarity and its application to predicting lack of interaction in traffic situations". In: *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. Nov. 2016, pp. 2512–2519. DOI: 10.1109/ITSC.2016.7795960.

[26] Zachary R. Doerzaph et al. *Live stop-controlled intersection data collection*. Tech. rep. 2007.

[27] Dmitri Dolgov et al. "Path Planning for Autonomous Vehicles in Unknown Semi-structured Environments". In: *The International Journal of Robotics Research* 29.5 (Apr. 2010), pp. 485–501.

[28] C. Dong, J. M. Dolan, and B. Litkouhi. "Intention estimation for ramp merging control in autonomous driving". In: *2017 IEEE Intelligent Vehicles Symposium (IV)*. June 2017, pp. 1584–1589.

[29] Chiyu Dong, John M Dolan, and Bakhtiar Litkouhi. "Interactive ramp merging planning in autonomous driving: Multi-Merging leading PGM (MML-PGM)". In: *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC2017)*. 2017.

[30] K. Driggs-Campbell, V. Govindarajan, and R. Bajcsy. "Integrating Intuitive Driver Models in Autonomous Planning for Interactive Maneuvers". In: *IEEE Transactions on Intelligent Transportation Systems* 18.12 (Dec. 2017), pp. 3461–3472.

[31] Dennis Fassbender, Benjamin C. Heinrich, and Hans-Joachim Wuensche. "Motion planning for autonomous vehicles in highly constrained urban environments". In: *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, 2016, pp. 4708–4713.

[32] Katerina Fragkiadaki et al. "Learning Visual Predictive Models of Physics for Playing Billiards". In: *arXiv:1511.07404 [cs]* (Nov. 2015).

[33] J. Funke et al. "Up to the limits: Autonomous Audi TTS". In: *2012 IEEE Intelligent Vehicles Symposium (IV)*. June 2012, pp. 541–547. DOI: 10.1109/IVS.2012.6232212.

[34] E. Galceran, E. Olson, and R. 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)*. Sept. 2015, pp. 3559–3565. DOI: 10.1109/IROS.2015.7353874.

[35] A. R. Geist et al. "Data Collection for Robust End-to-End Lateral Vehicle Control". In: *Proceedings of the ASME 2017 Dynamic Systems and Control Conference*. 2017.

[36] Xinli Geng et al. "A Scenario-Adaptive Driving Behavior Prediction Approach to Urban Autonomous Driving". In: *Applied Sciences* 7.4 (Apr. 2017), p. 426.

[37] D. González et al. "A Review of Motion Planning Techniques for Automated Vehicles". In: *IEEE Transactions on Intelligent Transportation Systems* 17.4 (Apr. 2016), pp. 1135–1145. ISSN: 1524-9050. DOI: 10.1109/TITS.2015.2498841.

[38] *Google Self-Driving Car Project Monthly Report*. May 2015.

[39] *Google Self-Driving Car Project Monthly Report*. Feb. 2016.

[40] J. Grégoire, S. Bonnabel, and A. de La Fortelle. "Priority-based intersection management with kinodynamic constraints". In: *2014 European Control Conference (ECC)*. June 2014, pp. 2902–2907. DOI: 10.1109/ECC.2014.6862377.

[41] Richard Gruner et al. "Spatiotemporal representation of driving scenarios and classification using neural networks". In: *Intelligent Vehicles Symposium (IV), 2017 IEEE*. IEEE, 2017, pp. 1782–1788.

[42] T. Gu and J. M. Dolan. "Toward human-like motion planning in urban environments". In: *2014 IEEE Intelligent Vehicles Symposium Proceedings*. June 2014, pp. 350–355. DOI: 10.1109/IVS.2014.6856493.

[43] Tianyu Gu et al. "Tunable and stable real-time trajectory planning for urban autonomous driving". In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Sept. 2015, pp. 250–256.

[44] C. Guo et al. "Toward Human-like Behavior Generation in Urban Environment Based on Markov Decision Process With Hybrid Potential Maps". In: *2018 IEEE Intelligent Vehicles Symposium (IV)*. June 2018, pp. 2209–2215.

[45] J. Hardy and M. Campbell. "Clustering obstacle predictions to improve contingency planning for autonomous road vehicles in congested environments". In: *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Sept. 2011, pp. 1605–1611. DOI: 10.1109/IROS.2011.6094952.

[46] Kaiming He et al. "Mask R-CNN". In: *2017 IEEE International Conference on Computer Vision (ICCV)* (2017), pp. 2980–2988.

[47] Stefan Hoermann, Daniel Stumper, and Klaus Dietmayer. "Probabilistic long-term prediction for autonomous vehicles". In: *Intelligent Vehicles Symposium (IV), 2017 IEEE*. 2017, pp. 237–243.

[48] Stefan Hoermann et al. "Entering crossroads with blind corners. A safe strategy for autonomous vehicles". In: *Intelligent Vehicles Symposium (IV), 2017 IEEE*. IEEE, 2017, pp. 727–732.

[49] Yeping Hu, Liting Sun, and Masayoshi Tomizuka. "Generic Prediction Architecture Considering both Rational and Irrational Driving Behaviors". In: *2019 22st International Conference on Intelligent Transportation Systems (ITSC), to appear*. 2019.

[50] Yeping Hu, Wei Zhan, and Masayoshi Tomizuka. "A Framework for Probabilistic Generic Traffic Scene Prediction". In: *IEEE 21st International Conference on Intelligent Transportation Systems (ITSC)*. 2018.

[51] Yeping Hu, Wei Zhan, and Masayoshi Tomizuka. "Probabilistic Prediction of Vehicle Semantic Intention and Motion". In: *IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2018.

[52] C. Hubmann, M. Aeberhard, and C. Stiller. "A generic driving strategy for urban environments". In: *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. Nov. 2016, pp. 1010–1016.

[53] Constantin Hubmann et al. "Decision making for autonomous driving considering interaction and uncertain prediction of surrounding vehicles". In: *Intelligent Vehicles Symposium (IV), 2017 IEEE*. IEEE. 2017, pp. 1671–1678.

[54] A. Jahangiri, H. A. Rakha, and T. A. Dingus. "Adopting Machine Learning Methods to Predict Red-light Running Violations". In: *2015 IEEE 18th International Conference on Intelligent Transportation Systems*. 2015 IEEE 18th International Conference on Intelligent Transportation Systems. Sept. 2015, pp. 650–655. DOI: 10.1109/ITSC.2015.112.

[55] E. Käfer et al. "Recognition of situation classes at road intersections". In: *2010 IEEE International Conference on Robotics and Automation (ICRA)*. May 2010, pp. 3960–3965. DOI: 10.1109/ROBOT.2010.5509919.

[56] Gregory Kahn et al. "PLATO: Policy Learning using Adaptive Trajectory Optimization". In: *arXiv:1603.00622 [cs]* (Mar. 2016).

[57] Christos Katrakazas et al. "Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions". In: *Transportation Research Part C: Emerging Technologies* 60 (Nov. 2015), pp. 416–442.

[58] M. Keller et al. "A Model Predictive Approach to Emergency Maneuvers in Critical Traffic Situations". In: *2015 IEEE 18th International Conference on Intelligent Transportation Systems*. Sept. 2015, pp. 369–374. DOI: 10.1109/ITSC.2015.69.

[59] Michael Kelly et al. "HG-DAgger: Interactive Imitation Learning with Human Experts". In: *to appear in IEEE International Conference on Robotics and Automation (ICRA)* (2019).

[60] ByeoungDo Kim et al. "Probabilistic Vehicle Trajectory Prediction over Occupancy Grid Map via Recurrent Neural Network". In: *arXiv:1704.07049 [cs]* (Apr. 2017). arXiv: 1704.07049.

[61] ZuWhan Kim et al. "A machine vision system for generating vehicle trajectories over extended freeway segments". In: *12th World Congress on Intelligent Transportation Systems*. 2005.

[62] S. Klingelschmitt and J. Eggert. "Using Context Information and Probabilistic Classification for Making Extended Long-Term Trajectory Predictions". In: *2015 IEEE 18th International Conference on Intelligent Transportation Systems*. Sept. 2015, pp. 705–711. DOI: 10.1109/ITSC.2015.120.

[63] S. Klingelschmitt et al. "Probabilistic situation assessment framework for multiple, interacting traffic participants in generic traffic scenes". In: *2016 IEEE Intelligent Vehicles Symposium (IV)*. June 2016, pp. 1141–1148. DOI: 10.1109/IVS.2016.7535533.

[64] Stefan Klingelschmitt, Volker Willert, and Julian Eggert. "Probabilistic, discriminative maneuver estimation in generic traffic scenes using pairwise probability coupling". In: *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on*. IEEE, 2016, pp. 1269–1276.

[65] Alexander Koenig, Tobias Rehder, and Soeren Hohmann. "Exact inference and learning in hybrid Bayesian Networks for lane change intention classification". In: *Intelligent Vehicles Symposium (IV), 2017 IEEE*. IEEE, 2017, pp. 1535–1540.

[66] R. Krajewski et al. "The highD Dataset: A Drone Dataset of Naturalistic Vehicle Trajectories on German Highways for Validation of Highly Automated Driving Systems". In: *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. Nov. 2018, pp. 2118–2125.

[67] A. Lawitzky et al. "Interactive scene prediction for automotive applications". In: *2013 IEEE Intelligent Vehicles Symposium (IV)*. June 2013, pp. 1028–1033. DOI: 10.1109/IVS.2013.6629601.

[68] Donghan Lee, Andreas Hansen, and J. Karl Hedrick. "Probabilistic inference of traffic participants' lane change intention for enhancing adaptive cruise control". In: *Intelligent Vehicles Symposium (IV), 2017 IEEE*. IEEE, 2017, pp. 855–860.

[69] Namhoon Lee et al. "DESIRE: Distant Future Prediction in Dynamic Scenes with Interacting Agents". In: *arXiv:1704.04394 [cs]* (Apr. 2017). arXiv: 1704.04394.

[70] Stéphanie Lefèvre, Christian Laugier, and Javier Ibañez-Guzmán. "Intention-Aware Risk Estimation for General Traffic Situations, and Application to Intersection Safety". In: *Inria Research Report* 8379 (Oct. 2013).

[71] Stéphanie Lefèvre, Dizan Vasquez, and Christian Laugier. "A survey on motion prediction and risk assessment for intelligent vehicles". In: *ROBOMECH Journal* 1.1 (July 2014), pp. 1–14.

[72] David Lenz et al. "Deep neural networks for markovian interactive scene prediction in highway scenarios". In: *Intelligent Vehicles Symposium (IV), 2017 IEEE*. IEEE, 2017, pp. 685–692.

[73] Jiachen Li, Wei Zhan, and Masayoshi Tomizuka. "Generic Vehicle Tracking Framework Capable of Handling Occlusions Based on Modified Mixture Particle Filter". In: *IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2018.

[74] Jiachen Li et al. "Generic Probabilistic Interactive Situation Recognition and Prediction: From Virtual to Real". In: *IEEE 21st International Conference on Intelligent Transportation Systems (ITSC)*. 2018.

[75] Liang Li et al. "A novel vehicle dynamics stability control algorithm based on the hierarchical strategy with constrain of nonlinear tyre forces". In: *Vehicle System Dynamics* 53.8 (Aug. 3, 2015), pp. 1093–1116.

[76] M. Liebner et al. "Driver intent inference at urban intersections using the intelligent driver model". In: *2012 IEEE Intelligent Vehicles Symposium (IV)*. June 2012, pp. 1162–1167. DOI: 10.1109/IVS.2012.6232131.

[77] Q. Lin et al. "MOHA: A Multi-Mode Hybrid Automaton Model for Learning Car-Following Behaviors". In: *IEEE Transactions on Intelligent Transportation Systems* (2018), pp. 1–8.

[78] Changliu Liu and Masayoshi Tomizuka. "Enabling safe freeway driving for automated vehicles". In: *2016 American Control Conference (ACC)*. IEEE. 2016, pp. 3461–3467.

[79] Changliu Liu, Wei Zhan, and Masayoshi Tomizuka. "Speed Profile Planning in Dynamic Environments via Temporal Optimization". In: *2017 IEEE Intelligent Vehicles Symposium (IV)*. June 2017.

[80] Alan Lukežič et al. "Discriminative Correlation Filter Tracker with Channel and Spatial Reliability". In: *International Journal of Computer Vision* (2018).

[81] Hengbo Ma et al. "Wasserstein Generative Learning with Kinematic Constraints for Probabilistic Prediction of Interactive Driving Behavior". In: *2019 IEEE Intelligent Vehicles Symposium*. 2019.

[82] Michael Mathieu, Camille Couprie, and Yann LeCun. "Deep multi-scale video prediction beyond mean square error". In: *arXiv:1511.05440 [cs, stat]* (Nov. 2015).

[83] M. McNaughton et al. "Motion planning for autonomous driving with a conformal spatiotemporal lattice". In: *2011 IEEE International Conference on Robotics and Automation (ICRA)*. May 2011, pp. 4889–4895.

[84] Kaouther Messaoud et al. "Relational Recurrent Neural Networks For Vehicle Trajectory Prediction". In: *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. 2019.

[85] Yoichi Morales et al. "Proactive driving modeling in blind intersections based on expert driver data". In: *Intelligent Vehicles Symposium (IV), 2017 IEEE*. IEEE, 2017, pp. 901–907.

[86] J. Morton, T. A. Wheeler, and M. J. Kochenderfer. "Analysis of Recurrent Neural Networks for Probabilistic Modeling of Driver Behavior". In: *IEEE Transactions on Intelligent Transportation Systems* 18.5 (May 2017), pp. 1289–1298. ISSN: 1524-9050. DOI: 10.1109/TITS.2016.2603007.

[87] Maximilian Naumann, Martin Lauer, and Christoph Stiller. "Generating Comfortable, Safe and Comprehensible Trajectories for Automated Vehicles in Mixed Traffic". In: *Proc. IEEE Intl. Conf. Intelligent Transportation Systems*. Hawaii, USA, Nov. 2018, pp. 575–582. DOI: 10.1109/ITSC.2018.8569658.

[88] Vicki L Neale et al. "An overview of the 100-car naturalistic study and findings". In: *National Highway Traffic Safety Administration, Paper* 5 (2005), p. 0400.

[89] H. Okuda et al. "Modeling and Analysis of Driving Behavior Based on a Probability-Weighted ARX Model". In: *IEEE Transactions on Intelligent Transportation Systems* 14.1 (Mar. 2013), pp. 98–112.

[90] B. Paden et al. "A Survey of Motion Planning and Control Techniques for Self-Driving Urban Vehicles". In: *IEEE Transactions on Intelligent Vehicles* 1.1 (Mar. 2016), pp. 33–55. ISSN: 2379-8858. DOI: 10.1109/TIV.2016.2578706.

[91] Deepak Pathak, Philipp Krahenbuhl, and Trevor Darrell. "Constrained Convolutional Neural Networks for Weakly Supervised Segmentation". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, pp. 1796–1804.

[92] Dan Pelleg, Andrew W Moore, et al. "X-means: Extending k-means with efficient estimation of the number of clusters." In: *ICML*. Vol. 1. 2000, pp. 727–734.

[93] Derek J. Phillips, Tim A. Wheeler, and Mykel J. Kochenderfer. "Generalizable intention prediction of human drivers at intersections". In: *Intelligent Vehicles Symposium (IV), 2017 IEEE*. IEEE, 2017, pp. 1665–1670.

[94] F. Poggenhans et al. "Lanelet2: A high-definition map framework for the future of automated driving". In: *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. Nov. 2018, pp. 1672–1679.

[95] Philip Polack et al. "The Kinematic Bicycle Model: a Consistent Model for Planning Feasible Trajectories for Autonomous Vehicles?" In: *2017 IEEE Intelligent Vehicles Symposium (IV)*. June 2017.

[96] Ewoud AI Pool, Julian FP Kooij, and Dariu M. Gavrila. "Using road topology to improve cyclist path prediction". In: *Intelligent Vehicles Symposium (IV), 2017 IEEE*. IEEE, 2017, pp. 289–296.

[97] Xiangjun Qian et al. "Motion planning for urban autonomous driving using Bézier curves and MPC". In: *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on*. IEEE, 2016, pp. 826–833.

[98] Jannik Quehl et al. "How Good is My Prediction? Finding a Similarity Measure for Trajectory Prediction Evaluation." In: *2017 IEEE 18th International Conference on Intelligent Transportation Systems (ITSC)*. 2017, pp. 120–125.

[99] Vasili Ramanishka et al. "Toward Driving Scene Understanding: A Dataset for Learning Driver Behavior and Causal Reasoning". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 7699–7707.

[100] Shaoqing Ren et al. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". In: *Advances in Neural Information Processing Systems*. 2015, pp. 91–99.

[101] Nicholas Rhinehart, Kris Kitani, and Paul Vernaza. "R2P2: A ReparameteRized Pushforward Policy for Diverse, Precise Generative Path Forecasting". In: *European Conference on Computer Vision*. Springer. 2018.

[102] Nicholas Rhinehart, Rowan McAllister, and Sergey Levine. "Deep Imitative Models for Flexible Inference, Planning, and Control". In: (Oct. 2018).

[103] Alexandre Robicquet et al. "Learning Social Etiquette: Human Trajectory Understanding In Crowded Scenes". In: *ECCV 2016*. Springer International Publishing, 2016, pp. 549–565.

[104] Andrey Rudenko et al. "Human Motion Trajectory Prediction: A Survey". In: *arXiv preprint arXiv:1905.06113* (2019).

[105] Dorsa Sadigh et al. "Planning for Autonomous Cars that Leverage Effects on Human Actions." In: *Robotics: Science and Systems*. 2016.

[106] João Salvado, Luís MM Custódio, and Daniel Hess. "Contingency planning for automated vehicles". In: *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, 2016, pp. 2853–2858.

[107] John M. Scanlon et al. "Potential of intersection driver assistance systems to mitigate straight crossing path crashes using US nationally representative crash data". In: *Intelligent Vehicles Symposium (IV), 2015 IEEE*. IEEE, 2015, pp. 1207–1212.

[108] M. Schreier, V. Willert, and J. Adamy. "An Integrated Approach to Maneuver-Based Trajectory Prediction and Criticality Assessment in Arbitrary Road Environments". In: *IEEE Transactions on Intelligent Transportation Systems* 17.10 (Oct. 2016), pp. 2751–2766. ISSN: 1524-9050. DOI: 10.1109/TITS.2016.2522507.

[109] J. Schulz et al. "Interaction-Aware Probabilistic Behavior Prediction in Urban Environments". In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Oct. 2018, pp. 3999–4006.

[110] Mohsen Sefati et al. "Towards tactical behaviour planning under uncertainties for automated vehicles in urban scenarios". In: *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE. 2017, pp. 1–7.

[111] Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. "Safe, Multi-Agent, Reinforcement Learning for Autonomous Driving". In: *arXiv preprint arXiv:1610.03295* (2016).

[112] Shai Shalev-Shwartz and Amnon Shashua. "On the Sample Complexity of End-to-end Training vs. Semantic Abstraction Training". In: *arXiv preprint arXiv:1604.06915* (Apr. 2016).

[113] Shai Shalev-Shwartz et al. "Long-term Planning by Short-term Prediction". In: *arXiv preprint arXiv:1602.01580* (2016).

[114] Tianmin Shu et al. "Perception of human interaction based on motion trajectories: From aerial videos to decontextualized animations". In: *Topics in cognitive science* 10.1 (2018), pp. 225–241.

[115] Dan Simon. *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. New York, NY, USA: Wiley-Interscience, 2006. ISBN: 0471708585.

[116] Liting Sun, Wei Zhan, and Masayoshi Tomizuka. "Probabilistic Prediction of Interactive Driving Behavior via Hierarchical Inverse Reinforcement Learning". In: *IEEE 21st International Conference on Intelligent Transportation Systems (ITSC)*. 2018.

[117] Liting Sun et al. "A Fast Integrated Planning and Control Framework for Autonomous Driving via Imitation Learning". In: *ASME 2018 Dynamic Systems and Control Conference*. American Society of Mechanical Engineers, Sept. 2018, pp. 1–11.

[118] Liting Sun et al. "Courteous autonomous cars". In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2018, pp. 663–670.

[119] Liting Sun et al. "Interpretable Modelling of Driving Behaviors in Interactive Driving Scenarios based on Cumulative Prospect Theory". In: *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. 2019.

[120] Q. Tran and J. Firl. "Modelling of traffic situations at urban intersections with probabilistic non-parametric regression". In: *2013 IEEE Intelligent Vehicles Symposium (IV)*. 2013 IEEE Intelligent Vehicles Symposium (IV). June 2013, pp. 334–339. DOI: 10.1109/IVS.2013.6629491.

[121] Chris Urmson. *The View from the Front Seat of the Google Self-Driving Car: A New Chapter*. Aug. 2016. URL: https://medium.com/@chris_urmson/the-view-from-the-front-seat-of-the-google-self-driving-car-a-new-chapter-7060e89cb65f.

[122] W. Wang, C. Liu, and D. Zhao. "How Much Data Are Enough? A Statistical Approach With Case Study on Longitudinal Driving Behavior". In: *IEEE Transactions on Intelligent Vehicles* 2.2 (June 2017), pp. 85–98. ISSN: 2379-8858. DOI: 10.1109/TIV.2017.2720459.

[123] Wenshuo Wang, Junqiang Xi, and Ding Zhao. "Learning and Inferring a Driver's Braking Action in Car-Following Scenarios". In: *IEEE Transactions on Vehicular Technology* (2018). DOI: 10.1109/TVT.2018.2793889.

[124] Wenshuo Wang, Weiyang Zhang, and Ding Zhao. "Understanding V2V Driving Scenarios through Traffic Primitives". In: *arXiv:1807.10422 [cs, stat]* (July 2018). arXiv: 1807.10422.

[125] Junqing Wei et al. "A behavioral planning framework for autonomous driving". In: *2014 IEEE Intelligent Vehicles Symposium Proceedings*. 2014 IEEE Intelligent Vehicles Symposium Proceedings. June 2014, pp. 458–464.

[126] Tim A Wheeler, Philipp Robbel, and Mykel J Kochenderfer. "Analysis of microscopic behavior models for probabilistic modeling of driver behavior". In: *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE. 2016, pp. 1604–1609.

[127] Thierry Wyder et al. "A Bayesian filter for modeling traffic at stop intersections". In: *Intelligent Vehicles Symposium (IV), 2015 IEEE*. IEEE, 2015, pp. 1252–1257. (Visited on 10/18/2015).

[128] Donghao Xu et al. "Scene-aware driver state understanding in car-following behaviors". In: *Intelligent Vehicles Symposium (IV), 2017 IEEE*. IEEE, 2017, pp. 1490–1496.

[129] Huazhe Xu et al. "End-to-end Learning of Driving Models from Large-scale Video Datasets". In: *arXiv:1612.01079 [cs]* (Dec. 2016).

[130] Wenda Xu et al. "Motion planning under uncertainty for on-road autonomous driving". In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. May 2014, pp. 2507–2512. DOI: 10.1109/ICRA.2014.6907209.

[131] Dongfang Yang et al. "Top-view Trajectories: A Pedestrian Dataset of Vehicle-Crowd Interaction from Controlled Experiments and Crowded Campus". In: *arXiv:1902.00487 [cs]* (Feb. 2019). arXiv: 1902.00487.

[132] Shun Yang et al. "Feature Analysis and Selection for Training an End-To-End Autonomous Vehicle Controller Using Deep Learning Approach". In: *2017 IEEE Intelligent Vehicles Symposium (IV)*. June 2017, pp. 1033–1038.

[133] Yuki Yoshihara et al. "Autonomous Predictive Driving for Blind Intersections". In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2017, pp. 3452–3459.

[134] Wei Zhan et al. "A non-conservatively defensive strategy for urban autonomous driving". In: *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. Nov. 2016, pp. 459–464.

[135] Wei Zhan et al. "Constrained Iterative LQR for On-Road Autonomous Driving Motion Planning". In: *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. Oct. 2017.

[136] Wei Zhan et al. "Constructing a Highly Interactive Vehicle Motion Dataset". In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2019.

[137] Wei Zhan et al. "INTERACTION Dataset: An INTERnational, Adversarial and Cooperative moTION Dataset in Interactive Scenarios with Semantic Maps". In: 2019.

[138] Wei Zhan et al. "Probabilistic Prediction from Planning Perspective: Problem Formulation, Representation Simplification and Evaluation Metric". In: *Intelligent Vehicles Symposium (IV), 2018 IEEE*. 2018, pp. 1150–1156.

[139] Wei Zhan et al. "Spatially-Partitioned Environmental Representation and Planning Architecture for On-Road Autonomous Driving". In: *2017 IEEE Intelligent Vehicles Symposium (IV)*. June 2017, pp. 632–639.

[140] Wei Zhan et al. "Towards a Fatality-Aware Benchmark of Probabilistic Reaction Prediction in Highly Interactive Driving Scenarios". In: *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE. 2018, pp. 3274–3280.

[141] Ding Zhao et al. "Accelerated Evaluation of Automated Vehicles Safety in Lane-Change Scenarios Based on Importance Sampling Techniques". In: *IEEE transactions on intelligent transportation systems* 18.3 (2016), pp. 595–607.

[142] Yu Zheng. "Trajectory Data Mining: An Overview". In: *ACM Trans. Intell. Syst. Technol.* 6.3 (May 2015), 29:1–29:41. ISSN: 2157-6904.

[143] Brian D Ziebart et al. "Maximum Entropy Inverse Reinforcement Learning." In: *AAAI*. Vol. 8. Chicago, IL, USA. 2008, pp. 1433–1438.

[144] Julius Ziegler et al. "Making Bertha Drive – An Autonomous Journey on a Historic Route". In: *IEEE Intelligent Transportation Systems Magazine* 6.2 (2014), pp. 8–20.

[145] Julius Ziegler et al. "Trajectory planning for Bertha - A local, continuous method". In: *2014 IEEE Intelligent Vehicles Symposium Proceedings*. June 2014, pp. 450–457.