TECHNISCHE UNIVERSITÄT MÜNCHEN

Fakultät für Informatik

Risk-Constrained Interactive Planning for Balancing Safety and Efficiency of Autonomous Vehicles

Julian Bernhard

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitz:		Prof. Dr. Susanne Albers
Prüfende der Dissertation:	1.	Prof. Dr.-Ing. habil. Alois C. Knoll
	2.	Prof. Mykel J. Kochenderfer, Ph.D.

Die Dissertation wurde am 07.12.2021 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 13.06.2022 angenommen.

Abstract

Balancing safety and efficiency when navigating in dense traffic remains an open challenge in autonomous driving. Safety envelopes restrict the allowed planning region and yield interpretable safety, yet, by not adjusting to the behavior of other participants, they sacrifice efficiency in dense traffic. Interactive planners anticipate the reactions of other traffic participants during planning which increases efficiency. Modeling safety is accomplished in a probabilistic manner. However, achieving a meaningful collision risk with such formulations is computationally too demanding in online planning. This thesis presents risk-constrained interactive online planning satisfying a specifiable maximum percentage of safety envelope violations over time. A specifiable envelope violation risk serves as an interpretable parameter balancing safety and efficiency. A comprehensive definition of risk requires coverage of other participants' microscopic behavior, which is achieved by defining behavior hypotheses partitioning a behavior space. A game-theoretic planning approach based on Monte Carlo Tree Search (MCTS) uses these behavior hypotheses to predict other traffic participants probabilistically. Using a robustness measure improves convergence by predicting worst-case outcomes to the autonomous vehicle with priority during the search. The frequency of safety envelope violations of human drivers in dense traffic inspires the development of an interpretable risk measure. A risk-constrained action selection strategy is developed for the MCTS planner to generate plans satisfying a specified envelope violation risk. Online risk-constrained planning is accomplished with variants of the MCTS planner parallelizing and warm starting the search tree with prior learned experiences encoded into neural networks. The approach is shown in simulation to be superior to state-of-the-art interactive planners in dense traffic with uncertainty about the microscopic behavior of other participants. It enables to balance safety and efficiency in an interpretable manner. These properties are preserved when applying experience-based and parallelized online planning.

Zusammenfassung

Eine offene Herausforderung bei der Navigation von autonomen Fahrzeugen in dichtem Verkehr ist die Abwägung zwischen Sicherheit und Effizienz. Sicherheitsküllkurven schränken den erlaubten Navigationsbereich ein und bieten interpretierbare Sicherheit. Da sie sich jedoch nicht an das Verhalten anderer Verkehrsteilnehmer anpassen, führen sie zu ineffizientem Verhalten in dichtem Verkehr. Interaktive Planer integrieren die Reaktionen anderer Verkehrsteilnehmer in die Planung und steigern so die Effizienz. Dabei modelieren sie Sicherheit mit probabilistischen Methoden. Die Berechnung eines aussagekräftigen Kollisionsrisikos mit solchen Formulierungen ist jedoch zur Laufzeit rechnerisch zu anspruchsvoll. In dieser Arbeit wird eine risikobeschränkte interaktive Planung vorgestellt, die einen vorgebbaren maximalen Prozentsatz an Verletzungen der Sicherheitshüllkurve über die Zeit berücksichtigt. Ein spezifizierbares Hüllkurvenverletzungsrisiko dient als interpretierbarer Parameter, der Sicherheit und Effizienz gewichtet. Eine umfassende Definition des Risikos erfordert die Berücksichtigung des mikroskopischen Verhaltens der anderen Verkehrsteilnehmer, was durch die Definition von Verhaltenshypothesen, die einen Verhaltensraum partitionieren, erreicht wird. Ein spieltheoretischer Planungsansatz, der auf einer Monte Carlo Baumsuche (MCTS) basiert, verwendet diese Verhaltenshypothesen, um andere Verkehrsteilnehmer probabilistisch vorherzusagen. Ein Robustheitsmaß evaluiert vorrangig nachteilige Zustände des autonomen Fahrzeuges während der Vorwärtssuche und verbessert hierdurch deren Konvergenz. Die Häufigkeit von Verletzungen der Sicherheitsküllkurven bei menschlichen Fahrern in dichtem Verkehr ist Grundlage für die Entwicklung einer interpretierbaren Risikometrik. Es wird eine risikobeschränkte Aktionsauswahlstrategie für den MCTS-Planer entwickelt, die Pläne generiert, die ein bestimmtes Hüllkurvenverletzungsrisiko erfüllen. Weiterhin wird eine risikobeschränkte Planung zur Laufzeit erreicht durch Parallelisierung des MCTS-Planers und Initialisierung des Suchbaums mit zuvor mit neuronalen Netzen erlernten Erfahrungen. Eine simulative Studie zeigt, dass der vorgestellte Ansatz im dichten Verkehr mit Unsicherheit über das mikroskopische Verhalten der anderen Verkehrsteilnehmer aktuellen interaktiven Planern überlegen ist. Er gewichtet dabei Sicherheit und Effizienz auf interpretierbare Weise. Diese Eigenschaften bleiben auch bei der Anwendung erfahrungsbasierter und parallelisierter Planung zur Laufzeit erhalten.

Contents

1 Introduction

Balancing safety and efficiency is a significant challenge regarding autonomous driving. In the upcoming decades, automation levels will increase from assistance systems, e.g., lane-keeping and traffic jam assistants, already introduced in the market, to partly and fully automated driving [1]. The potential benefits arising out of these revolutionizing technologies will certainly fundamentally influence our everyday lives [2]. In general, automated driving promises an increase in safety. In contrast, higher levels of automation will progressively improve efficiency in mobility bringing a high level of comfort and reduction of travel cost to passengers [3]. Higher automation levels require Autonomous Vehicles (AVs) to solve increasingly complex driving tasks, exemplified in Fig. 1.1, being characterized by close interaction with humans and uncertainty about human driving styles. Navigating efficiently in such dense traffic situations requires planning approaches modeling the reactions of other participants during planning. However, such *interactive planners* miss a suitable mechanism to balance safety and efficiency.

In dense traffic, accidents have reduced severity due to decreased speed. Humans can thus avoid a too conservative driving style by not strictly adhering to physically required safe distances [6, 7]. They anticipate interactions between traffic participants to maintain the overall traffic flow. Highly automated vehicles will significantly affect the efficiency of mixed traffic [8]. When motion planners of AVs adhere to a strictly safety-oriented driving style neglecting uncertainty and interactions, this fosters abrupt safety maneuvers. Such sudden maneuvers reduce

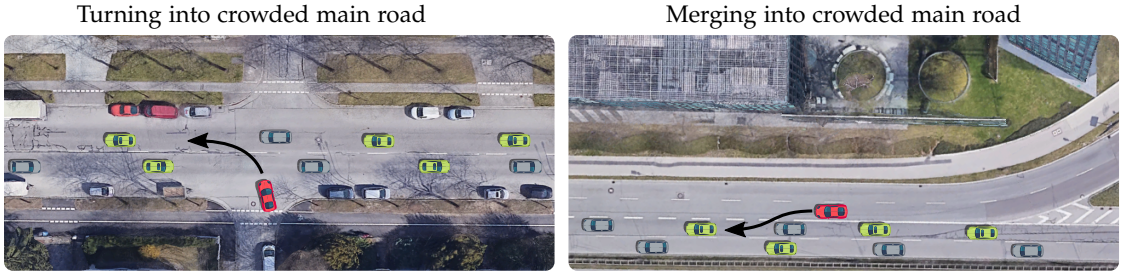


Figure 1.1.: Examples for dense traffic situations demanding an interpretable way to balance safety and efficiency in a planning component of an Autonomous Vehicle (AV). Backgrounds taken from [4, 5].

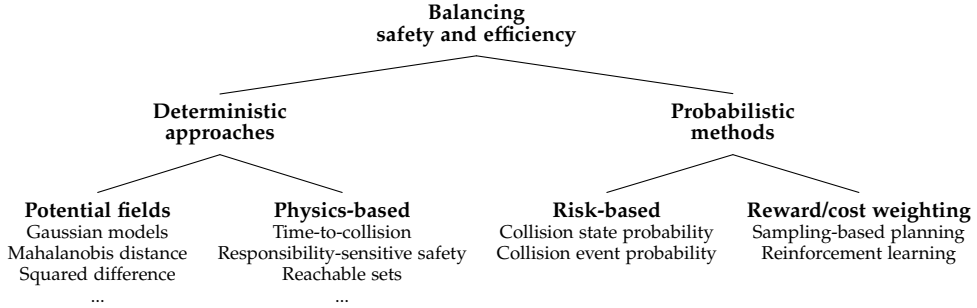


Figure 1.2.: Overview of concepts to balance safety and efficiency in motion planning for AVs.

the enjoyment of driving and negatively affect the acceptance of these systems [9]. Even more critical, abrupt braking interrupts the traffic flow and facilitates dangerous reactions of other participants, which may provoke rear-end crashes [10–12]. It thus seems plausible to equip AVs with an understanding of the risk of interactions with other road users. However, developing an everyday societal awareness and acceptance of the risks of AVs [13, 14] requires developing technical concepts to enable risk to be used within interactive motion planning. Though there exist interactive planning approaches which find efficient decisions in dense traffic [15], they do not integrate meaningful risk measures.

This thesis contributes risk-constrained interactive planning for AVs to balance their safety and efficiency when navigating in dense traffic. In particular, the presented concept fills the gap of having available a meaningful risk definition for automated driving in dense traffic at speeds below $50\,\mathrm{km\,h^{-1}}$ where a lower accident severity justifies making trade-offs between safety and efficiency. For applicability in dense traffic, an interactive planning approach based on Monte Carlo Tree Search (MCTS) is developed, which constrains the risk of violating safety envelopes while considering the uncertainty about the behavior of other participants. It uses probabilistic predictions to cover the variety of human driving styles based on defining a behavior space of other participants and reduces the complexity of MCTS by evaluating worst-case outcomes of the AV with priority. Further, the proposed approach applies offline learning of prior experiences and parallelization to enable online planning.

1.1. Balancing Safety and Efficiency in Autonomous Driving

Existing approaches to balance safety and efficiency of AVs can be differentiated into two major directions [16], deterministic and probabilistic methods (cf. Fig. 1.2).

1.1.1. Deterministic Approaches

Deterministic approaches for balancing safety and efficiency do not incorporate probability information from current or past observed states. Instead, they design a metric function that quantifies the degree of safety based on currently observable dynamic properties of the environment [17].

Potential fields are metrics that decay with increasing distance between ego vehicle, and other participants [18–20]. They provide a continuous evaluation of safety in the current state. By

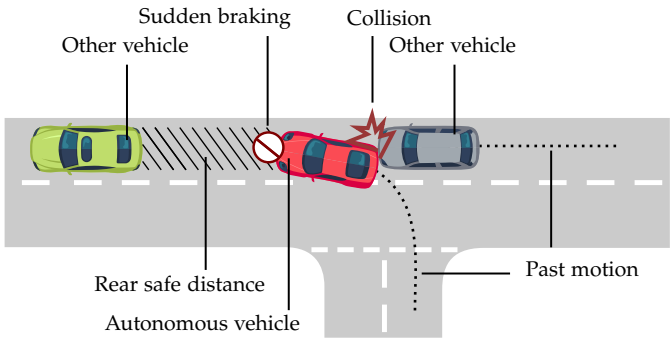


Figure 1.3.: Potentially unsafe situations in dense traffic due to conservative driving. During a left turn of the AV, its leading vehicle (green) brakes. To avoid violating the rear safe distance, the AV performs a sudden braking maneuver. Being not expected by the oncoming other vehicle (grey), it collides with the AV.

fitting the potential field to human driving data, an understanding of different levels of risk is obtained [18, 21]. Physics-based metrics employ a physically-realistic model of other participants' dynamics to categorize the current state into safe or unsafe. Simpler models categorize based on the time-to-collision [22]. More complex approaches check violation of a safety envelope which is defined using lateral limits [23], lateral and longitudinal limits [24, 25], or reachable sets [26, 27].

A possibility to arrive at a balance of safety and efficiency with potential fields is to integrate them as continuous cost criteria into the optimality function of the motion planner [18]. Such concepts are applied for both interactive [28–30] and non-interactive planners [31]. Physics-based approaches do not aim to balance safety and efficiency. Instead, they strictly prioritize safety by forcing the ego trajectory to stay within the safety envelope [24, 32]. Tuning the safety envelope by adapting physical parameters such as accelerations and response times, e.g., using human driving data [33, 34], or online from observed behavior [22] provides a certain balance of safety and efficiency. Nevertheless, not applying the true physical limits as parameters softens the strict safety guarantees given by these approaches. Non-interactive planning frequently uses envelope restrictions to target safe motion planning [26, 35].

Cost terms and safety envelopes disregard the probability of encountered traffic situations. Thus, these concepts prevent the definition of a risk metric that relates the parameterization of the envelope or cost function to the statistics of safety violations in the environment. Further, a common argument against the use of envelope restrictions in dense traffic is that it fosters overly cautious behavior [36]. Humans do not always adhere to fail-safe planning assumptions and violate, e.g., the safe distance to front vehicles in dense traffic [32, p. 136]. In some situations, this may even counteract the original safety objective when human drivers react in a dangerous way to a conservative driving style. Several studies [10–12] show that compared to humans, AVs are involved over-proportionally into low severity accidents where the AV is hit at the rear end by a human driver. These analyses suggest that overly cautious driving of AVs may not be a meaningful safety goal in low severity situations characterized by high traffic density and interactivity between participants (cf. Fig. 1.3).

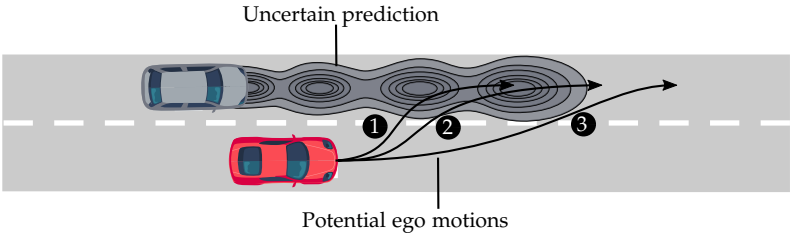


Figure 1.4.: Probabilistic notion of safety under uncertain behavior of other traffic participants. The uncertainty of predicted motion states increases with prediction time. The figure depicts a prediction marginalized over time. The calculation of the collision risk $P_{\text{col}}(x)$ considers the uncertainty of future states. In this example, the collision risk decreases from ego motions 1) to 3), $P_{\text{col}}(1) > P_{\text{col}}(2) > P_{\text{col}}(3)$.

1.1.2. Probabilistic Methods

Probabilistic methods to balance safety and efficiency in motion planning include information about the probability of unsafe states (cf. Fig. 1.4). A common approach is to define a risk measure that serves as a constraint in the planner's optimality function. Parts of this review are based on previous work presented in [37].

In the functional safety sense, the term risk is probabilistic and defined as the combination of the probability of occurrence of harm and the severity of that harm [38]. Existing probabilistic risk definitions often consider collision as harmful events in risk-based planning approaches [38–41]. Probabilistic collision risk is used in [42] to model lane changing by applying risk measures to value distributions. These concepts arose in finance and are applied to robotics and autonomous driving [42, 43]. Constraining motion planning by collision risk is proposed in [39] for a Partially Observable Markov Decision Process (POMDP) planner and in [38] to incorporate various uncertainties into a Model Predictive Control (MPC) algorithm. Müller and Buchholz [44] constrain the risk of violating the safe distance. A qualitative calculation of risk based on Fuzzy sets is given in [45]. The presented approaches make use of the Collision State Probability (CSP), the probability of spatial overlap at discrete times [46].

Three aspects are missing in existing probabilistic risk definitions. 1) The employed prediction of other participants during planning does not effectively cover behavior variations and identify harmful states, which may cause underestimating the actual risk. 2) The CSP considers occurrences of harmful states at discrete time points. It thus neglects the continuity of the driving environment. In contrast, the risk definition in the functional safety sense applies normalization by the driven time or miles in the environment. The Collision Event Probability (CEP) being better suited to express the duration aspect has been used in post-analysis of planned motions [47] and in planning approaches requiring a pre-generation of risk maps [48]. 3) Collision probability can only serve as a safety measure if its approximation during planning is in the order of fatal events per driving hour, $P_{\text{fatal}} \approx 10^{-6}/\text{h}$ [49]. However, satisfying such magnitudes requires an unrealistically large number of prediction samples of other participants' behavior. This drawback makes a risk metric defined over collision as a harmful event infeasible for online planning.

Additionally, prescribed approaches apply long-term, i.e., maneuver-based prediction of other participants neglecting interactions. Interactive planners anticipate how others react to the ego-motion already during planning which is especially meaningful in dense traffic [50]. However,

they lack a meaningful definition of safety. Existing interactive planners employ *single-objective* optimality criteria with manual or data-based cost tuning to avoid collisions [51–56]. They maximize the expected return, which combines collision cost and success reward over predicted future states. This optimality criterion can be regarded as a probabilistic approach to balance safety and efficiency. Sampling-based planners are common approaches to dealing with uncertainty in interactive planning. However, they exhibit combinatorial complexity over the number of participants and explored motion primitives (cf. Sec. 2.5). Implementing risk constraints into multi-objective interactive planners increases computational complexity even further. Learning plans offline, before execution, e.g., using constrained dynamic programming in a discretized state space [57] partly circumvents this problem. However, such approaches do not easily generalize to non-learned environments.

1.2. Motivating Interpretable Risk for Interactive Planning

In summary, existing approaches to balance safety and efficiency in motion planning cannot meet the requirements for dense traffic characterized by high uncertainty about the behavior of other participants and their inherent interactivity. A prioritization of safety with secondary consideration of efficiency based on safety envelope restrictions [32] is meaningful for traffic with higher severity. In dense traffic, these concepts foster conservative driving with potentially dangerous reactions of humans [10–12]. Probabilistic risk criteria are promising as a safety goal in dense traffic. However, existing probabilistic definitions of risk miss computational feasibility, coverage of behavior uncertainty, and interpretation regarding statistically measured safety violations in the environment. Interactive planning algorithms enable the generation of efficient plans in dense traffic to avoid conservative driving. However, they lack a meaningful specification of risk. Partly since risk-based optimality additionally increases the already higher computational complexity of interactive compared to maneuver-based planning.

The goal of this thesis is, therefore, the development of a risk definition and accompanying interactive motion planner for AVs to balance safety and efficiency under the presence of behavior uncertainty, which overcomes the discussed drawbacks of existing risk definitions. Firstly, a comprehensive definition of such a risk measure must cover all variations of other participants' behavior. Secondly, apart from referring to safety violations in the current traffic situation, it must consider the accumulation of risk for all potential future traffic cases. By that, the risk definition achieves that the safety violations averaged over the whole driving time are quantitatively related to the specified risk level given as a parameter to the motion planning algorithm. Thirdly, it shall not be based on collisions as harmful events since it is computationally infeasible to calculate acceptable levels of collision risk during online planning. The previous demands on a suitable risk measure for interactive planning are subsumed under the term interpretable risk formalism:
Interpretable risk formalism: *A risk formalism is interpretable if a quantitative mapping exists between specified risk and the observed safety statistic of planned motions under this formalism.*

This concept does not rely on a specific definition of safety or efficiency and is conceptualized in Fig. 1.5. An AV, also denoted in the remainder of this thesis, the ego vehicle, wants to perform

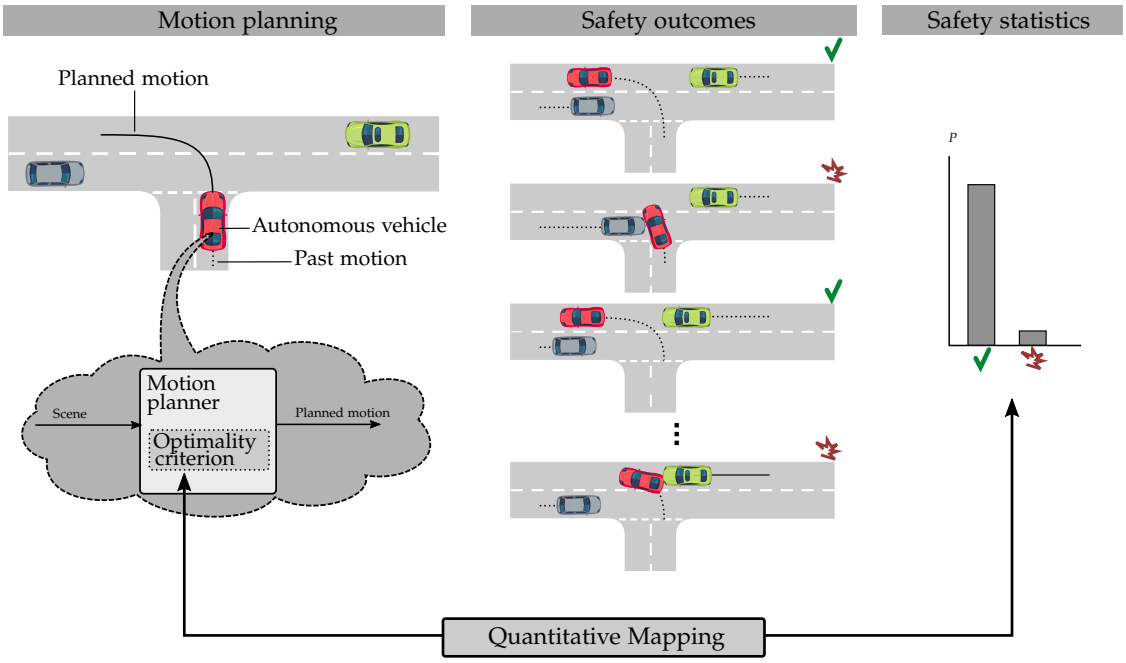


Figure 1.5.: Concept of an interpretable risk formalism. Executing a planned motion in the environment potentially yields unsafe outcomes due to behavior variations of other participants. An interpretable risk formalism defines a quantitative mapping between the optimality criterion used by the motion planner and the resulting safety statistic.

a left turn from a side road into the main road. There is oncoming human traffic on both lanes. The motion of the ego vehicle is planned according to a certain optimality criterion given the scene description of its surroundings. Assuming the ego vehicle can repeatedly follow the planned motion in the same situation, the variation in human driving behavior yields different safety outcomes in each of these cases. An interpretable risk formalism provides a defined mapping from specified risk to statistically averaged safety outcomes.

1.3. Contributions to Risk-Constrained Interactive Planning

This thesis develops an interpretable risk formalism that allows specification of a maximum envelope violation risk, defined as the maximum percentage of driven time a safety envelope is allowed to be violated. The formalism requires an additional optimality constraint to be integrated into an interactive planner. The resulting multi-objective interactive planning approach generates plans for which the specified maximum risk follows the statistically observed percentage of envelope violations. The uncertainty in human driving behavior is covered by using sample-efficient probabilistic predictions in behavior spaces. The planner integrates offline learning of prior experience and parallelization to reduce the computational demands during online planning.

Specifically, this thesis presents the Risk-Constrained Robust Stochastic Bayesian Game (RC-RSBG), which models risk-constrained interactive planning under behavior uncertainty in a game-theoretic manner. This novel decision theoretic model extends the Stochastic Bayesian

Game (SBG) [58], game-theoretically modeling irrational behavior with robustness-based optimality of the Robust Markov Decision Process (RMDP) and ideas from the Constrained POMDP (C-POMDP). Along the chapters of this thesis a variant of Simultaneous-Move MCTS (SM-MCTS) is developed solving the RC-RSBG. A comprehensive definition of risk requires coverage of other participants' microscopic behavior, which is achieved by defining behavior hypotheses partitioning a behavior space. Given posterior beliefs over the behavior hypothesis, other participants' motions can be predicted probabilistically during SM-MCTS. The search employs worst-case action selection within behavior hypotheses to improve sample efficiency and back-propagation of time-normalized envelope violations to ensure consistency of the risk estimate. A risk-constrained action selection strategy in the SM-MCTS ensures the satisfaction of the risk formalism. Online risk-constrained planning is accomplished with variants of the MCTS planner parallelizing and warm starting the search tree with prior learned experiences encoded into neural networks.

Therefore, this thesis presents the following five major contributions to the field of interactive motion planning for AVs:

1. **Coverage of behavior variations:** To define risk metrics over behavior uncertainty which are interpretable with respect to hazard statistics in the environment, the prediction model must use a sample space covering all continuous variations in behavior occurring potentially in the environment. The proposed interactive planning approach applies behavior spaces with an accompanying design process splitting behavior spaces into behavior hypotheses. Using this design process in combination with a specific type of posterior belief update, the interactive prediction of other traffic participants covers their expected continuous behavior variations in the environment.

2. **Sample-efficient interactive planning:** Interactive planning requires evaluation of the planned ego-motion together with all potential reactions of other participants. The number of combinations of ego and other vehicles' motions becomes infinite when predicting continuous behavior variations. Therefore, this thesis contributes a decision-theoretic model that integrates worst-case optimality over behavior hypothesis. Planning under this model using SM-MCTS sample-efficiently explores combinations of ego and other vehicles' motions violating a risk constraint while avoiding conservative solutions obtained by worst-case considerations over the entire behavior space.

3. **Interpretable risk for interactive planning**: The proposed approach integrates an interpretable risk constraint into interactive planning. Inspired by how often humans violate safety envelopes statistically, this thesis formalizes the risk of violating a safety envelope over time. The proposed interactive planning approach finds an optimal motion plan which adheres to a maximum envelope violation risk given the uncertainty in other participants' behavior. By integrating the duration of safety envelope violations into the risk formalism, the proposed risk definition becomes statistically interpretable concerning the actual statistic of envelope violations observed in the environment. A multi-objective interactive planning approach is presented, which integrates the risk formalism using constrained

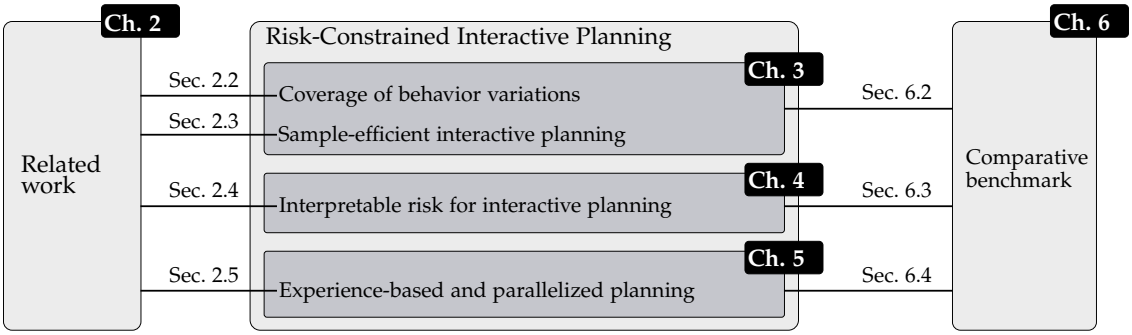


Figure 1.6.: Contributions within the structure of this thesis.

policy optimization within the SM-MCTS.

4. **Online risk-constrained planning**: This thesis contributes a risk-constrained interactive planner capable of generating plans under real-time constraints. Existing concepts accelerating online planning with learned prior experience and parallelization are well understood for single-objective planning. This thesis contributes an adaptation and analyses of these concepts for the multi-objective setting. It presents a concept to benefit from a priori learned value functions to improve the performance of the risk-constrained planner and shows that parallelizing the search has greater benefits in multi-objective planning compared to single-objective planning. As a result, online risk-constrained interactive planning becomes computationally feasible.

5. **Comparative benchmark**: This thesis provides an extensive comparative study on interactive planning. It analyzes a variety of state-of-the-art sampling-based interactive planners against the proposed approach. Apart from a qualitative analysis of specific driving situations, this work provides statistical evaluations over multiple scenarios and two scenario types to assess the efficiency and interpretability of the risk formalism. Therefore, an OpenSource benchmarking framework, BARK (**B**ehavior Benchm**ark**), is contributed to systematically evaluate the effects of prediction uncertainty on the statistically observed envelope violation risk.

1.4. Structure of the Thesis

The main contributions of this thesis are structured as depicted in Fig. 1.6. The thesis starts with Chapter 2 on related work in interactive planning. After positioning interactive planning within the context of an AV architecture, the remaining related work sections discuss the previous work of each contribution. Chapter 3 then introduces the RSBG, a game-theoretic model combining robustness-based optimality with the SBG modeling irrational behavior in gameplay. This chapter contributes the coverage of continuous behavior variations and sampling-efficient interactive planning using a variant of SM-MCTS. The following Chapter 4, proposes the interpretable risk formalism, and an extended game-theoretic model the RC-RSBG which integrates the risk formalism into the RSBG. It describes how to interactively plan under this risk formalism by

extending the SM-MCTS from Chapter 3 to solve the RC-RSBG. Chapter 5 then presents a parallelized variant of the RC-RSBG planner and how to learn and apply prior experiences to achieve online planning capability. Chapter 6 provides chapter-wise a comparative benchmark of each proposed contribution. Future work extending the work described in this thesis is provided in Chapter 7. The main outcomes of this thesis are summarized in Chapter 8.

2 State of the Art in Interactive Planning for Autonomous Driving

This chapter presents related work on interactive planning algorithms for AVs. The integration of such planning concepts into the architecture of AVs is explained in Sec. 2.1. Sec. 2.2 describes existing concepts to predict other traffic participants in interactive planning. Probabilistic and multi-agent interactive planning is the focus of Sec. 2.3. Sec. 2.1 describes how existing optimality criteria of interactive planners are used to balance safety and efficiency. Previous concepts to reduce the computational demands of interactive planning are presented in Sec. 2.5.

2.1. Interactive Planning within Autonomous Driving Architecture

Sense-plan-act is a well-known architectural software concept in robotics to implement autonomous systems interacting with an environment [59]. Similar functional tasks are realized in so called software driving stacks of AVs [60] by a perception, planning and control unit [61, 62] (cf. Fig. 2.1). This separation provides a simplistic perspective onto the often more interwoven architectural concepts [63–67]. The perception component integrates raw sensor data, preprocessing, and feature detection and fuses multiple sources of preprocessed sensor data, e.g., coming from lidars, radars, cameras, and localization measurements, into an abstract representation of the environment [61]. The fused representation contains the perceived road geometry and static and dynamic objects in a grid- or object-based form [68]. Additionally, the representation is often augmented with uncertainty information modeling potential errors in the state [69] and road boundary estimates [70]. Thereby, using a high-definition predefined map of the environment improves the quality of the fusion process. The task of the planning layer is to create a drivable trajectory consisting of a time-stamped sequence of future dynamic states [71]. The created trajectory must adhere to the non-holonomic constraints of the vehicle [72] to be trackable by a subsequent trajectory following controller [73, 74]. Apart from this classical architecture, end-to-end planning tries to establish a learning-based paradigm to replace all architectural components

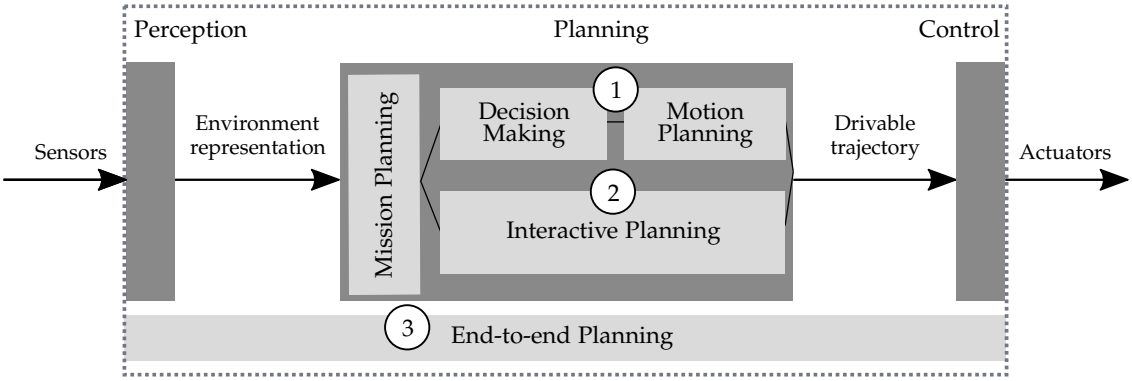


Figure 2.1.: Planning approaches within simplified functional architecture of AVs (inspired by Schwarting et al. [15]). The perception layer creates a fused environment representation. The planning layer performs higher level mission planning and either 1) uses combined decision making and motion planning or 2) interactive planning to generate a drivable trajectory passed to the control layer. In contrast, end-to-end approaches 3) learn to drive directly based on sensor data.

[75, 76]. Due to substantial difficulties in achieving safety with such approaches [77], they are not further reviewed here.

The planning layer receives the fused environment representation and additional map information. It commonly consists of a mission planner managing the route from the current location of the Autonomous Vehicle (AV) to the routing target [61]. Given this route and the environment representation, a subsequent layer computes the drivable trajectory in a receding horizon fashion, limiting the required duration and length of the planned motion [15].

Planning for AVs has a large body of research and industrial applications [15, 78, 79]. However, no go-to solution exists for finding a safe, comfort-optimizing trajectory irrespective of the current traffic situation and integrating the various types of perception and behavior uncertainties. Existing functional architectures and methodological variants of the planning layer divide into two major concepts [15]: 1) Interactive behavior planning and 2) decision making and motion planning. In the latter approach, the decision making component decides for a maneuver class, i.e., homotopic variant [80, 81] for which the motion planner generates a drivable trajectory (cf. Fig. 2.2). It is thus also frequently referred to as maneuver-based planning. A popular variant in this category is path-velocity decomposition. The decision-making computes a static path describing a track the AV should follow [78]. The resulting path does not integrate the time domain. To adhere to the dynamic constraints of the vehicle, the motion planner generates a drivable trajectory closely following the preplanned path [82]. Path-velocity decomposition is especially meaningful in static environments, e.g., in valet parking [83, 84]. Though variants exist to apply this concept in dynamic environments, it is limited to slow-moving obstacles, e.g., pedestrians [56].

In highly dynamic environments characterized by faster moving objects, it becomes necessary to directly plan a feasible trajectory within the allowed free space in the current planning horizon, considering the time-dependency of motions. The decision making layer then passes also dynamic information with the maneuver variant [85] and evaluates prediction uncertainty to select a homotopy [81]. Assuming that other vehicles' motions are independent of the ego

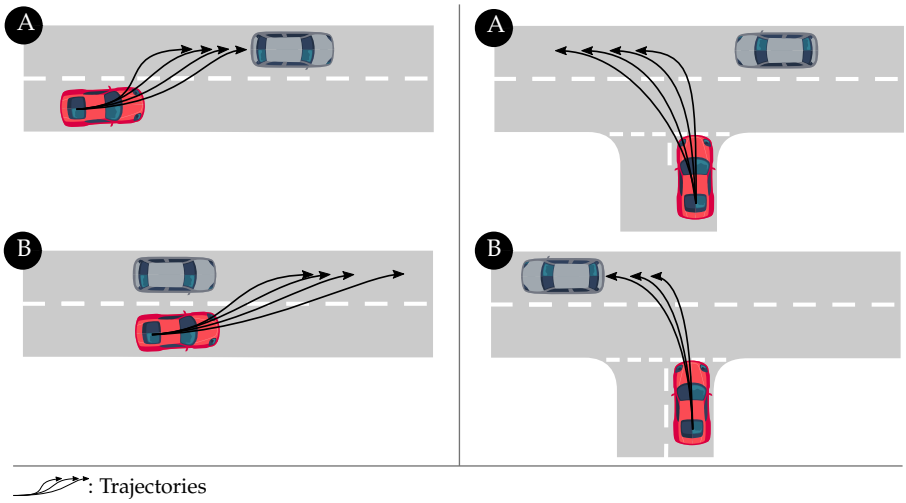


Figure 2.2.: Homotopic variants A and B in two traffic situations. Trajectories belong to the same homotopic variant or maneuver class if there exists a continuous mapping to transform between the trajectories [80].

vehicle's movements within the planning horizon, the time-state space divides into higher-cost areas that include collision states and regions with proximity to other participants as well as regions of lower cost. The planner then finds an optimal cost-minimizing trajectory using optimization [71, 86, 87], sampling [88] or selection from a predefined trajectory set [89]. Separating the selection of maneuver variants and the planning process greatly simplifies the resulting planning problem, which lowers computational demands of maneuver-based planning. However, dense traffic impedes selecting the best homotopy without the availability of the planned motion. Additionally, with increasing traffic density, the available low-cost planning space is reduced or even vanishes, restricting the reasonable trajectories to a set of conservative driving motions.

Interactive planners model the interactions between traffic participants during the planning process [50]. Stating how the ego-motion affects others' reactions and this, in turn, the available drivable space reveals an increased solution space for finding a feasible plan. In general, the evaluation of reactions of other participants during planning increases computational demands compared to maneuver-based planning [15]. However, as discussed in Sec. 1.1, interactive planning is a necessary pathway to allow for seamless integration of AVs into dense traffic. The following sections discuss related work on interactive planning divided into how approaches interactively predict, plan, balance safety and efficiency, and improve computational feasibility in real-time applications.

2.2. Interactive Prediction

Planning algorithms require an accurate prediction model of other traffic participants [90]. Employed models must cover the variations in human driving behavior. These behavior variations arise at different abstraction levels. Human driving intents define the near-term goals of a driver on a higher level with a discrete set of labels. Thereby, a near-term goal is defined either for the

route of the ego vehicle, e.g., by modeling intents "turn left" or "go straight" [91], or concerning another traffic participant, e.g., with intents "give way" or "take way" [55]. Apart from higher-level goals, humans show two forms of continuous microscopic behavior variations, intra-driver, and inter-driver variability [92]. The intra-driver variability comprises the time-dependency of the behavior of a single driver, whereas the inter-driver variability includes differences of driving styles between humans [93]. Both forms of variations are subject to randomness and vary in a continuous manner [94]. Fig. 2.3 visualizes the relations between these concepts. Maneuver-based prediction focuses on applications in maneuver-based planning approaches [39, 81, 95, 96] and is not further reviewed here. Prediction usable in interactive planning requires a meaningful definition of the microscopic behavior of other participants and must take low computational resources. Given these requirements, the prediction concepts employed by existing interactive planners differentiate into three major directions.

Intent-based prediction tracks beliefs over intents of other drivers using different forms of Bayesian inference. Pedestrian intents are modeled as hidden goal positions in a discretized state space in [56] or using pose classification in [97]. Other works define driving intents over preferences of lanes and track them using a Bayesian network [98, 99], or over route preferences in an intersection and track beliefs with a Bayes classifier assuming Gaussian-distributed observations of routes depending on the vehicles' locations [100]. Apart from intents related to map properties, intents can also model the interactions between participants, e.g., yield and take way, and be detected using probabilistic classifiers trained from data [55, 101]. An interaction-aware model integrating lane and route preferences and maneuver-based interaction is proposed in [102]. Joint inference of ego and other vehicles' intents can resolve the interdependence between intents [51]. Beliefs over predefined high-level policy types, e.g., modeling gap keeping or turning, are obtained using a change point detection algorithm in [103]. In [104], the authors define policy types using a set of manually specified Kalman filters and predict with the Kalman filter giving maximum posterior belief. Intents defined with temporal logic are used in [105]. To define the microscopic behavior of a specific intent, the presented methods apply deterministic predictions, e.g., car-following models [106] or add Gaussian noise to the predicted actions [55, 56, 100, 102], with parameters learned from data in [98], or they neglect microscopic randomness at all [101, 103, 105]. Intent-based models focus on the first level of variations depicted in Fig. 2.3. They do not explicitly model inter- and intra-driver variations nor adapt the microscopic prediction to other observed drivers. Further, human intents are not physically measurable, making specification of ground truth labels ambiguous. Though studies for ground truth labeling [107] and integration of detected intent cues into motion planning [108] exist, the conduction of such studies becomes infeasible for all types of intents and traffic scenarios.

Reward-based prediction models assume that participants act according to a reward function. Global reward functions assume that each participant acts such that, depending on a cooperativeness level, the reward of other participants is also maximized [51–53, 109, 110]. Independent reward modeling assumes that the autonomous car and human drivers individually maximize their rewards [28, 111]. In reward-based prediction, the microscopic motion model implicitly arises by respecting the reward functions and based on the action space of other participants

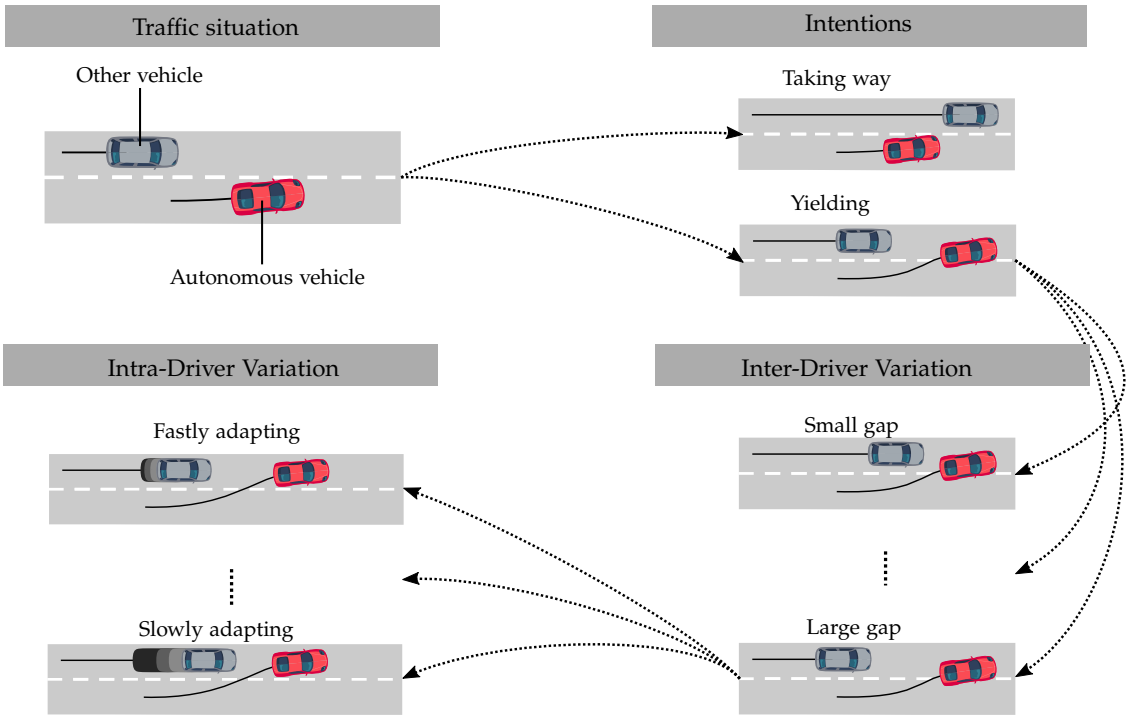


Figure 2.3.: Intents, inter- and intra-driver variations in human driving behavior. During a lane change of an AV another vehicle can have two intentions. How yielding is performed depends on the driver which leaves differing gap sizes varying continuously between smaller and larger gaps. An individual driver may then choose to adapt the desired distance to the AV over time and show continuous variations in between fast and slow adaptations.

being continuous [109, 112, 113], mainly discrete [52, 53] or making use of higher level policies over lower level actions [114]. The combination of cooperative models with intent-based concepts is frequent. Inverse reinforcement learning can be used to learn weights of a linear reward function to define distracted and attentive human driver types [28, 29]. The intents, yield and no yield, define the action space in a cooperative model in [101], yet, with the underlying microscopic models being deterministic. In [115], the authors assume the correctness of a cooperative model if all meaningful reward parameterizations lead to the same homotopy class for each other driver, respectively. If not, they trigger a fallback safety plan. Adapting to and predicting microscopic behavior in reward-based models is accomplished by tracking beliefs over the weights of a reward function [111] or the cooperativeness level [116], or by defining specific parameters, e.g., to represent the aggressiveness of other drivers, in the reward function [51]. Probabilistic microscopic driver models can improve search tree exploration in cooperative planning [117, 118]. However, how such exploration enhancements soften the assumption of a global reward function remains unclear. At the beginning of the search, the microscopic model dictates the explorative behavior. Later in the search, the influence of reward-based exploration starts to take over. Overall, reward-based prediction reduces the variety of human intra- and inter-driver variations into a single or small set of parameters. Reward functions can be tuned based on past observations [111, 119] and studies show coincidence of human merging behavior with the assumption of global reward maximization [110]. However, the random nature of

microscopic human behavior is not expressed using deterministic reward functions.

Neural networks are capable to predict trajectories [120–122] and actions [91]. Some of these approaches are impractical for interactive planning due to applying end-to-end concepts [122] or the computational demands of neural network inference [120, 122]. A learning-based interaction-aware microscopic model is proposed in [91] and suggested by the authors to be used in interaction-aware planning approaches. Using a small neural network allows for faster inference, and the approach demonstrates to model subtleties in human driving variation. Nevertheless, it predicts the action of only a single agent and does not integrate past state information into the prediction. In [123], the authors apply inverse hierarchical reinforcement learning to represent intents and the underlying trajectory distributions in a reward-based prediction approach.

Another class of prediction approaches employs classical driver models, e.g., used to simulate car-following and lane-changing [106, 124]. Instead of using offline calibration [125], the approaches apply online adaptation of the continuous model parameters to achieve more accurate predictions. Bayesian approaches to track parameter values are used in [126] estimating a single parameter for the pedestrian walking direction. Estimating multiple parameters of car-following models is performed in [127] in a framework of Bayesian reinforcement learning, in [128] using regularized regression and in [129–131] using particle filtering. However, these models track the probability that the past observed actions of other participants are expressed with *one* set of parameters. Intra-driver variations are therefore not genuinely represented with such approaches. To account for intra-driver variations, i.e., changing model parameters of a single participant over time, it requires estimation of the probability of *ranges* of parameters leading to the observed past actions.

Overall, state-of-the-art methods focus on different aspects of human behavior predictions. Defining risk for behavior uncertainty requires a comprehensive prediction model integrating both inter- and intra-driver variations, including adaptation to past observed states. Chapter 3 presents a prediction model to target these requirements since they are only partially covered by existing approaches.

2.3. Planning Integrating Interactions

Various terms exist to express that a planning algorithm takes into account the reactions of other drivers during planning, e.g. interactive [50], cooperative [52, 53, 109, 112, 132], collaborative [110], social-aware [111] or courtesy-aware [101]. Though, in multi-agent concepts collaboration refers to cooperation without prior knowledge [133] both terms are frequently employed to model the egoistic planning perspective in autonomous driving [52, 53]. This thesis refers to approaches under these terms in the following as interactive planners.

In general, two levels of driver reactions exist. Interactions at the homotopy level model *if* other drivers react to maneuvers of the AV, e.g., consider *if* another participant can yield if the ego vehicle touches its lane proactively. Reactions within a homotopy represent *how* other participants react to microscopic movements of the ego vehicle, e.g., consider *how* close other drivers approach when the ego vehicle touches its lane proactively.

There exist planning approaches which model interactions only at the homotopy level. Compared to rule-based approaches for homotopy selection (cf. Sec. 2.1), interactive variants select the optimal *sequences* of homotopic variants according to the combined expected utility [101], by using Monte Carlo sampling [103] or hierarchical Monte Carlo Tree Search (MCTS) [53]. Other approaches model partial observability within Partially Observable Markov Decision Process (POMDP) planning to select between different maneuver variants [39] or define maneuver sequences using Linear Temporal Logic (LTL) in a belief Markov Decision Process (MDP) solved with value iteration [105]. Both levels of interactions are considered in [134] separately by using MCTS to define a higher level option policy over lower-level learned control policies. Other concepts do not *explicitly* integrate reaction into the planning process. Instead, interactivity is integrated by the use of an interactive prediction concept. In [111] the authors use the Model Predictive Control (MPC) approach presented in [135] and parameterize the reward function based on intent-specific beliefs. A similar approach is presented in [136]. Monte Carlo sampling of trajectories distributed based on beliefs of microscopic behavior parameters and selection of the reward-maximizing trajectory is used in [90].

An interactive planner should evaluate how its generated plan creates and affects available homotopies in the current traffic situation to solve dense traffic situations. The above interactivity concepts relying only on prediction or homotopy selection do not satisfy this requirement. When separately planning on both levels, it is cumbersome to define meaningful homotopy types. Therefore, a large body of work deals with a combined planning paradigm. These approaches can be categorized into probabilistic and multi-agent concepts and are presented in the following.

2.3.1. Probabilistic Methods

Probabilistic planning searches for optimal plans that maximize the expected reward or cost over predicted future environment states. Since the standard probabilistic decision models assume that a single agent interacts with the environment, other traffic participants' random behavior is represented by stochastic environment transitions. Intent-based and microscopic prediction models (cf. 2.2) are used in conjunction with well-known sequential decision-making frameworks under this paradigm, as detailed in the following.

The MDP assumes that transitions to the next environment state depend only on the current environment state and the applied action [137]. It is the fundamental decision-theoretic concept in Reinforcement Learning (RL), a framework to learn an optimal sequence of decisions by repeatedly interacting with the environment [138]. Using MDPs for interactive planning predicts other participants only conditioned on the current environment state. The combination of Deep Learning (DL) with RL allows for offline model-free learning of optimal plans in continuous environments [139, 140] leading to a variety of planning approaches for AVs using Deep Reinforcement Learning (DRL) [131, 134, 141–146]. To reduce the errors in the optimality of the plans obtained with DL, value iteration in discrete state spaces is applied in [57] or regression methods in [147]. A drawback of using MDPs for planning for autonomous vehicles is that MDPs do not model partial observability of the environment state. However, the future traffic situation can often be more accurately predicted by taking into account hidden information, e.g.,

about the intents of other traffic participants. Though recurrent neural networks can be trained to infer hidden information from past states [148], their implicit representation of beliefs lacks interpretability. Further, the learned policy implicitly assumes the behavior of other participants simulated in the training process to be the microscopic prediction model. For obtaining interpretability of the risk definition, it must be understood which behavior variations have been encountered during training. However, this is not straightforwardly established with offline training in simulation.

Intent-based prediction is often used with Partially Observable Markov Decision Process (POMDP) planning integrating beliefs over intents. POMDPs model sequential decisions under partial observability of the true environment state [137]. Different variants of belief-state planning exist which incorporate beliefs into the planning process. An extensive overview is given in [149]. Especially relevant to this thesis is QMDP planning. It samples environment states from the current intention belief and predicts the future environment states based on these sampled states [130, 150, 151]. Predicting how beliefs change in future environment states allows the planner to anticipate the information value of future states. This generates so-called information gathering behavior [50]. Such behavior is meaningful for resolving situations in which the intention belief is ambiguous. Since these approaches are computationally demanding, the complexity of the problem is often reduced by planning with only longitudinal actions [56, 126]. A real-time capable POMDP planner for different traffic situations also including perception uncertainty is proposed in [50]. Learning-based approaches which integrate beliefs as input to a neural network have advantages regarding computationally feasibility [116]. Presented approaches use intention-based prediction models. Beliefs over behavior parameters are included in [130, 152]. However, existing POMDP planning approaches cannot plan given beliefs over continuous ranges of parameters as they occur when modeling intra-driver behavior variations. Optimal decisions under unknown parameters of the transition function in a MDP can be represented as Bayes-Adaptive MDP (BAMDP). In [127], the authors use this paradigm to represent unknown microscopic behavior parameters in an intersection scenario and solve it offline by framing the problem as discrete POMDP. Yet, the approach requires offline planning in discrete state spaces and omits time-variations of behavior parameters.

Overall, the benefit of probabilistic approaches is their natural integration of behavior uncertainty into interactive planning. Yet, presented concepts show deficiencies concerning the integration of beliefs over microscopic behavior variations.

2.3.2. Multi-Agent Methods

Multi-agent planning explicitly models each traffic participant as an agent within a multi-agent environment or multi-agent game. In contrast, to decision-theoretic frameworks, which focus on the self-interested decision-making of humans and bounded rationality due to limited planning time or cognitive capabilities, game-theoretic concepts primarily assume that all agents act rationally concerning the equilibrium of the game [153, 154].

In interactive planning for AVs, game-theoretic methods are the fundamental principle to plan under reward-based prediction concepts. Thereby, the reward functions define a non-zero-sum

game, i.e., a game with no clear winner. After solving the equilibrium strategies for all agents, the strategy of the agent representing the AV defines the optimal ego vehicle plan. The computational efforts to obtain these strategies rise with the number of agents due to the interdependence of strategies. In [155], the authors, therefore, employ a Stackelberg game formulation which assumes that one agent suggests a strategy and a single other agent adapts. The optimizations are solved for higher-level actions and passed to a lower-level MPC planner. A similar two-agent formulation is used in [29]. Another two-player concept using alternating gradient steps over cost maps to model driver interactions is presented in [30]. In [119], the authors employ an iterative method to resolve the leader-follower problem in a multi-agent Stackelberg game. Belief tracking over a set of possible local equilibria is used to estimate the currently dominating equilibrium in [156]. Presented methods support non-linear vehicle dynamic models. However, this requires iterative approaches to solve the non-linear optimization problems.

In contrast, Mixed Integer Programming (MIP) allows to find optimal global solutions in multi-agent problems [113], yet, it requires a linearized representation of the optimization problem. Kessler and Knoll [112] sample independent motion trees for each participant using a discrete set of motion primitives and use Mixed Integer Linear Programming (LP) (MILP) to solve for optimal ego behavior. Planning in continuous action spaces with MIP requires linearized vehicle models. In [109], the authors assume straight road segments and employ a triple integrator as a vehicle model to be able to solve for optimal plans using Mixed Integer Quadratic Programming (MIQP). A linearized bicycle model is presented in [157] and integrated in [113] into a linear differential game. It is solved using MIQP to plan in a cooperative racing task under arbitrary road curvatures.

However, current MIP formulations and solvers are limited when it comes to modeling uncertainty. In [158], the authors formulate non-cooperative driving as a multi-agent dynamic game in belief space and solve it using iterative Linear Quadratic Gaussian control (iLQG). By using a linear formulation of the belief updates with Gaussian observation and dynamic models, the complexity of the solver scales only linearly with the planning horizon compared to exponential scaling with point-based POMDP solvers [158]. Nevertheless, the approach is limited to simple belief representations that cannot represent non-linear belief updates over parameters of microscopic behavior models. Intention parameters are integrated into a Bayesian Game formulation in [51] and solved using non-convex optimization, yet, only in a discrete domain.

Arbitrary transition dynamics can be used with sampling-based multi-agent planning. Monte Carlo sampling of velocity trajectories along predefined paths is used in [115]. Better exploration of larger search spaces is achieved with MCTS having a long history in game-playing algorithms [159]. In its multi-agent variant, denoted Simultaneous-Move MCTS (SM-MCTS) [160], agents independently select actions in stages of the game. Lenz et al. [52] first apply this concept to cooperative planning for AVs. Progressive widening of the discrete action set is applied in [114] to target the problem of discrete action spaces. Compared to optimization-based approaches which provide tighter bounds on optimality [113], SM-MCTS only eventually converges to the optimal global solution given infinite search time [160]. However, global optimality is only beneficial if the assumption of other agents acting rationally in the game transfers exactly to reality.

This is an unprovable case in applications of AVs given the uncertainties about other participants' behaviors (cf. Sec. 2.2). SM-MCTS supports arbitrary transition dynamics and benefits from a decoupled action selection mechanism compared to single-agent probabilistic planning approaches. However, presented work using SM-MCTS misses the integration of randomness in the transition functions to be able to integrate the microscopic behavior variations.

Overall, state-of-the-art interactive planning lacks a holistic integration of inter- and intra-driver behavior variations. In chapter 3, this thesis therefore proposes, the Robust Stochastic Bayesian Game (RSBG) a game-theoretic model integrating beliefs over inter- and intra-driver behavior variations and a novel SM-MCTS planner to solve this model.

2.4. Interactive Optimality Criteria

As discussed in Sec. 1.1 there exist non-probabilistic and probabilistic ways to balance safety and efficiency in planning. In interactive planning, both approaches find frequent use.

Optimization-based interactive planning requires differentiable cost criteria and therefore often applies linear combinations of differentiable cost terms [28–30, 155]. Frequently employed terms are Gaussian functions expressing the distance to other vehicles [28, 29], quadratic deviation of desired speed [28], distance functions to model road and lane boundaries [29] or more complex spatio-temporal cost maps [30]. The linear weights of these cost terms are often extraced from data, e.g., by using Inverse Reinforcement Learning (IRL) [29, 111, 119]. A chance-term weighting the collision probability is defined in [158] based on beliefs over position uncertainty. In MIP planners, logical [109] or soft constraints [113] are used to prevent collisions, and combined with additional constraints on the satisfaction of longitudinal and lateral safe distances [161]. Presented optimization-based interactive planners use non-probabilistic ways to balance safety and efficiency since the evaluation of the optimality criteria is not informed by the probability that a high- or low-cost event occurs. These approaches do not express the probabilistic nature of an interpretable risk metric. Thus, they are not suited to achieve the interpretable risk formalism proposed in Sec. 1.2.

Sampling-based interactive planners frequently use manually tuned rewards and costs to balance collisions and goal-directed plans [52, 55, 127, 136]. Goal-directed plans are expressed by penalizing deviations from the desired velocity [52, 55], deviations from the lane center [55], large accelerations [52, 55] and an incorrect lane position [55]. Safety is expressed by giving large negative rewards for collisions [52, 55], including longitudinal distances to other participants [52] or assigning negative rewards if a physical safety measure, e.g., the Time-To-Collision (TTC), falls below a threshold [162]. Sampling-based, e.g., POMDP, SM-MCTS and RL planners, use a probabilistic approach to balance safety and efficiency. They consider the probability of state transitions and accompanying rewards and costs in their expectation-based optimality criteria [137]. The actual probabilities depend on the transition model given by the employed interactive prediction model (cf. Sec. 2.2) and the exploration strategy used for sampling or during offline training. Due to the use of *single-objective* optimality presented methods subsume safety and efficiency criteria into a single value. The reward and cost criteria and the probability that

the planned motion leads to unsafe states are connected to the expectation of future states. However, other factors such as discounting future rewards and costs and the dependence of the optimal planned motion on the amount of exploration affect the balance of safety and efficiency unpredictably.

Other methods to integrate the safety criteria into an interactive planner use online verification to monitor an RL planner, even supporting active learning in real driving situations [144]. However, by strictly prioritizing safety, the underlying interactive planning approach becomes redundant, and the drawbacks in dense traffic discussed in Sec. 1.1 arise. A similar monitoring approach using Responsibility-Sensitive Safety (RSS) for planning at the homotopy level is proposed in [115]. In [130], the authors include collision-free motions into the action space of a POMDP planner. Robust control approaches consider worst-case outcomes in their optimality definition and are used in [163] to plan under the presence of inter-driver behavior variations. However, these approaches rely on single-objective optimality. Model-checking approaches maximize the probability of a plan to satisfy an automaton specification [54] or constrain the probability that actions violate an LTL specification [57]. Nevertheless, both approaches apply discrete state spaces. For continuous state spaces, previous work on risk-constrained interactive planning focuses on finding an optimal sequence of maneuver variants at the homotopy level with risk defined over collision events [39, 164, 165].

Overall, existing optimality definitions for interactive planning prevent integration of an interpretable risk formalism (cf. Sec. 1.2). In chapter 4, this thesis therefore proposes the Risk-Constrained Robust Stochastic Bayesian Game (RC-RSBG), a game-theoretic model which integrates an interpretable risk formalism over uncertainty of inter- and intra-driver behavior variations.

2.5. Learning and Performance of Interactive Planning

To integrate the reaction of other participants during planning, interactive planners must evaluate a large number of potential plans. The number of plans increases exponentially with the length of the planning horizon and the number of traffic participants whose reactions should be considered (cf. Fig. 2.4). Considerations of computational performance, therefore, play a crucial role in the development of interactive planners.

Offline planning avoids computationally demanding evaluations of all potential plans during online planning. Instead, it aims to precompute optimal plans for specific scenarios in advance. Offline planning requires a suitable representation of state space and precomputed plan to recover an appropriate optimal plan during the online planning and execution step. Frequently, discrete state spaces and a tabular representation of the plan are used, in BAMDP [127], constrained MDP [57] or multi-agent planning [51]. In [57], the authors suggest linear or nearest-neighbor interpolation to extend tabular representations to continuous state spaces. In [99], the authors use a decision tree to represent the state space with a finite set of α-vectors in a POMDP value iteration algorithm. Online, they select the dominating α-vector based on the current belief. Partial online replanning to account for inaccuracies in the precomputed

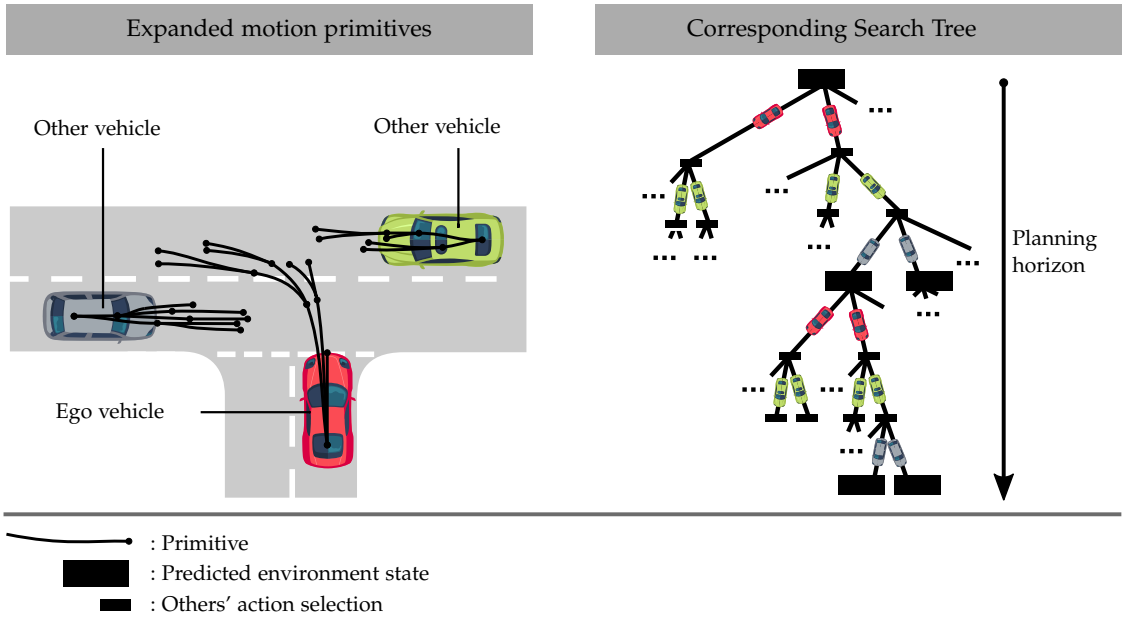


Figure 2.4.: Exponential complexity of sampling-based interactive planning. The number of possible joint actions of ego and other vehicles increases exponentially with the number of traffic participants. The number of possible future state sequences, i.e., possible plans, increases exponentially with the length of the planning horizon over the size of the joint action space.

plans is presented in [39]. The approach maintains the search tree for the online phase to avoid a discrete representation. However, the authors leave open how to deal with discrepancies of observed and precomputed states. Overall, a discrete model of the state space and plan becomes problematic in dense traffic since a slight change of state can affect the optimality of a plan. The significant difficulty with such offline planners is thus selecting an appropriate state and plan representation.

Model-free RL in simulation is a form of offline planning. The use of neural networks to represent a learned policy of an AV allows for continuous state [142–145] and action spaces [166]. Online planning then benefits from neural networks' relatively low inference time to evaluate the learned policy online. Multiple inferences of the policy network may be required to obtain a trajectory [167]. Even though DRL overcomes the problem of discretization in offline planning, generalization of learned policies to arbitrary road layouts and traffic situations is still ongoing research, e.g., in [141].

Strict online planning does not perform precomputation and applies different strategies to allow for real-time capability. Exploration and guidance towards optimal plans is improved by using denser [52, 55] and continuous [28, 29] reward specifications, by grouping similar actions [114] or by using larger prediction time spans [99]. Reducing the size of the action space [150], considering only longitudinal actions [56] and applying path velocity decomposition [115] decreases the number of combinatorial options required to be evaluated by the planner. Other strategies to reduce computational demands are to replace collision checks based on polygonal hulls with an approximation of shapes using circles [112, 113] or to shorten the planning horizon to the near future as, e.g., in [119]. Previous approaches are applicable irrespective of the applied

planning approach.

Other strategies are applied only with specific planning methods to improve their real-time capability. In receding horizon planning, initialization of the following planning step reuses past calculations, which also helps to keep consistency between subsequent plans [50]. In sampling-based approaches, this resorts to reusing parts of the search tree [50] whereas warm starting can be used in optimization-based planning [113]. Sampling-based approaches can benefit from multiple resources, e.g., CPU cores, by running various searches in parallel [168] or parallelizing over a single tree requiring mutexes to lock critical paths [52].

Combinations of offline learning with online tree search achieved super-human playing strength in the game of Go [169, 170]. In the context of interactive planning for autonomous driving, related approaches exist to reduce computational demands and improve the anytime capability of sampling-based planners. Microscopic prediction models are learned from human driving data to guide exploration in a SM-MCTS planner in [117, 118]. A combined approach to learning action probabilities and value functions in a reinforcement learning fashion while generating demonstrations in simulation is presented in [171] for POMDP planning. A similar approach in an end-to-end planning framework using grayscale images as input is presented in [172].

Overall, achieving real-time capability with interactive planners requires combining the beforementioned concepts dependent on the planner type. However, it remains unclear how existing concepts transfer to multi-objective interactive planning. Chapter 5, therefore, proposes performance improvements for multi-objective, i.e., risk-constrained planning utilizing root parallelization and offline learning.

3 Game- and Robustness-Based Interactive Planning in Behavior Spaces

This chapter proposes an interactive planning algorithm for AVs which leverages probabilistic predictions of microscopic intra- and inter-driver behavior. It incorporates beliefs over behavior hypotheses into a game-theoretic formulation, the Robust Stochastic Bayesian Game (RSBG) and solves it using Simultaneous-Move MCTS (SM-MCTS). The main contributions of this chapter are

- the formulation of the interactive planning problem as Stochastic Bayesian Game (SBG) modeling irrationality in human driving behavior,
- the design of behavior hypotheses for the Stochastic Bayesian Game (SBG) using partitioning of behavior spaces for accounting for inter-driver behavior variations,
- the definition of sum posteriors over behavior hypotheses to model intra-driver behavior variations,
- a novel game-theoretic model, the Robust Stochastic Bayesian Game (RSBG), which extends the Stochastic Bayesian Game (SBG) with robustness-based optimality to reduce sample-complexity when planning in continuous behavior spaces,
- a variant of SM-MCTS to approximately solve the Robust Stochastic Bayesian Game (RSBG).

The work presented in this chapter is based on [173]. The chapter starts with a review on ad-hoc coordination in multi-agent systems in Sec. 3.1. The problem of interactive planning and prediction using the SBG is developed in Sec. 3.2. Sec. 3.3 presents behavior spaces for interactive behavior prediction. The integration of robustness-based optimality into the SBG is given in Sec. 3.4. Finally, Sec. 3.5 develops the RSBG planner using SM-MCTS.

3.1. Review on Ad-Hoc Coordination in Multi-Agent Systems

The problem of interactive planning for autonomous driving can be represented as a non-cooperative multi-agent system (cf. Sec. 2.3.2). The Autonomous Vehicle (AV), thereby, is an

intelligent agent, interacting with to a large extend unknown other agents, the human drivers. Background on the SBG which models interactions with irrational agents is given in the next section. The following section then describes existing methods to define the behavior of other agents in the SBG using agent behavior types.

3.1.1. Introduction to Stochastic Bayesian Games

The problem of on-the-fly interaction with unknown agents is introduced as ad-hoc coordination in [133]. Among several approaches to solve the ad-hoc coordination problem [174, 175], the type-based method has shown to be particularly useful. It uses a predefined set of agent types. Each type commonly maps observation histories to probabilities over actions allowing to track posterior beliefs over types by incorporating the probability of actions under each type [133, 176]. Albrecht and Ramamoorthy [154] formalize the type-based approach for solving the ad-hoc problem as SBG. The model combines stochastic games, representing uncertainty in environment transitions, with Bayesian games. In the SBG, one controls a single agent, which uses a hypothetical type space to reason about the behavior of other agents. Modeling planning for AVs as SBG comes with two major benefits:

- **Integration of belief information:** Game-theoretic models assume that at some point in the game, the strategies of all agents form a Nash equilibrium. At equilibrium, no player can benefit from switching strategies. However, since multiple equilibria can exist, a form of coordination, e.g., by communicating planned actions between agents, may be needed to decide on a common equilibrium strategy [177, 178]. Bayesian games integrate additional information in the form of beliefs over agent types [179]. Providing additional information based on beliefs of agent types resolves the coordination problem. An example in autonomous driving is belief tracking over reward functions (cf. 2.2).

- **Modeling of irrationality:** Assuming that humans drive strictly rational according to game-theoretic payoff functions is questionable given that humans act irrationally in many other parts of their life [180], e.g., in economic activities, they act under incomplete instead of rational contracts [181]. In contrast, to Bayesian games, the SBG assumes that the type space of other agents is unknown. By using a hypothetical type space, not necessarily defined using payoff functions, the SBG models and adapts to the irrational behavior of other agents.

Albrecht and Ramamoorthy [154] introduce the Harsanyi Bellman Ad-Hoc (HBA) algorithm to plan optimal policies for the SBG. However, they focus on discrete state and action spaces. Since modeling microscopic behavior variations requires continuity in both state and actions, this thesis proposes a new optimality definition in Sec. 3.4 to improve sample efficiency of the SBG when planning for AVs.

3.1.2. Review on Designing Agent Type Spaces

Designing an appropriate type space, also denoted behavior hypothesis set or simply hypothesis set in the remainder of this thesis, is crucial to ensure the validity of the SBG for modeling interactions with unknown agents. Previous work frequently uses small hypothesis sets in simpler domains defined by domain experts [133, 154]. Intention prediction of other drivers (cf. Sec. 2.2) in a multi-agent setting [28] can also be regarded as using discrete sets of behavior hypotheses. Integrating continuity into behavior hypotheses can be broadly categorized into approaches using a parameterized set of hypotheses or learning the hypothesis set on the fly during task completion. Methods in the former category either build a hypothesis set by sampling hypothesis out of a parameterized hypothesis space [182] or adapt online the parameters of a predefined set of hypotheses [176, 183]. However, such methods only consider a single parameter set for each hypothesis and do not model types which *cover* a specific part of the parameter space. With Q-learning, [174] or decision trees [184] the hypothesis set can be adapted on the fly avoiding the definition of a continuous hypothesis model. However, an online adaptation of the hypothesis set is impractical when the task is characterized by short interaction times as given in interactions between human drivers.

Type-based methods provide an expressive way to model unknown behavior in a multi-agent setting. They allow for fast adaptation to observed behavior and modeling the randomness of microscopic behavior variations using stochastic types. However, in order to apply the type-based approach to comprehensively model inter- and intra-driver behavior variations in interactive planning, the designed type space must fulfill three key properties:

- **Coverage of Variations:** Microscopic behavior variations of human drivers are governed by hidden factors, e.g., unknown parameters of the desired velocity or the desired distance to leading vehicles [93]. Such hidden decision factors relevant to the behavior of an agent in a specific situation must be included in an agent model. For this, the developed agent model must reason within the space of possible decision factors. That means that for type-based methods, the set of behavior hypotheses must cover all of these factors. Therefore, this chapter presents a hypothesis design process in Sec. 3.3.3 over behavior spaces motivated in Sec. 3.3.1 to cover the hidden factors defining microscopic behavior variations.

- **Independence of Intents:** The developed agent model must be independent of the higher-level goals of the agents to avoid the definition of a goal space. For instance (cf. Sec. 2.2), the intentions of human drivers are not directly measurable and are only estimated indirectly, e.g., based on observed driving trajectories. The absence of measured intents prevents the definition of an all-encompassing space of intents. The causal model of microscopic behavior presented in Sec. 3.3.2 avoids that the agent model depends on intents.

- **Time-Changing Behavior:** The developed agent model must support modeling of changing behavior over time to account for intra-driver variations. Using the sum posterior to estimate the beliefs over types can be interpreted as an *or* combination of past observations. Sec. 3.3.4 shows that such a belief estimation accounts, in addition to inter-driver variations, also for intra-driver variations.

Presented properties are related to the desirable properties of agent models proposed in [185].

3.2. Interactive Planning as Stochastic Bayesian Game

This section formalizes the problem of interactive planning for AVs using the Stochastic Bayesian Game (SBG). It starts with the definition of the SBG, followed by separate problem definitions for interactive prediction using behavior hypotheses and the planning of optimal policies under the SBG. Lastly, this section discusses the assumptions coming with the proposed formulation.

3.2.1. Model Definition

The proposed model considers traffic environments structured by lane and road boundaries with a certain number of vehicles within the field of view of the AV. Examples of two scenarios and the state space definition are depicted in Fig. 3.1. Interactive planning in such environments can be formally defined as SBG consisting of

- a set of N interacting vehicles or agents. Each agent j has a fully observable dynamic state $o_j^t = (x_j^t, y_j^t, vel_j^t, \alpha_j^t)$ at time t with x_j^t and y_j^t denoting the Cartesian coordinates, vel_j^t the velocity and α_j^t the orientation with respect to the center of the jth vehicle's rear axis. The ego agent is denoted with index $j = i$,
- a joint environment state $o^t = (o_1^t, o_2^t, \dots, o_N^t, M) \in \mathcal{O}$ fully observable by all agents including the current map state M consisting of road and lane layout and geometry,
- a continuous action space for each other vehicle $a_j^t \in A_j$ and the agents' joint action $a^t = (a_1^t, a_2^t, \dots, a_N^t) \in A$ with joint action space A. The ego agent uses the action space A_i,
- an environment transition function $T_{\text{env}} : \mathcal{O} \times A \to \mathcal{O}$ which defines the next environment state $o^{t+1} \in \mathcal{O}$ based on the current state $o^t \in \mathcal{O}$ and applied joint action $a^t \in A$,
- and for each agent j
  - a true type space $\theta_j^* \in \Theta_j^*$,
  - a true stochastic policy $\pi_j : \mathcal{H}_{\text{o}} \times \Theta_j^* \to [0, 1]$ which depends on the observation action history $H_{\text{o}}^t \in \mathcal{H}_{\text{o}}$ up to time t, $H_{\text{o}}^t = (o^0, a^0, o^1, a^1, \dots, o^t)$,
  - a reward function $u_j : \mathcal{O} \times A \to \mathbb{R}$,
  - and a prior true type distribution $\Delta_j^* : \Theta_j^* \to [0, 1]$.

The game starts at time t_{start}. Each agent samples randomly a type from its prior distribution $\theta_j^* \sim \Delta_j^*$. Iteratively all agents choose actions according to their polices $a_j^t \sim \pi_j(H_{\text{o}}^t, \theta_j^*)$ at each time step t and transition to the next environment state o^{t+1} according to the environment transition function. This process continues until a terminal criterion, e.g., the ego agent reaches a target lane, is satisfied.

This work implements a deterministic transition function using longitudinal accelerations along the lane center for other vehicles' actions. The ego action set consists of the motion

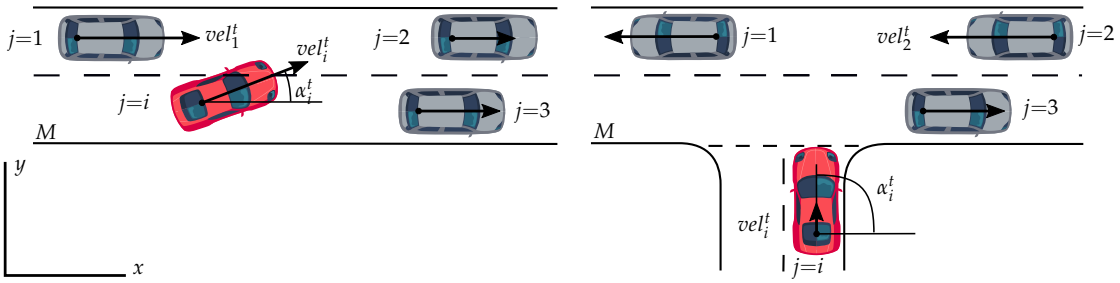


Figure 3.1.: Definition of the state space for the SBG. A collection of $N = 4$ agents is given in a merging and left turning scene. Vehicles can fully observe a joint environment state $o^t = (o_1^t, o_2^t, \dots, o_N^t, M)$ consisting of fully observable dynamic states $o_j^t = (x_j^t, y_j^t, vel_j^t, \alpha_j^t)$ at time t and static map information M.

primitive actions lane following with constant accelerations and lane changing implemented by a lane following controller assuming single track vehicle dynamics, and gap keeping based on the Intelligent Driver Model (IDM) [186].

3.2.2. Prediction Problem

The AV controls a single agent, i. It knows the action space and can fully observe past actions of the other agents and the joint environment state by having access to the observation-action history H_o^t. However, the true type space Θ_j^*, true policies π_j and prior true type distribution Δ^* are unkwown to the AV. Instead, the AV applies

- a hypothetical type space Θ and a set of stochastic policies for other agents $j \neq i$, i.e., behavior hypothesis $\pi_{\theta^k} : A^k \times \mathcal{H}_o \to [0, 1]$, $\theta^k \in \Theta$, $k \in \{1, \dots, K\}$ with A^k being the action space of hypothesis k. In the remainder of this thesis, the set of stochastic policies is also referred to as hypothesis set.

- a hypothetical prior type distribution $\Delta : \Theta \to [0, 1]$ which defines prior probabilities $P(\theta^k)$ of the hypothetical types for belief tracking. The prior $P(\theta^k)$ specifies the initial probability of type k before the start of the game.

Given the observation-action history for all other agents at time t, the planning component tracks posterior beliefs $\Pr(\theta^k | H_o^t, j)$ over the hypothetical types θ^k for each agent j. These beliefs in combination with the set of stochastic behavior hypothesis define a mixture distribution $\widehat{\pi}_j$ predicting the microscopic variations in human driving behavior for each other agent j:

$$\widehat{\pi}_j(a_j | H_o^t) = \sum_{\forall k} \Pr(\theta^k | H_o^t, j) \cdot \pi_{\theta^k}(a_j | H_o^t) \tag{3.1}$$

The difficulty lies in designing a hypothesis set and belief update to cover human drivers' intra- and inter-driver behavior variations. This chapter approaches this problem and presents behavior spaces for interactive behavior prediction in Sec. 3.3.

3.2.3. Planning Problem

Given the microscopic prediction model defined in Eq. (3.1), the planner must find an optimal policy $\pi_i : A_i \times \mathcal{H}_o \to [0,1]$ maximizing the reward of the ego agent u_i. Such an optimal policy in an SBG is defined with the HBA algorithm using a combination of belief-weighting and Bellman updates [154]. In the remainder of this thesis, an index $-i$ denotes all agents except i. Concatenation is also denoted at the index level, giving, e.g., for the joint action $a{=}a_{i,-i}$. The optimal policy of the AV according to the HBA algorithm [154] follows the optimality criterion $a_i^t = \text{argmax}_{a_i} E(H_o^t, a_i)$, where

$$E(H_o, a_i) = \sum_{\theta_{-i} \in \Theta_{-i}} \Pr(\theta_{-i}|H_o) \sum_{a_{-i} \in A_{-i}} Q_R(H_o, a_{i,-i}) \prod_{\substack{j \neq i \\ a_j \in a_{-i} \\ (\theta^k, j) \in \theta_{-i}}} \pi_{\theta^k}(a_j|H_o) \tag{3.2}$$

is the expected cumulative reward for agent i taking action a_i after observing the last state o in history H_o. The sum over posterior beliefs is thereby taken over the possible combination of types for all agents $\theta_{-i} \in \Theta_{-i}$ with $\Pr(\theta_{-i}|H_o) = \prod_{(\theta^k,j)\in\theta_{-i}} \Pr(\theta^k|H_o, j)$. The joint action space of other agents $A_{-i} = \bigtimes_{(\theta^k,j)\in\theta_{-i}} A^k$ is defined for a specific combination of types. The probability that a joint action of other agents a_{-i} arises is given by multiplying hypotheses action probabilities for each agent in the current combination of types θ_{-i}. The Bellman part of HBA is

$$Q_R(H_o, a) = u_i(o, a) + \gamma \max_{a_i \in A_i} E(\langle H_o, a, o' \rangle, a_i) \tag{3.3}$$

and defines the expected cumulative future reward of agent i when joint action a is executed in observation state o after history H_o. Future rewards are discounted by γ. The concatenation of action and observation to the previous history is denoted with $\langle\cdot\rangle$. This thesis employs a deterministic joint transition function. Therefore, the expectation over potential subsequent states o', as provided in [154], is dropped in this definition of $Q_R(\cdot)$.

Monte Carlo Tree Search (MCTS) can be used to find approximate solutions to this problem [187]. Yet, the HBA algorithm in Eq. (3.2) and (3.3) is defined for a discrete action space only. Finding an optimal policy is impeded when the action space of the other agents is continuous since this results in an infinite size of the joint action space. Thus, this thesis proposes a sample-efficient variant of the SBG, the RSBG, integrating robustness-based optimality, in Sec. 3.4, and an accompanying SM-MCTS to find approximate solutions for the RSBG. The presented approach focuses on planning under continuous action spaces of other agents. The ego agent uses a set of discrete actions as in the original HBA algorithm.

This chapter uses a single-objective optimality criterion assuming the reward function u_i encodes collisions and successfully reaching a goal state. The following Chapter 4 extends the planning problem to the multi-objective domain and includes the interpretable risk definition.

3.2.4. Model Assumptions

The problem formulation comes with the following assumptions. The SBG assumes full observability of the environment state. Therefore, the game-theoretic model applied in this thesis does not take into account perception uncertainty. The SBG supports stochastic environment transition functions. However, this thesis applies deterministic environment transitions. It, therefore, does not take into account execution uncertainties arising from deviations between planned and executed trajectory.

Several works consider perception [41, 50, 188] and execution uncertainty [189, 190] in planning. The Interactive Partially Observable Markov Decision Process (POMDP) (I-POMDP) is a multi-agent model which models *both* uncertainty in the observability of the physical states *and* uncertainty about the behavior types of other agents. This allows to express nested beliefs [149] such as "I belief that you belief that I will change lane.". Though, the I-POMDP is more general than the SBG, there are additional computational difficulties when solving the model for an optimal plan [153]. This thesis focuses on a systematic understanding of how uncertainty about microscopic behavior variations affects efficiency and safety. Integration of other uncertainty concepts is discussed as future work in Sec. 7.2.

The number of vehicles defining agents within the SBG is reduced to the $N-1$ vehicles nearest to the AV. Other vehicles are not considered in the state transition function, i.e., assumed to be non-existent. This thesis employs a discrete set of ego actions. Other work already introduced a variant of MCTS which enables continuous action spaces for the ego agent [114].

This thesis focuses on modeling vehicle-to-vehicle interactions and does not consider other traffic participants, e.g., pedestrians. It thus employs an agent-independent type space Θ and prior type distribution Δ reflecting that all hypotheses are equally likely for any human driver. In contrast, the original formulation of the SBG [58] uses agent-dependent definitions.

3.3. Behavior Spaces for Interactive Behavior Prediction

This section presents a concept that models the influences of intents and other hidden factors onto the microscopic intra- and inter-driver behavior variations. The concept designs behavior hypotheses by partitioning a *behavior space* thereby avoiding an explicit definition and labeling of driving intentions.

3.3.1. Motivating Example

Fig. 3.2 provides a motivation for behavior spaces. The AV V_{A} wants to change onto a lane being occupied by an oncoming rear vehicle V_{R} and another front vehicle V_{F}. A common, yet unsatisfying, approach in interactive prediction (cf. Sec. 2.2) is to model the intent of the rear vehicle V_{R} as yielding or taking way and then incorporate this information into planning.

The average human would associate an intent "yielding" with opening up a gap for another vehicle, and the intent "not yielding" with the contrary of not opening a gap. To avoid the explicit definition of intents, one can look for other cues which are sufficient to predict the evolvement

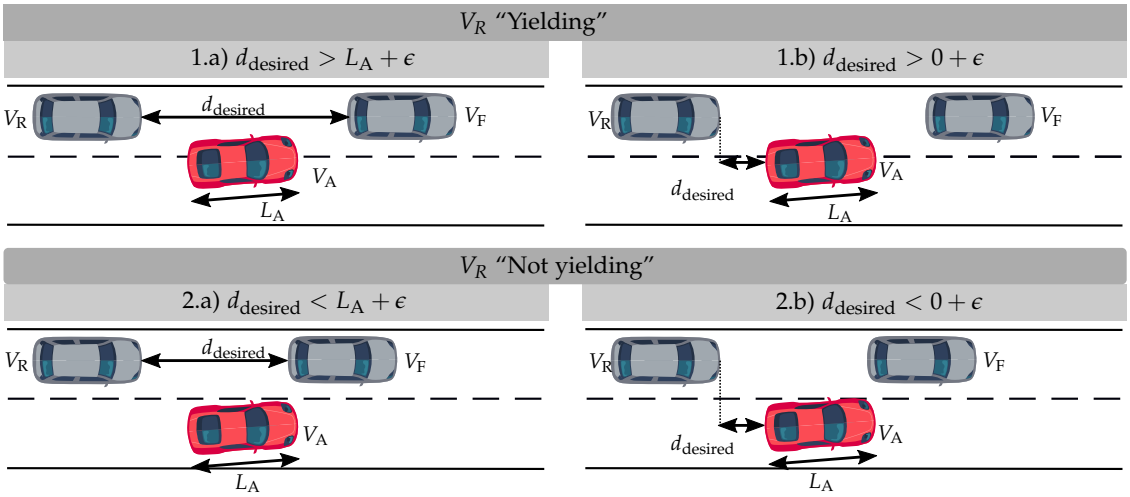


Figure 3.2.: Motivating example for behavior spaces. The yielding intent of rear vehicle V_R can be expressed independently of the *defined leading vehicle* V_F (1.a and 2.a) or V_A (1.b and 2.b) in a microscopic driver model. Relevant is how the desired safe distance d_{desired} being a model parameter relates to the minimum required merging gap L_A and a safety margin ϵ. Behavior spaces make use of this consideration to specify intention-independent prediction models.

of the scene for a small time duration. Such cues are hidden parameters of microscopic driver models specifying continuous behavior variations in human driving. The desired time headway T_{desired} of a rear vehicle V_R to a leading vehicle V_L is such a parameter in the car-following model IDM [186] (cf. App. A.1). It specifies a velocity-dependent safe distance $d_{\text{desired}} = T_{\text{desired}} \cdot vel_R^t$ under stationary conditions influenced by the current velocity vel_R^t of the rear vehicle. A common approach in interactive prediction is to model different intents by assuming a different leading vehicle V_L during model evaluation [55], $V_L \widehat{=} V_F$ or $V_L \widehat{=} V_A$. The following cases exist influencing the lane changing option of the AV in different ways:

1. **Giving way**:
   a) **Front vehicle leading ($V_L \widehat{=} V_F$):** If the desired time headway is such that d_{desired} to the front vehicle is by a safety margin ϵ larger than the *length of the AV* L_A, a lane change is viable.
   b) **Autonomous vehicle leading ($V_L \widehat{=} V_A$):** If the desired time headway is such that d_{desired} to the AV is larger than a safety margin ϵ, a lane change is viable.
2. **Taking way:**
   a) **Front vehicle leading ($V_L \widehat{=} V_F$):** If the desired time headway is such that d_{desired} to the front vehicle is by a safety margin ϵ smaller than the *length of the AV* L_A, a lane change is not viable.
   b) **Autonomous vehicle leading ($V_L \widehat{=} V_A$):** If the desired time headway is such that d_{desired} to the AV is smaller than a safety margin ϵ, a lane change is not viable.

Fig. 3.3 visualizes the joint distribution $f(\text{acc}_{\text{IDM}}, T_{\text{desired}})$ over the output acceleration, acc_{IDM}, of the IDM model and the time headway T_{desired} for the two options of choosing a leading

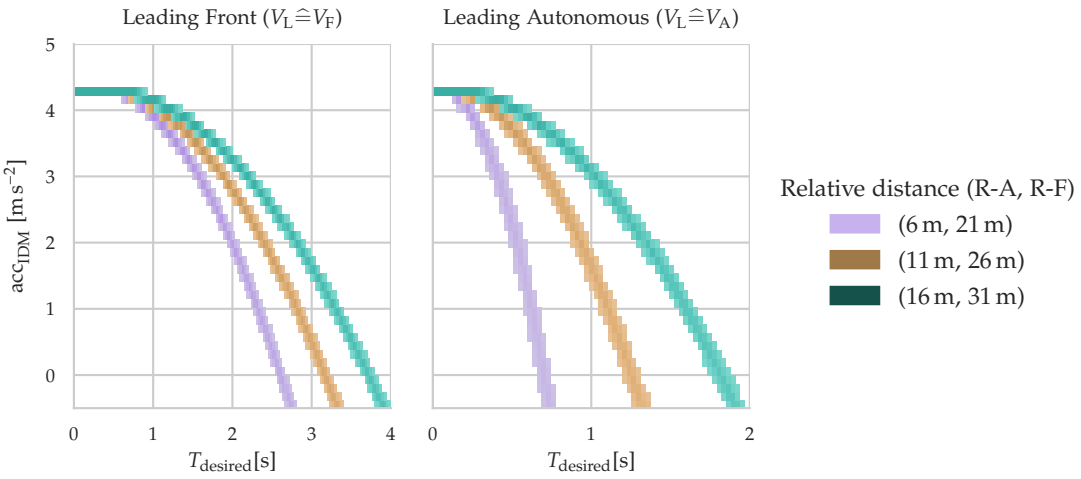


Figure 3.3.: Comparison of IDM outputs for the two intent parameterizations discussed in Fig. 3.2. A distribution $f(\mathrm{acc}_{\mathrm{IDM}}, T_{\mathrm{desired}})$ is obtained by adding small uniform noise to relative positions (R-A: rear-autonomous, R-F: rear-front) and velocities and is given for different front-rear distances between vehicles. An IDM parameterized with leading vehicle being the front vehicle on the same lane ($V_L \widehat{=} V_F$) can cover acceleration outputs of an IDM parameterized with leading vehicle being the AV ($V_L \widehat{=} V_A$) when varying the time headway T_{desired}. Details on the data generation are given in App. A.2.

vehicle. According to the definition of the SBG (cf. Sec. 3.2.1), the AV is able to observe the past actions and dynamic state of the rear vehicle and must infer the hidden information from these observations. Thus, the desired time headway and the chosen leading vehicle are not observable. However, the joint distributions indicate that it is sufficient to estimate T_{desired} based on an IDM parameterized to take way, i.e., applying $V_L \widehat{=} V_F$. When varying T_{desired}, the model also covers the outputted accelerations of the model parameterized to yield. This redundancy allows representing intents implicitly by a variation of the continuous model parameter. In addition, microscopic variations are explicitly predicted by the microscopic model.

The next section uses this finding to formalize a behavior space model for interactive prediction.

3.3.2. Behavior Space Model

The behavior space model assumes that a hypothetical policy and an accompanying behavior space definition exist for a specific driving scenario.

Definition 3.1 (Hypothetical Policy) *A hypothetical policy is a deterministic policy*

$$\pi^* : \mathcal{H}_o \times \mathcal{B}_j^t \to A_j \tag{3.4}$$

with $b_j^t \in \mathcal{B}_j^t$ being the jth agent's behavior state at time t and $\mathcal{B}_j^t \subset \mathbb{R}^{N_B}$ its behavior space of dimension N_B. The definition of the hypothetical policy and the behavior space are such that the behavior state b_j^t is a physically interpretable quantity.

Agent j acts within its behavior space $\mathcal{B}_j^t$ by selecting a behavior state b_j^t from $\mathcal{B}_j^t$ in each

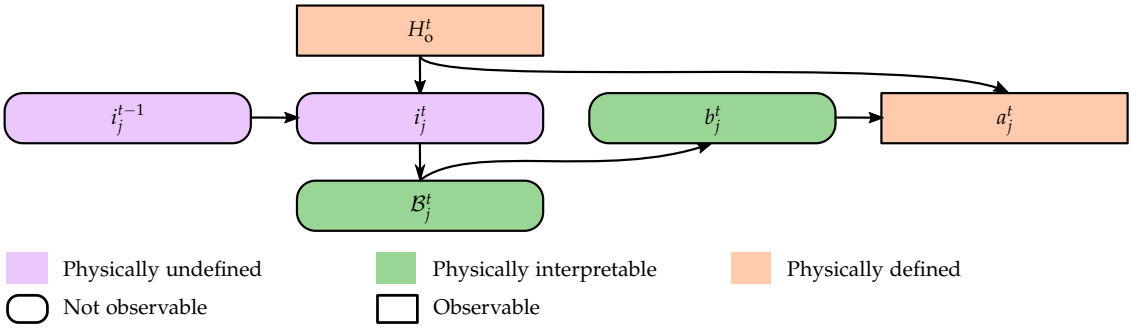


Figure 3.4.: Causal diagram to model the conditional dependence of intentions, behavior space and state, and actions for other agents j. Behavior spaces $\mathcal{B}_j^t$ are affected by intent states i_j^t and span a range of possible behavior states b_j^t upon which the other agent's policy depends.

time step before choosing an action according to π^*. The other agents' behavior spaces $\mathcal{B}_j^t$ and their current behavior state b_j^t are not observable. In this model, the intention of an agent j at time t, defined using intention states $i_j^t \in \mathcal{I}$, e.g., in the previous example $i_j^t =$ "yield" or $i_j^t =$ "take way", have a *causal effect* on the behavior space of an agent. Causal models define an interventional type of conditional distribution instead of the observational variant [191]. The observational distribution $P(\mathcal{B}_j^t | i_j^t)$ can only exist if the joint distribution $P(\mathcal{B}_j^t, i_j^t)$ exists. This is not the case since intention states i_j^t are physically undefined and therefore not measurable. The interventional distribution of behavior spaces $P(\mathcal{B}_j^t | do(i_j^t =$ "yield"$))$ is obtained in a controlled experiment when an agent acts under a defined intent.

The causal diagram in Fig. 3.4 illustrates the relations between the random variables in the behavior space model that accounts for

- inter-driver variations by integrating *agent-dependent behavior spaces* $\mathcal{B}_j^t$ affected by intents,
- intra-driver variations by *time-dependent behavior states* selected from agent-dependent and time-dependent behavior spaces $\mathcal{B}_j^t$.

However, experiments to obtain an interventional distribution require accurate models of intents which may be cumbersome to define and ambiguous or incomplete depending on the complexity of the traffic situation. It becomes clear that the definition of the actual behavior hypotheses set should not rely on the definition of intents. To avoid the definition of intention models, the behavior space model relies on the desirable property of *physically interpretability* of behavior states. An expert can define a full behavior space $\mathcal{B}$, comprising the individual behavior spaces $\mathcal{B}_j^t$ ($\mathcal{B}_j^t \subset \mathcal{B}$), by looking at the physically realistic situations. For instance, for the motivating example in Fig. 3.2, it is straightforward to define the physical boundaries of a behavior state modeling the desired time headway, $b \widehat{=} T_{\text{desired}}$, between agent j and i during lane changing with the one-dimensional behavior space $\mathcal{B} = \{b | b \in [0, d_{\max}/vel_{\min}]\}$ where $d_{\max}$ is the maximum sensor range and $vel_{\min}$ the minimum physically feasible velocity of the modeled front vehicle. The next section presents a design process to define hypothesis sets based on the full behavior space $\mathcal{B}$.

3.3.3. Hypothesis Design Process

The standard type-based method [154] defines each type θ^k such that it can closely match a *single* unknown policy π_j of another agent j. The following approach defines a collection of hypotheses, each covering a certain part of the continuous behavior space $\mathcal{B}$. Thus, *multiple* hypotheses equally participate in representing an unknown policy π_j.

Specifically, a uniform partition of the full behavior space $\mathcal{B} = \mathcal{B}^1 \cup \mathcal{B}^2 \cup \ldots \cup \mathcal{B}^K$, $\forall l \neq k : \mathcal{B}^l \cap \mathcal{B}^k = \emptyset$ is used forming K hypothesis $\pi_{\theta^k} : \mathcal{H}_o \times A^k \rightarrow [0,1]$, $k \in \{1,\ldots,K\}$*. The probability distribution over actions is defined in terms of the hypothetical policy π^* and the part of the behavior space $\mathcal{B}^k$ assigned for hypothesis k.

The behavior space model presented in the previous section defines that an agent selects a behavior state b_j^t in each time step from its behavior space $\mathcal{B}_j^t$. It is unknown how this selection is performed. The selection process could be expressed time-dependently using a stochastic process model, e.g., Gaussian processes. Such a model would require fitting or belief tracking of model parameters. However, a fast adaptation of such model parameters to observed intra-driver behavior variations seems unrealistic based on a few past observations. Instead, a more general approach is to assume that other agents uniformly sample a behavior state $b_j^t \sim \mathcal{U}(\mathcal{B}_j^t)$ in each time step. Given this assumption, a uniform probability density is defined over a part of the behavior space $\mathcal{B}^k$ as

$$f_k(b) = \begin{cases} \frac{1}{||\mathcal{B}^k||_V} & b \in \mathcal{B}^k \\ 0 & \text{else} \end{cases} \tag{3.5}$$

with $||\cdot||_V$ measuring the volume of a space.

Definition 3.2 (Behavior Hypothesis k) *Given a full behavior space $\mathcal{B}$ partitioned into K hypotheses behavior spaces $\mathcal{B}^k$, the corresponding densities f_k, and a hypothetical policy from Def. 3.1, the behavior hypothesis k is defined as*

$$\pi_{\theta^k}(a_j|H_o^t) = \Pr(\{b \,|\, \forall b \in \mathcal{B}^k,\, \pi^*(b, H_o^t) = a_j\}) \tag{3.6}$$

with $Pr(\cdot)$ denoting the probability of a set of behavior states under density f_k. The action space of the behavior hypothesis k becomes

$$A^k = \{a_j | \, \forall b \in \mathcal{B}^k,\, \pi^*(b, H_o^t) = a_j\}. \tag{3.7}$$

The action space A^k is *continuous*, since different behavior states typically imply different actions. Fig. 3.5 visualizes exemplarily the design of a hypothesis set according to the proposed method.

Classical driver models are often overdetermined with respect to the model parameters when observing the action only, e.g., the IDM [129]. Therefore, different parameter configurations yield the same outputted action. Given an abstract measure $|\cdot|$ of how many samples sufficiently represent a continuous space, the sample sizes of action and behavior space are approximately

*This thesis applies a uniform partitioning whereas the approach also supports other forms of partitioning, e.g., to obtain a higher belief resolution for certain parts of the behavior space

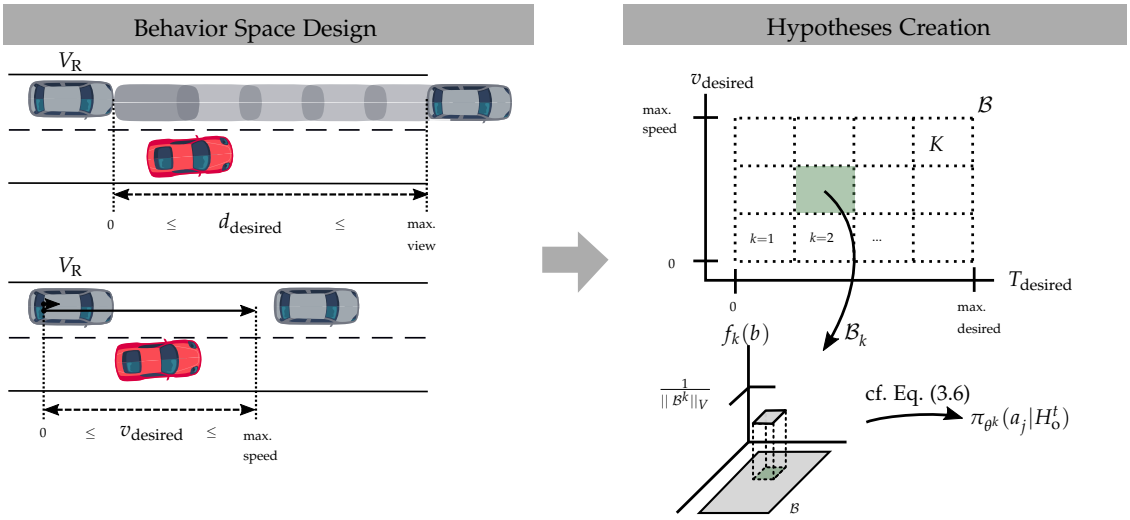


Figure 3.5.: Hypotheses design in behavior spaces. The two-dimensional full behavior space $\mathcal{B}$ covers realistic values of the desired time headway T_{desired} and desired velocity v_{desired} of an IDM model for vehicle V_R in a merging situation. The full behavior space $\mathcal{B}$ is partitioned arbitrarily into $K = 12$ parts. Each hypotheses $\pi_{\theta^k}(a_j|H_o^t)$ is defined over a single part $\mathcal{B}_k$ assuming uniform sampling of behavior states $b_j^t \sim f_k(b)$ *within* a hypothesis.

equal $|A^k| \approx |\mathcal{B}^k|$ for lower-dimensional behavior spaces, e.g., over a single parameter such as T_{desired}. For higher-dimensional behavior spaces, i.e., over multiple parameters of the IDM, the sample size of the action space is much smaller compared to the sample size of the behavior space, $|A^k| \ll |\mathcal{B}^k|$. Ideally, the behavior space is designed such that $|A^k| \approx |\mathcal{B}^k|$ to achieve optimal separation between the hypothesis.

The advantage of defining a hypothesis set over a range of parameters compared to using regression or belief tracking over a single parameter set as, e.g., in [128] is twofold:

- Both approaches must fix some driver model parameters due to over-determinism and estimate the remaining parameters. When estimating a single parameter set, the method is more sensitive to incorrect settings of other parameters. In contrast, employing beliefs over a parameter range better tolerates inaccuracies of the other model parameters.
- The parameter range of a single hypothesis accounts for intra-driver variations *within* the hypothesis behavior space $\mathcal{B}^k$. The next section 3.3.4 shows how *multiple* hypotheses express time dependence of behavior states by using a specific posterior update. In contrast, single parameter sets cannot express time dependence.

Given the definition of the hypothesis set, the following section describes how to track posterior beliefs over hypotheses.

3.3.4. Leveraging the Sum Posterior for Modeling Intra-Driver Variations

The posterior belief $\Pr(\theta^k|H_o^t, j)$ represents how each behavior hypothesis $\pi_{\theta^k}(a_j|H_o^t)$ contributes to the microscopic prediction of an agent j as defined in Eq. (3.1). Albrecht [179] defines the

posterior, related to classical Bayes filtering [59, 137], as a combination of the likelihood of the jth agent's actions in the history of observations, denoted as $L(H_o^t|\theta^k, j)$, and the prior probability of a hypothesis $P(\theta^k)$ as

$$\Pr(\theta^k|H_o^t, j) = \frac{L(H_o^t|\theta^k, j) \cdot P(\theta^k)}{\sum_{\widehat{\theta}^k \in \Theta} L(H_o^t|\widehat{\theta}^k, j) \cdot P(\widehat{\theta}^k)} \tag{3.8}$$

and proposes three approaches and useful applications for likelihood calculation: 1) Product posterior 2) Sum posterior and 3) Correlated Posterior. In [179], the likelihood calculation for the product posterior is defined as

$$L(H_o^t|\theta^k, j) = \prod_{a_j^{t'} \in H_o^t} \pi_{\theta^k}(a_j^{t'}|H_o^{t'}), \tag{3.9}$$

with $a_j^{t'} \in H_o^t$ denoting the actions of agent j in observation history H_o^t. For the sum posterior, it is formulated as

$$L(H_o^t|\theta^k, j) = \sum_{a_j^{t'} \in H_o^t} g(t - t')\pi_{\theta^k}(a_j^{t'}|H_o^{t'}) \tag{3.10}$$

with $g(\cdot)$ being a time-dependent weighting factor to reduce the influence of more past action probabilities onto the posterior estimate.

According to [179], product posteriors converge to a pure type distribution which is helpful in the case of agents having a fixed type, i.e., a type that does not change over time. Sum posteriors converge under certain conditions to a mixed (and pure) type distribution, modeling that agent types change over time. The correlated posterior extends the sum posterior to express the correlation of types between agents. In the behavior space and hypotheses model presented in the previous section, an agent's actual type is not represented by a single hypothetical type but a combination of types, i.e., behavior hypotheses. The product posterior is, therefore, an invalid choice since it is zeroed under changing types. Further, the model assumes that an agent's behavior space $\mathcal{B}_j^t$ is independent of other agents' behavior. Dependence exists only via the action-observation history H_o^t. It is thus not required to model correlations in the posterior calculation.

The sum posterior in combination with hypotheses designed according to Sec. 3.3.3 is most suitable to account for intra-driver variations which is discussed in the following. Fig. 3.6 visualizes three special cases how the unknown behavior space of an agent $\mathcal{B}_j^t$ can be positioned relative to the hypothesis behavior spaces for a time span $\mathrm{T} = \{t_1, \ldots, t_2, \ldots, t_3\}$:

1. $\mathcal{B}_j^t \subseteq \mathcal{B}_k$, $\forall t \in \mathrm{T}$: The unknown behavior space $\mathcal{B}_j^t$ is a subset of the behavior space $\mathcal{B}_k$ of a single hypothesis $\pi_{\theta^k}(a_j^t|\cdot)$ for all times. In this case, time-dependence is modeled using uniform sampling of behavior states, as discussed in the previous section. The posterior calculation does not influence the modeling of time dependence in this case. The posterior belief can be correct after observing a single action at time $t = t'$.

2. $\mathcal{B}_j^t \subseteq (\mathcal{B}_k \cup \mathcal{B}_{k'} \cup \ldots)$, $\forall t \in \mathrm{T}$: The unknown behavior space $\mathcal{B}_j^t$ is a subset of the union

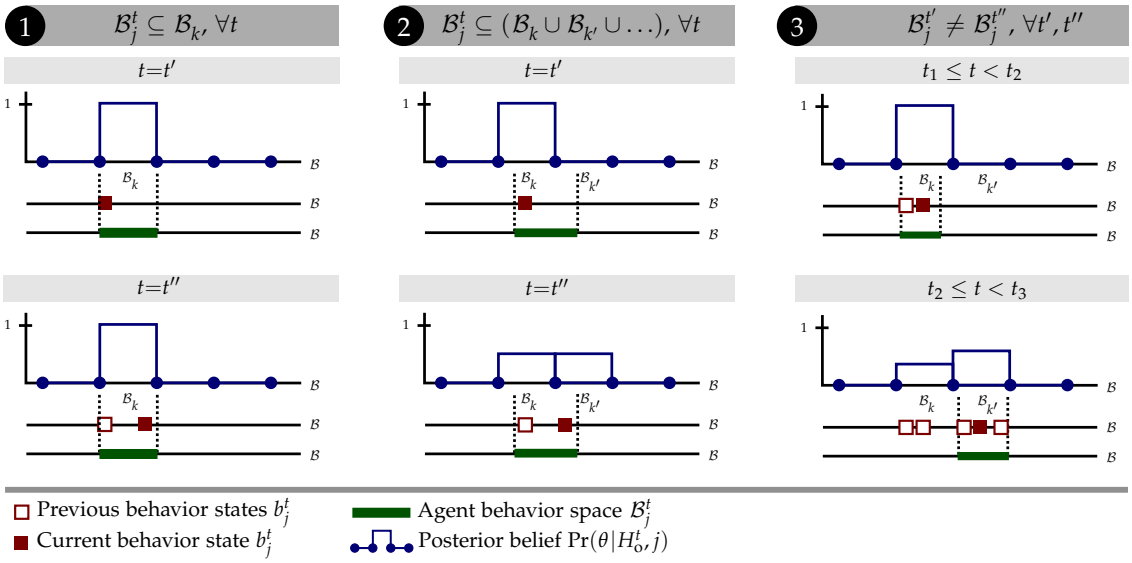


Figure 3.6.: Capturing intra-driver variations using sum posteriors. (1) The posterior belief over hypotheses $\Pr(\theta^k|H_o^{t'}, j)$ can be correct after one time step if the unknown behavior space of agent $\mathcal{B}_j^t$ is a subset of kth hypothesis behavior space $\mathcal{B}_k$. (2) If $\mathcal{B}_j^t$ covers multiple hypothesis behavior spaces $\mathcal{B}_k$ and $\mathcal{B}_{k'}$ the sum posterior achieves an *Or* combination of probabilities. (3) When $\mathcal{B}_j^t$ changes over time summing of probabilities achieves only a gradual shift. The examples assume optimal separation of hypotheses (cf. Sec. 3.3.3).

of two or more parts of the behavior space $\mathcal{B}_k, \mathcal{B}_{k'}, \ldots$ for all times. The following consideration assumes that at time t' an agent j samples a behavior state $b_j^{t'}$ from $\mathcal{B}_j^t$, being also in $\mathcal{B}_k$, and at time $t'' > t'$ samples a behavior state $b_j^{t''}$ from $\mathcal{B}_j^t$, being also in $\mathcal{B}_{k'}$. This yields under the condition of near optimal separation of hypotheses discussed in the previous section for the probability of observing the agent's action under these time steps, $\pi_{\theta^k}(a_j^{t'}|\cdot) \gg \pi_{\theta^{k'}}(a_j^{t'}|\cdot)$ and $\pi_{\theta^k}(a_j^{t''}|\cdot) \ll \pi_{\theta^{k'}}(a_j^{t''}|\cdot)$. Summing the probabilities of each type yields a balanced posterior with $\Pr(\theta^k|H_o^{t'}, j) \approx \Pr(\theta^{k'}|H_o^{t'}, j)$ at time t''. The summing operation can be interpreted as an *Or* combination of probabilities. Using this interpretation, one observes that a belief for a specific type θ^k reflects how likely it is that the unknown agent behavior space falls partly into the hypothesis behavior space ($\mathcal{B}_j^t \cap \mathcal{B}_k$). Multiple beliefs in a sense cover the unknown behavior space of another agent.

3. $\mathcal{B}_j^t \subseteq \mathcal{B}_k, \forall t : t_1 \leq t < t_2$ and $\mathcal{B}_j^t \subseteq \mathcal{B}_{k'}, \forall t : t_2 \leq t < t_3$: The unknown behavior space $\mathcal{B}_j^t$ changes after some time, being first a subset of hypothesis behavior space $\mathcal{B}_k$ and then from $t \geq t_2$ a subset of hypothesis behavior space $\mathcal{B}_{k'}$. Such changes can, e.g., arise due to a changing intent of agent j and occur over a longer time horizon. Before the change, action probabilities are higher for hypothesis k, $\pi_{\theta^k}(a_j^t|\cdot) \gg \pi_{\theta^{k'}}(a_j^t|\cdot) \forall t : t_1 \leq t < t_2$ and after the change higher for hypothesis k', $\pi_{\theta^k}(a_j^t|\cdot) \ll \pi_{\theta^{k'}}(a_j^t|\cdot), \forall t : t_2 \leq t < t_3$. Ideally, the belief $\Pr(\theta^{k'}|H_o^{t'}, j) = 1$ immediately at time $t = t_2$ as it would be the case with product-based posteriors zeroing out the other posterior belief. The summing operation achieves only a gradual shift of the beliefs. It requires the history to contain more actions observed after the change of $\mathcal{B}_j^t$ than before to zero out $\Pr(\theta^k|\cdot, j)$. This time lag can be adjusted by reducing

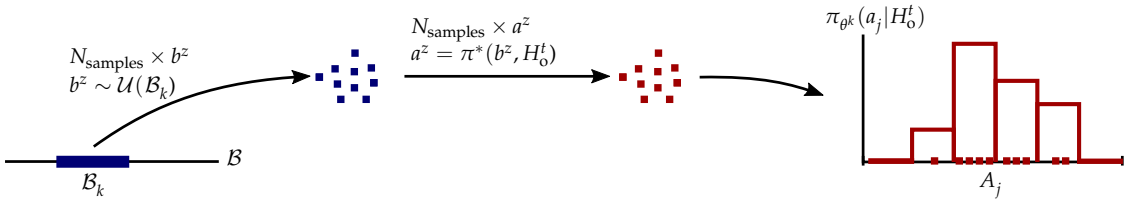


Figure 3.7.: Histogram approximation of probability density $\pi_{\theta^k}(a_j|H_o^t)$ of kth behavior hypotheses. For N_{samples} behavior states b^z, which are sampled uniformly from the hypothesis behavior space $\mathcal{B}_k$, the action a^z is calculated based on the hypothetical policy $a^z = \pi^*(b^z, H_o^t)$. The probability density over the action space A_j is approximated using a histogram.

the influence of past observed actions onto the belief using the time-dependent weighting factor $g(\cdot)$. Therefore, the likelihood calculation considers only the last L_H observed actions at time t with

$$g(t') = \begin{cases} 1.0 & t' > t - L_H \\ 0.0 & t' \leq t - L_H \end{cases} \tag{3.11}$$

Overall, there is a trade-off between getting coverage of the current unknown behavior space of agent $\mathcal{B}_j^t$ and fast adaptation to a change of this space.

These exceptional cases are more theoretical to understand the sum posterior's capabilities to represent intra-driver variations. In actual applications of belief tracking, arbitrary mixtures of the presented cases occur.

3.3.5. Sampling-Based Action Density Approximation

To actually calculate the sum posterior during interaction with other agents, the probability density $\pi_{\theta^k}(a_j|H_o^t)$ must be given according to Eq. (3.6). It can be seen as the probability density of a function of uniform random variables. The input to the hypothetical policy is the uniform density over the hypothesis behavior space f_k. In the case of classical driver models, e.g., the IDM, the hypothetical policy π^*, is a non-linear, non-reversible mapping between behavior state and action, $\pi^*(b, H_o^t) = a_j^t$. Analytical calculations of the density become cumbersome or infeasible for a higher dimensional behavior space, i.e., comprising multiple model parameters.

Instead, a histogram is used to approximate the probability density $\pi_{\theta^k}(a_j|H_o^t)$. The action space A_j is decomposed into bins of equal width Δw_b. Then, the hypothetical policy is evaluated for behavior states b^z, $z \in \{1, \ldots, N_{\text{samples}}\}$ sampled from the uniform density $b^z \sim f_k$ (cf. Sec. 3.3.3) which gives a collection of actions a^z. The ratios of actions within the bins over the total number of actions define a histogram (cf. Fig. 3.7). Though, a density $\pi_{\theta^k}(a_j^t|H_o^t)$ can then be obtained by dividing the ratios through the width of the bins, this step is not necessary due to the normalization of posteriors given in Eq. (3.8). The limits of the action space A_j are determined by the vehicles' feasible actions, e.g., given by longitudinal acceleration limits when using the IDM as hypothetical policy.

Such a density estimation must be performed at each time step for each agent and hypotheses based on the current history H_o^t. These computations could be accelerated by parallelizing the sampling process and the individual posterior updates in an actual application.

3.4. Robust Stochastic Bayesian Game

This section first analyzes the sample complexity of planning with the Harsanyi Bellman Ad-Hoc (HBA) algorithm (cf. Sec. 3.2.3) in continuous behavior spaces. It then motivates and integrates robustness-based optimality into the Stochastic Bayesian Game (SBG) giving the Robust Stochastic Bayesian Game (RSBG), and shows that the integration yields reduced sample complexity when planning under continuous behavior variations of other drivers.

3.4.1. Sample Complexity of the Stochastic Bayesian Game in Behavior Spaces

With the definition of the behavior hypotheses set and the tracking of posterior beliefs, the AV is able to interactively predict continuous behavior variations. However, approximating a solution to the HBA algorithm (cf. Sec. 3.2.3) with Simultaneous-Move MCTS (SM-MCTS) is computationally demanding for a continuous space of joint actions. To get further insight into the problem, the sample complexity of Eq. (3.2) and Eq. (3.3) can be calculated for the proposed hypothesis definition. Given that the full behavior space $\mathcal{B}$ is decomposed into K equal parts $\mathcal{B}_k$ to define the hypotheses set and assuming a near optimal separation of hypotheses (cf. Sec. 3.3.3), the sample size of the action space of hypothesis k is

$$|A^k| \approx |\,\mathcal{B}|/K. \tag{3.12}$$

For a specific combination of types θ_{-i}, the joint action space of other agents is defined as cartesian product of the hypotheses action spaces, $A_{-i} = \bigtimes_{(\theta^k,j)\in\theta_{-i}} A^k$, and its sample size is

$$|A_{-i}| = \prod_{j=1}^{N_{-i}} |A^k| \approx (|\,\mathcal{B}|/K)^{N_{-i}} \tag{3.13}$$

with $N_{-i}{=}N{-}1$ being the number of other agents. Solving the SBG with SM-MCTS applies root sampling of the current combination of hypotheses θ_{-i} from the posterior beliefs before the start of a search iteration [187] to then use it for forward prediction during tree search. The number of possible sampling options is $|\Theta_{-i}|{=}K^{N_{-i}}$. At a single search iteration, joint actions are sampled from the joint action space A_{-i} at selection, expansion and rollout steps. The sample complexity thus is $|A_{-i}|^{T_\mathrm{p}}$ to achieve a balanced search tree with depth T_p for a single combination of types. Including root sampling, the asymptotic worst-case sample complexity of the HBA algorithm solved with SM-MCTS in behavior spaces becomes

$$\mathcal{O}(|\Theta_{-i}| \cdot |A_{-i}|^{T_\mathrm{p}}) \overset{\text{cf. A.3}}{=} \mathcal{O}_{\mathrm{SBG}}(|\,\mathcal{B}|^{N_{-i}T_\mathrm{p}} K^{N_{-i}-N_{-i}T_\mathrm{p}}). \tag{3.14}$$

The ego action space A_i is seen as a constant and is therefore dropped from the analysis. Increasing K reduces the sample complexity since $N_{-i} - N_{-i}T_\mathrm{p}$ is always negative given that $T_\mathrm{p} > 1$. Nevertheless, it *exponentially depends* on the prediction depth T_p and the number of other agents N_{-i} over the sample size of the behavior space $|\,\mathcal{B}|$ which reduces the applicability of the

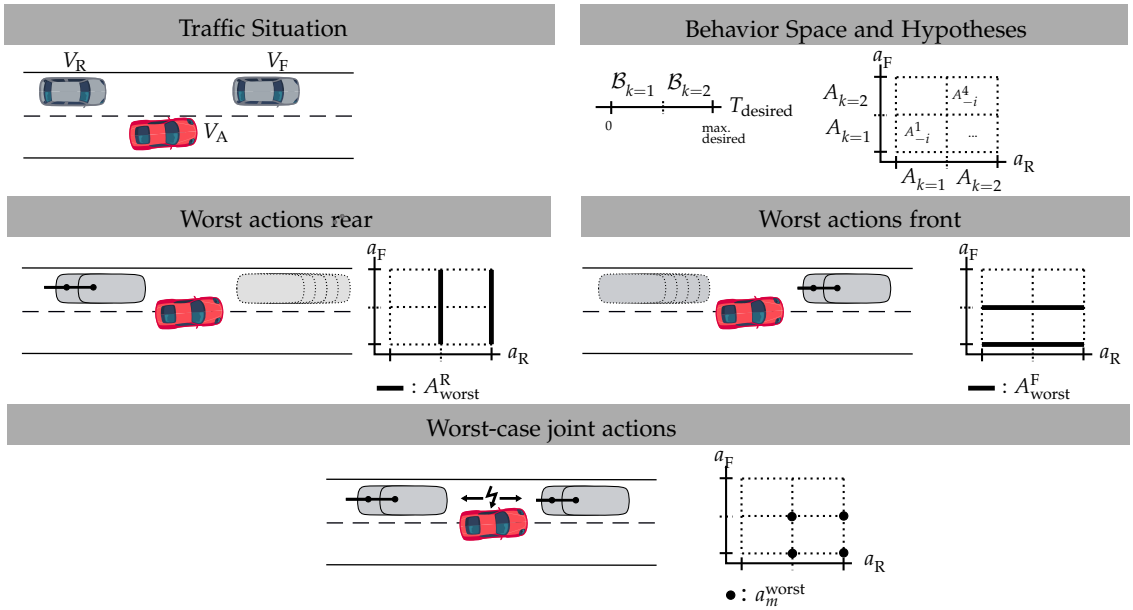


Figure 3.8.: Motivation for robustness-based optimality in interactive traffic. In a lane changing scenario, a one-dimensional behavior space over T_{desired} is partitioned into two hypotheses. Sample complexity is reduced by using a decoupled selection of the worst-case action from the agent-specific worst-case action spaces $A^{\text{R}}_{\text{worst}}$ and $A^{\text{F}}_{\text{worst}}$ to form the joint worst-case action a^{worst}_m.

SBG in interactive planning for AVs.

The following section motivates a more-sample efficient optimality criterion for interactive planning in behavior spaces.

3.4.2. Motivation for Combining Robustness with Agent Behavior Hypothesis

This section motivates the integration of robustness-based optimality into the SBG. Fig. 3.8 depicts a lane-changing scenario for which a hypotheses set is defined based on the methodology of Sec. 3.3. The full behavior space over the desired time headway T_{desired} is partitioned into two hypotheses. Each joint action space, A^m_{-i}, $m \in \{1, \dots, 4\}$ for one of the four combination of types ($|\Theta_{-i}| = 4$) contains an *infinite* number of possible joint actions which must be considered in the tree search.

The key idea to reduce the sample complexity of the SBG is to prioritize evaluation of joint actions fulfilling specific criteria. In the context of autonomous driving, it is reasonable to prioritize joint actions which sacrifice safety, i.e., which lead to worst-case outcomes for the AV. In the example, such a situation occurs when during lane changing of the AV (V_A), the rear vehicle (V_R) desires a small distance to the AV. In contrast, the front vehicle (V_F) aims for a large gap to its leading vehicle. In that case, the gap available for a merge of the AV decreases significantly, and an unsafe situation, e.g., a collision, may become likely. Sampling such worst-case joint actions with priority can reduce sample complexity since it does not require evaluating all possible joint actions.

However, the following argumentation shows that considering worst-case outcomes over the full behavior space $\mathcal{B}$ is similar to predicting other participants using reachable sets. The behavior states $b \in \mathcal{B}$ span a set of physically reachable states given that each behavior state b maps to a physical action a according to the hypothetical policy $a = \pi^*(b, H_o^t)$, and the full behavior space is designed to comprise all physically realistic behaviors. Planning which only considers the worst-case prediction within the set of physically reachable states is similar to planning under reachable sets. As motivated in Sec. 1.1.1, set-based prediction yields conservative driving in dense traffic, potentially leading to the freezing vehicle symptom. Also, such an approach completely neglects the information about the behavior of other drivers available from the posterior beliefs over hypotheses.

One observes in the example that there exists for each combination of hypotheses a single, worst-case joint action a_m^{worst} concerning the ego agent. Further, this joint action consists of subjective worst-case actions of the other agents from the action spaces $A_{\text{worst}}^{\text{R}}$ and $A_{\text{worst}}^{\text{F}}$. The sample complexity is reduced by letting each agent select a subjective worst-case action within its hypothesis action space. This decoupled action selection provides a meaningful approximation of the global worst-case joint action in dense traffic scenarios. Integration of belief information over hypothesis is achieved with this concept since, due to root sampling of combinations of types, worst-case joint actions are selected proportionally to the beliefs.

Considering the worst-case outcome over a parameter space in decision making is also referred to as robustness-based optimality. A review of decision-theoretic models applying this concept is given in the next section. Sec. 3.4.4 then reduces the sample complexity of the SBG by formalizing combined robustness- and game-based planning as Robust Stochastic Bayesian Game (RSBG).

3.4.3. Review of Robustness-Based Optimality

The robustness of a plan or policy to *continuous* modeling errors has long been studied in the control and reinforcement learning community [192–195]. The Robust Markov Decision Process (RMDP) framework searches for a solution which is optimal under the worst-case parameter realizations of a (possibly continuous [196]) set of parameters of the transition function, denoted uncertainty set. The main challenge with the robustness criterion is finding an uncertainty set which avoids overly conservative policies [197, 198].

Combinations of robust optimization and Bayesian decision making have been investigated in reinforcement learning [197] and game theory [199]. The latter approach denoted Robust game theory, applies the worst-case operation *over the type space* to omit dependency on posterior type-beliefs in the expected value calculation. In contrast to their work, the RSBG applies the worst-case operation *over the parameter space* of each individual type k. The parameter space corresponds with the above hypothesis design process to the behavior space part $\mathcal{B}_k$ for hypothesis k. The posterior beliefs are not omitted, but are used to weight the worst-case returns each obtained under a different combination of types.

A Robust Markov Decision Process (RMDP) models uncertainty about the parameters of the transition function p in an MDP [193]. It can be viewed as a two-agent stochastic game where an adversary attempts to minimize the expected return of the controlled agent i by selecting

the transition function p inducing the worst-case outcome. The robust Bellman equation [196] is defined as

$$Q_R(a_i, o) = r(o, a_i) + \gamma \max_{a'_i} \inf_{p \in \mathcal{P}} \mathbb{E}^p[Q_R(a'_i, o')].$$

In the multi-agent case, the worst-case assumption is applied over the other agents' joint action giving the robust Bellman equation

$$Q_R(a, o) = r(o, a) + \gamma \max_{a_i \in A_i} \min_{a_{-i} \in A_{-i}} Q_R(a_{i,-i}, o') \tag{3.15}$$

with minimax learning objective [194]. Its formulation of robustness is related to the RSBG presented in the following section.

3.4.4. Model Definition and Sample Complexity Reduction

Sec. 3.4.2 suggests to apply decoupled worst-case action selection for the SBG to reduce sample complexity. The conservativeness of a pure robustness-based optimality criterion can be avoided by combining the optimality criteria of the robust Bellman equation (cf. Eq. (3.15)) and the SBG (cf. Eq. (3.2) and Eq. (3.3)). The resulting RSBG lets other agents act adversarially *only within* their hypothesis by defining the worst-case operation over the hypotheses action spaces A^k.

Definition 3.3 (Robust Stochastic Bayesian Game (RSBG)) *Given an SBG with state, action and agent definitions according to Sec. 3.2.3, the RSBG uses a different optimality criterion with the expected cumulative reward for the ego agent i being defined as* $E(H_o, a_i) =$

$$\begin{aligned} &\sum_{\theta_{-i} \in \Theta_{-i}} \Pr(\theta_{-i}|H_o) \left[\min_{a_{-i} \in A_{-i}} Q_R(H_o, a_{i,-i}) \right] \overset{\substack{\text{decoupled} \\ \text{action} \\ \text{selection}}}{=} \\ &\sum_{\theta_{-i} \in \Theta_{-i}} \Pr(\theta_{-i}|H_o) \left[Q_R(H_o, a_{i,-i}) \,:\, a_{-i} = (a_j, \ldots)_{\forall j},\ a_j = \underset{\substack{a'_j \in A_k \\ a'_{-j} \sim A_{-j}}}{\operatorname{argmin}} Q_R(H_o, (a_i, a'_j, a'_{-j})) \right]. \end{aligned} \tag{3.16}$$

whereas the definition of the Bellman equation remains unchanged as in Eq. (3.3)*.*

The worst-case selection over the joint hypothesis action space A_{-i} is simplified based on the motivation of Sec. 3.4.2. Each other agent j selects individually a worst-case action a_j with respect to the ego agent's return Q_R out of the agent's hypotheses action space A_k given by the type combination θ_{-i}. Agent j is not informed by the actions of other agents and chooses its worst-case action in a decoupled manner. The joint action in the minimizing operation thus includes for the respective other agents $-j$ random actions a'_{-j} sampled from their joint action space A_{-j} given by the type combination θ_{-i}, and consists of the ego agent's, agent j's and other agents' $-j$ actions. Concatenation of the individual worst-case actions of each other agent defines the joint worst-case action a_{-i} used in the evaluation of $Q_R(H_o, a_{i,-i})$.

Given the optimality definition of the RSBG, its sample complexity is derived similarly as for the SBG in Sec. 3.4.1. The minimum operation has a sample complexity equal to the sampling

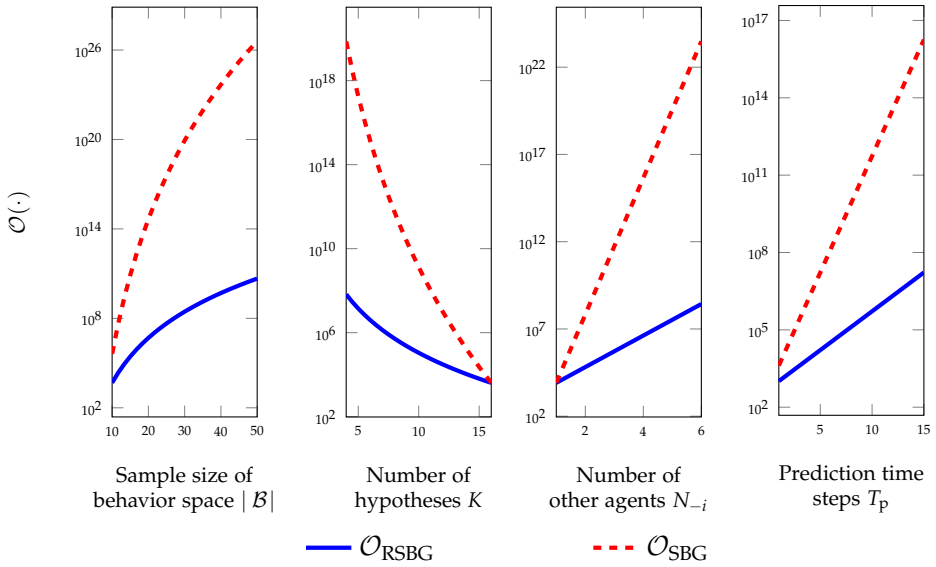


Figure 3.9.: Comparison of asymptotic worst-case sample complexities of $\mathcal{O}_\mathrm{RSBG}$ and $\mathcal{O}_\mathrm{SBG}$. To compare over a specific factor, other parameters are kept constant with $K = 8$, $|\mathcal{B}| = 16$, $N_{-i} = 3$ and $T_\mathrm{p} = 10$. In all cases, the RSBG shows reduced sample complexity. When $K = |\mathcal{B}|$, the sampling complexities become equal. The complexity calculations assume a balanced search tree. In practice, meaningful tree exploration reduces the actual required samples during planning for the SBG and RSBG.

size of the hypothesis action space $|A_k|$ since each other agent independently determines its worst-case action. The asymptotic worst-case sample complexity of the RSBG therefore is

$$\mathcal{O}(|\Theta_{-i}| \cdot |A_k|^{T_\mathrm{p}}) \overset{\text{cf. A.4}}{=} \mathcal{O}_\mathrm{RSBG}(|\mathcal{B}|^{T_\mathrm{p}} K^{N_{-i}-T_\mathrm{p}}). \tag{3.17}$$

The dependency of the sample complexity on the sample size of the behavior space is thus *reduced by a factor N in the exponent* compared to $\mathcal{O}_\mathrm{SBG}$. This reduction of sample complexity is important to enable interactive planning over continuous microscopic behavior variations. Fig. 3.9 gives details how the sample complexities of RSBG and SBG behave for different influencing factors.

The RSBG not only reduces the sample complexity, but it also changes the optimality of plans becoming in principle more conservative than with the SBG. However, in the context of autonomous driving, this is meaningful. On the one hand, the conservativeness of the policy can be controlled via the number of defined types. Further, prioritizing the sampling of unsafe situations is also favorable for risk-constrained planning. The next section presents a variant of SM-MCTS to find approximately optimal solutions to the RSBG.

3.5. Planning for the Robust Stochastic Bayesian Game

This section first reviews planning under uncertainty with Monte Carlo methods. It then presents a variant of SM-MCTS to interactively plan with the RSBG.

3.5.1. Review of Monte Carlo Planning under Environment Uncertainties

Monte Carlo Tree Search (MCTS) is a well-known tree search algorithm interleaving node selection, expansion and backpropagation steps in each search iteration [159]. Due to its anytime property, it is especially relevant in applications of online planning. Different variants of MCTS are widely applied to plan under uncertainties in game- and decision-theoretic models.

To deal with imperfect information of state in stochastic games, there exist determinization and information set approaches [159]. Determinization samples multiple perfect information games with fully observable initial game states from the stochastic game. It averages the results from non-probabilistic planning applied over each deterministic game [200]. Information sets include multiple determinized states with the same information value to select a move in the game [201]. Both approaches do not take into account belief information.

For single-agent models, e.g., Partially Observable Markov Decision Processes (POMDPs), different online planning approaches exist to consider partial observability of environment states. Point-based value iteration [202] approximates the belief value function for a finite set of belief points relying on the convexity property of the belief value function. It requires explicit modeling of the probability distributions and becomes infeasible in large state spaces [203]. Monte Carlo approaches simulate environment state transitions using a generative model enabling implicit definition of probability distributions. Partially Observable Monte Carlo Planning (POMCP) selects actions using Upper Confidence bound applied to Trees (UCT) in an MCTS algorithm to plan for POMDPs. Starting at the root observation state, being sampled from the current posterior belief, it predicts action-observation histories with a generative model. The resulting search tree represents future beliefs based on history nodes [203]. The Adaptive Belief Tree (ABT) planner improves online planning by reusing the search tree even when the generative and observed model deviate [204]. The Determinized Sparse Partially Observable Tree (DESPOT) [205] diminishes the curse of dimensionality of POMDP planning by holding a reduced set of action-observation histories characterized by fixed action sequences. Presented algorithms do not directly transfer to continuous state and action spaces. When using progressive widening [206] to balance exploration and exploitation, these algorithms find the policy of a simplified POMDP, the QMDP model which assumes full observability of states after one prediction step [137]. To avoid this behavior, the authors in [151] search over belief states instead of action-observation histories when using progressive widening. Bayesian optimization is used in [156, 207] to implement continuous actions in an MCTS planner for POMDPs. Continuous state spaces are represented by higher level features to enable use of value iteration POMDP solvers in [208, 209].

Simultaneous movements of agents form another source of uncertainty. These can be accounted for by using independent search trees which are not informed by the moves of other players [201, 210]. In general, mixed strategies are optimal in simultaneous movement settings. UCT selects actions deterministically based on action-values and counts. In SM-MCTS a decoupled variant, Decoupled Upper Confidence Bound (DUCB), is applied which selects actions independently for each player. However, DUCB yields a pure strategy, whereas action selection using Exponential-Weight Algorithm for Exploration and Exploitation (EXP3) [211] converges

to a Nash equilibrium in simultaneous move settings [212] as formally analyzed in [160]. Counterfactual Regret Minimization (CFR) is a planning technique for imperfect information games which chooses an action proportionally to the regret of not selecting the action in a previous visit of the state [213]. CFR has the drawback of having to evaluate the whole game tree in each search iteration. To reduce memory and computational demands, Monte Carlo variants of CFR consider only portions of the game tree in each iteration [214], employ abstracted games [215] or make use of pre-trained neural networks [216].

Planning given beliefs over agent behavior types or environment transition functions similarly applies variants of MCTS [187, 217]. The Bayes-Adaptive MDP (BAMDP) models beliefs over unknown parameters of the environment transition function in a single agent model. The SBG models beliefs over unknown types of other agents. The Bayes-Adaptive Monte Carlo Planning (BAMCP) algorithm proposed by [217] transfers the POMCP algorithm to BAMDPs. Bayes-Adaptive planning with Function Approximation (BAFA) extends BAMCP to continuous state spaces by online learning of a Q-function parameterized on the state-action history [218]. A formulation of the BAMDP as discrete POMDP is used in [127] to enable offline planning over a discrete set of uncertain parameters sampled from the prior distribution. In contrast, BAFA avoids belief discretization and achieves online performance by using interpolation between online learned belief Q-functions [218]. A drawback of BAFA is that it generalizes between beliefs with function approximation. Ad-hoc coordination, e.g., modeled as SBG, is related to planning for BAMDPs. However, ad-hoc coordination additionally considers simultaneous movements of multiple agents. Research in this domain focuses on games with a discrete action and state space and planning using value iteration [154] or MCTS [187, 219, 220]. In continuous state spaces, as discussed for POMDP and BAMDP planning, feature encoding and function approximation are similarly applied [174].

The following sections present a SM-MCTS planner for the RSBG in continuous state and action spaces as well as large belief spaces over hypotheses sets and relate it to presented work in planning under uncertainty.

3.5.2. Overview

Planning for the RSBG is based on SM-MCTS. The planning algorithm receives the last observed state o^t and the beliefs over the hypotheses for each agent $\Pr(\theta^k | H_o^t, j)$, $\forall j$. It iteratively employs selection, expansion, rollout and backpropagation in each search iteration until exceeding a maximum number of iterations $\mathsf{N}_{\text{iters}}$ or search time $\mathsf{T}_{\text{search}}$. Fig. 3.10 gives an overview of these steps with algorithmic details given in Alg. 1 and Alg. 2. Actions are chosen independently in stages similar to [52, 53] in selection, expansion and rollout steps. With the resulting joint action, the next observation state o' is predicted from the node's current observation o with the environment transition function (cf. Sec. 3.2.1). The prediction time span τ_{predict} increases linearly with the search depth d with minimum prediction time span τ_a. In contrast to previous work [52, 53] applying SM-MCTS to interactive planning for AVs, separate selection mechanisms are used for ego and other agents in EgoActionSelection and OtherActionSelection. Each selection strategy returns actions a_i or a_j at stage nodes $\langle H_o \rangle$. Stage nodes are uniquely

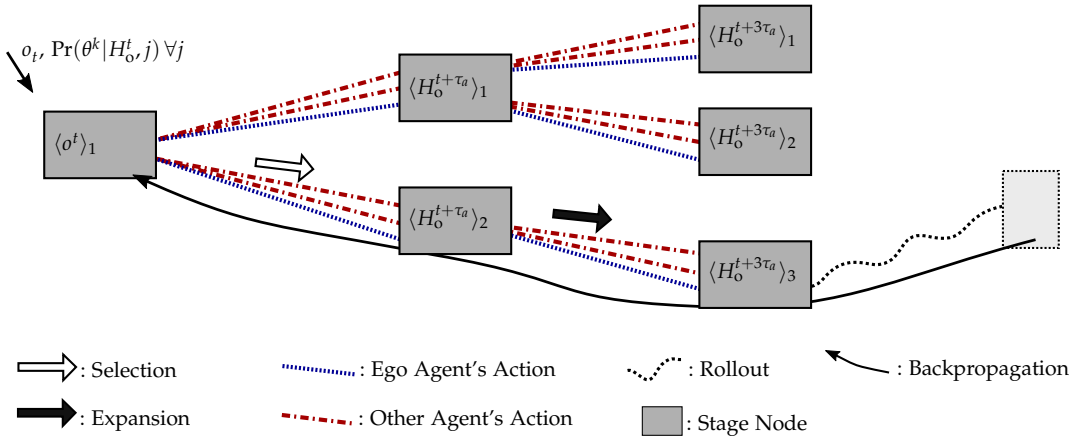


Figure 3.10.: Main planning steps of the RSBG planner using a variant of SM-MCTS. The search method receives the current observed environment state o^t and posterior beliefs for each agent and hypothesis $\Pr(\theta^k|H_o^t, j)\ \forall j$. In a stage-wise manner, ego and other agents choose individual actions with the joint action defining the transition to the next stage nodes $\langle\cdot\rangle$ in selection, expansion and rollout steps. Returns are backpropagated and the next search iteration starts. The prediction time span increases with search depth d according to $\tau_{\text{predict}} = d \cdot \tau_a$.

indexed by the sequence of observations and joint actions leading to the node. Concatenation of the history is expressed as $\langle H'_o \rangle = \langle H_o, (a_i, a_j), o' \rangle$. The selection algorithms are detailed in Sec. 3.5.4 and 3.5.5.

The complexity of online belief state planning in continuous state and action spaces demands low dimensional belief spaces, e.g., given with intent-based beliefs in autonomous driving [55]. In contrast, the proposed definition of the RSBG inherently comes with a vast belief space of size $N \cdot K$ comprising beliefs for each combination of agent and type. Intent beliefs should be distinct, e.g., to encode clearly that other participants yield or take way. Beliefs over hypotheses in behavior spaces *can be ambiguous* since, per definition of these, multiple beliefs represent the actual microscopic behavior of another driver. Given this particular structure and meaning of the belief space, the AV must not show information gathering behavior. It is therefore meaningful to resort to QMDP planning with the RSBG. Root sampling of behavior hypotheses causes this approximation and is discussed in Sec. 3.5.3.

For ease of presentation, a recursive formulation of the planning algorithm is presented. The actual implementation uses a sequential call of selection, expansion, rollout, and backpropagation steps.

Algorithm 1 Main search method of the RSBG planner.

```
function SEARCH(o^t, Pr(·|H_o^t, ·))
    repeat
        for j = 1 ... N_{-i} do
            θ'_j ~ Pr(·|H_o^t, j)          ▷ Root sampling of hypotheses (cf. 3.5.3)
        SIMULATE(⟨o^t⟩, {θ'_1, ..., θ'_{N_{-i}}}, 1)
    until MAXITERATIONS() or TIMEOUT()
    return EGOACTIONSELECTION(⟨o^t⟩)    ▷ Greedy action selection (cf. 3.5.5)
```

Algorithm 2 Simulation step of the RSBG planner integrating selection, expansion and rollout.

function SIMULATE($\langle H_o \rangle, \{\theta'_1, \ldots, \theta'_{N_{-i}}\}, d$)
  if $d > d_{\max}$ **or** ISTERMINAL($\langle H_o \rangle$) **then**
    return 0 ▷ Terminal state or maximum search depth reached
  if FIRSTNODEVISIT($\langle H_o \rangle$) **then**
    INITNODESTATISTICS() ▷ Zeroing of action values and counts
    return RANDOMROLLOUT($\langle H_o \rangle, \{\theta'_1, \ldots, \theta'_{N_{-i}}\}, d$) ▷ Perform rollout (cf. 3.5.5) for newly expanded node
  $a_i \leftarrow$EGOACTIONSELECTION($\langle H_o \rangle$)
  for $l = 1 \ldots N_{-i}$ **do**
    $a_{-il} \leftarrow$OTHERACTIONSELECTION($\langle H_o \rangle, l, \theta'_l$)
  } Separation of ego and others' action selection
  $\tau_{\text{predict}} \leftarrow d \cdot \tau_a$ ▷ Search-depth-dependent prediction time
  $(o', r) \leftarrow$ENVIRONMENTMOVE($H_o, (a_i, a_{-i}), \tau_{\text{predict}}$) ▷ Prediction given joint action $a = (a_i, a_{-i})$
  $R' \leftarrow$SIMULATE($\langle H_o, (a_i, a_j), o' \rangle, \{\theta'_1, \ldots, \theta'_{N_{-i}}\}, d+1$) ▷ Recursive selection and expansion
  $R \leftarrow r + \gamma \cdot R'$
  $N(\langle H_o \rangle) \leftarrow N(\langle H_o \rangle) + 1$
  $N(\langle H_o \rangle, a_i, i) \leftarrow N(\langle H_o \rangle, a_i, i) + 1$ ▷ Backpropagation for UCT selection (cf. 3.5.5)
  $Q_R(\langle H_o \rangle, a_i, i) \leftarrow Q_R(\langle H_o \rangle, a_i, i) + (R - Q_R(\langle H_o \rangle, a_i, i))/N(\langle H_o \rangle, a_i, i)$
  for $l = 1 \ldots N_{-i}$ **do**
    $N(\langle H_o \rangle, a_{-il}, l) \leftarrow N(\langle H_o \rangle, a_{-il}, l) + 1$ ▷ Backpropagation for worst-case selection (cf. 3.5.4)
    $Q_R(\langle H_o \rangle, a_{-il}, l) \leftarrow Q_R(\langle H_o \rangle, a_{-il}, l) + (R - Q_R(\langle H_o \rangle, a_{-il}, l))/N(\langle H_o \rangle, a_{-il}, l)$
  return R

3.5.3. Root Sampling of Hypotheses

The BAMCP algorithm [217] applies root-sampling of parameters of the transition function. The RSBG planner employs root-sampling of types of other agents (cf. Alg. 1) which has the same effect of influencing the transitions to the next predicted environment states. In a discrete action and state-space BAMDP, action-observation histories are a sufficient statistic to implicitly represent the beliefs over transition parameters [217]. However, sampling the same actions and states from the belief becomes unlikely in continuous state and action spaces, making histories insufficient as a belief statistic. Without explicit tracking of beliefs in the nodes of the search tree, root sampling then leads to a QMDP approximation [151]. However, as discussed in the previous section, QMDP planning is a meaningful compromise in the context of RSBG planning in behavior spaces.

3.5.4. Worst-Case Action Selection of Other Agents

The RSBG defined in Sec. 3.4, lets each other agent j select individually a worst-case action a_j with respect to the ego action-value function from the agent's hypothesis action space. Obtaining the minimizing action requires knowledge about how the action influences the ego return. An exact calculation is not possible since the ego return depends on the future actions of all agents, including the actual ego policy. Instead, the worst-case actions are iteratively approximated during the search. For this, the expected action-returns *with respect to the ego agent's reward function*, $Q_R(\langle H_o \rangle, a_j, j)$ are maintained separately for each other agent j during back-propagation steps. The action selection of other agents switches between

- **action sampling** from the current root-sampled behavior hypothesis of that agent, $a_j \sim \pi_{\theta'_j}(\cdot|H_o)$. A new ego return for the sampled action is obtained after backpropagation, and

Algorithm 3 Worst-case action selection of other agents

function OTHERACTIONSELECTION($\langle H_o \rangle, j, \theta_j$)
  if $|A_j(\langle H_o \rangle)| \leq k_0 N(\langle H_o \rangle)^{\alpha_0}$ **then** ▷ Progressive widening
    $a_j \sim \pi_{\theta_j}(\cdot|H_o)$ ▷ Sampling from current root-sampled hypothesis
    $A_j(\langle H_o \rangle) \leftarrow A_j(\langle H_o \rangle) \cup \{a_j\}$ ▷ Adding to set of expanded actions
    return a_j
  else
    $a_j \leftarrow \text{argmin}_{a'_j \in A_j(\langle H_o \rangle)} Q_R(\langle H_o \rangle, a'_j, j)$ ▷ Worst-case action selection
    return a_j

- **worst-case selection** among the set of previously expanded actions $A_j(\langle H_o \rangle)$, i.e., selecting the action which *minimizes* the ego return $Q_R(\langle H_s \rangle, a_j, j)$.

Progressive widening [206] with parameters k_0 and α_0, is used to switch between these mechanisms. Depending on the number of expanded actions $|A_j(\langle H_o \rangle|$ and the node visit count $N_j(\langle H_o \rangle)$, a new action is sampled from the hypothesis (cf. Alg. 3). This approach ensures sufficient exploration of A^k to discover the subjective worst-case action.

Each expanded action in $A_j(\langle H_o \rangle)$ is affiliated with a specific hypothesis. Due to root sampling, these affiliations are distributed according to the posterior belief over hypotheses for that agent. The probability that a worst-case action is from the action space of a particular hypothesis is thus proportional to its posterior belief showing that this action selection strategy satisfies approximately the optimality criterion of the RSBG given in Def. 3.3. The authors of this thesis propose hypotheses-based worst-case action selection in [173] to avoid the approximative nature of the previous strategy. However, further research for Chapter 4 showed that the selection mechanism motivated here has the same benefits to approximate robustness-based optimality while avoiding frequent action changes between search iterations which, in SM-MCTS, can cause an insufficient depth of the search tree.

3.5.5. Ego Action Selection And Rollout Policy

A common strategy to select actions in MCTS is UCT [221] which selects an ego action a_i at stage node $\langle H_o \rangle$ maximizing

$$Q_{\text{UCT}}(\langle H_o \rangle, a') = Q_R(\langle H_o \rangle, a') + \kappa \cdot \sqrt{\frac{2 \ln N(\langle H_o \rangle)}{N(\langle H_o \rangle, a', i)}}. \tag{3.18}$$

with $N(\langle H_o \rangle)$ being the total visit count of the node, $N(\langle H_o \rangle, a', i)$ being the ego agent's selection count of action a' from previous search iterations and κ a parameter balancing exploration

Algorithm 4 Ego action selection using return normalization and UCT.

function EGOACTIONSELECTION($\langle H_o \rangle$)
  if NOTALLACTIONSEXPANDED() **then**
    return RANDOMUNEXPANDEDACTION()
  $R_{\text{min}} \leftarrow \min_{a' \in A_i} Q_R(\langle H_o \rangle, a', i)$
  $R_{\text{max}} \leftarrow \max_{a' \in A_i} Q_R(\langle H_o \rangle, a', i)$
  $a_i \leftarrow \text{argmax}_{a' \in A_i} \frac{R_{\text{min}} - Q_R(\langle H_o \rangle, a', i)}{R_{\text{max}} - R_{\text{min}}} + \kappa \cdot \sqrt{\frac{2 \ln N(\langle H_o \rangle)}{N(\langle H_o \rangle, a', i)}}$
  return a_i

Algorithm 5 Random rollout using root-sampled behavior hypotheses.

function RANDOMROLLOUT($\langle H_o \rangle, \{\theta'_1, \ldots, \theta'_{N_{-i}}\}, d$)
  if $d > d_{max}$ **or** ISTERMINAL($\langle H_o \rangle$) **then**
    return 0
  $a_i \sim \mathcal{U}(A_i)$
  for $l = 1 \ldots N_{-i}$ **do**
    $a_{jl} \leftarrow$OTHERACTIONSELECTION($\langle H_o \rangle, l, \theta'_l$)
  $\tau_{predict} \leftarrow d \cdot \tau_a$
  $(o', r) \leftarrow$ ENVIRONMENTMOVE($H_o, (a_i, a_j), \tau_{predict}$)
  return $r + \gamma \cdot$RANDOMROLLOUT($\langle H_o, (a_i, a_j), o' \rangle, \{\theta'_1, \ldots, \theta'_{N_{-i}}\}, d + 1$)

and exploitation. Action return values $Q_R(\langle H_o \rangle, a')$ are normalized between $[0, 1]$ by dynamically adapting normalization bounds based on the minimum and maximum of current return estimates. Function EGOACTIONSELECTION in Alg. 4 implements this selection step.

When multiple players select actions simultaneously and independently using Decoupled Upper Confidence Bound (DUCB), its deterministic nature can cause convergence to only local Nash equilibria [222] due to not all possible joint actions being sufficiently explored. In such cases EXP3 is often a better option [159, 201, 210, 222] as discussed in Sec. 3.5.1. However, selecting other agents' actions partially at random in the RSBG planner ensures sufficient variation of the ego returns and exploration of the joint action space. Therefore, UCT action selection is a viable option. Further, it has frequently been applied in existing interactive planning approaches for AVs [114, 117]. In Chapter 4 this ego action selection mechanism is replaced to enable risk-constrained planning.

The random rollout is a common approach to calculate heuristic values in MCTS at newly expanded nodes with the major advantage of not relying on any domain knowledge [159]. Algorithm 5 shows the rollout implementation used in the RSBG planner. In each rollout step, the ego action is sampled from the discrete set of ego actions. For each other agent, an action is sampled from the agent's root-sampled behavior hypothesis. The joint actions are executed until reaching a terminal state of the environment or the maximum search depth. The discounted cumulative reward is returned.

4 Risk-Constrained Interactive Planning in Behavior Spaces

This chapter presents the RC-RSBG planner, an interactive planning approach satisfying an interpretable risk formalism (cf. Sec. 1.2). It generates plans for which the observed statistic of safety envelope violations corresponds to the specified risk of violating a safety envelope given uncertainty about the behavior of other drivers. Specifically, this chapter

- develops an interpretable risk formalism defining the time-normalized risk of violating a safety envelope under uncertainty of other traffic participants' behavior,
- defines the Risk-Constrained Robust Stochastic Bayesian Game (RC-RSBG) integrating the interpretable risk formalism into the Robust Stochastic Bayesian Game (RSBG) presented in Chapter 3 to model risk-constrained interactive decisions under behavior uncertainty,
- extends the RSBG planner from Chapter 3 to solve the RC-RSBG. The RC-RSBG planner integrates risk-constrained ego action selection making use of a Constrained POMDP (C-POMDP) solver and backpropagation of risk estimates,
- defines safety envelopes for lane changing and intersection scenarios to be used with the RC-RSBG planner.

The work presented in this chapter is based on [173]. The chapter starts with motivating and formalizing the interpretable risk formalism in Sec. 4.1. The following Sec 4.2 presents the RC-RSBG. The accompanying risk-constrained interactive planner is outlined in Sec. 4.3 with the risk-constrained stochastic policy optimization being detailed in Sec. 4.4. Finally, Sec. 4.5 develops the definitions of the safety envelopes.

4.1. Developing an Interpretable Risk Formalism

This section motivates an interpretable risk formalism based on human safety violations in dense traffic. It formalizes such safety violations as envelope violation risk and defines the problem of risk-constrained interactive safety.

4.1.1. Leveraging Human Safety Statistics as Interpretable Risk Formalism

As discussed in Sec. 1.1 and 2.4, optimality criteria used in existing interactive planning approaches prevent an integration of interpretable risk. This section motivates an interpretable risk formalism by considering the safety statistics of human drivers in dense traffic. According to traffic laws, a driver must keep a sufficient safety distance from other drivers [6]. Analytical calculations of safe distances, which guarantee collision-freeness when all drivers adhere to them, are given in [223].

However, humans do not always adhere to legal safety in dense traffic. In [224], the authors evaluate the percentage of safe distance violations in the NGSIM dataset. They find a low number of violations. Pek et al. [7] integrate the response time of human drivers into the safe distance formulation and analyze the percentage of safe lane changes in the NGSIM dataset. Their more realistic safe distance formulation reveals that "only 39.83 % of the lane changes are classified as safe". Since the recorded traffic data is during the early morning, the rush hour presumably caused denser traffic. Therefore, their results can be interpreted as humans keeping a certain balance between safety and efficiency in dense traffic by violating safe distances during lane changes. Esterle et al. [6] analyze the percentage of violations *per driven time* in traffic situations where an ending lane requires vehicles to merge. They find in situations extracted from the INTERACTION dataset that human drivers do not keep a safe distance in 4 % to 8 % of driven time and detect that around 25 % of lane changing situations are unsafe. The above studies focus on the violation of longitudinal safety.

More complicated legal definitions of safety are also violated by human drivers. Pek [32] analyzes the number of failed plan verifications in a test drive through inner-city areas. The safety of a plan is defined using reachable sets in combination with fail-safe motion planning. They find that 1.64 % of scenarios cannot be verified as safe, with the majority of cases arising due to violations of the safe distance. Presumably, the number of violations is much lower in this case since it is averaged over the whole test route, which considered varying traffic density and maneuvers, i.e., lane changes, but also car-following situations.

The presented analyses show that humans violate formal definitions of safety, e.g., defined based on safe distances or more complex safety envelopes in certain traffic situations. Humans seem to balance safety and efficiency in a comprehensible way by

- **adhering to** a legal definition of safety *in most cases*. This situation can also be denoted as staying within a safety envelope given by the legal definition. The presented studies reveal that the violations are not equally distributed among scenarios. During lane changing, an increased amount of violations occurs compared to car-following situations.

- **accepting the risk** of violating the legal safety definition. Humans possibly compromise the uncertainty about the reactions of other drivers and the safety of the situation. Given the presented studies, it is therefore meaningful to assume that humans behave such that a formal safety definition is violated with some probability β over the driven time. Humans may make such a compromise ($\beta > 0$) only in low severity situations, e.g., dense traffic with low average traffic speed. They may adjust β to tune safety versus efficiency and

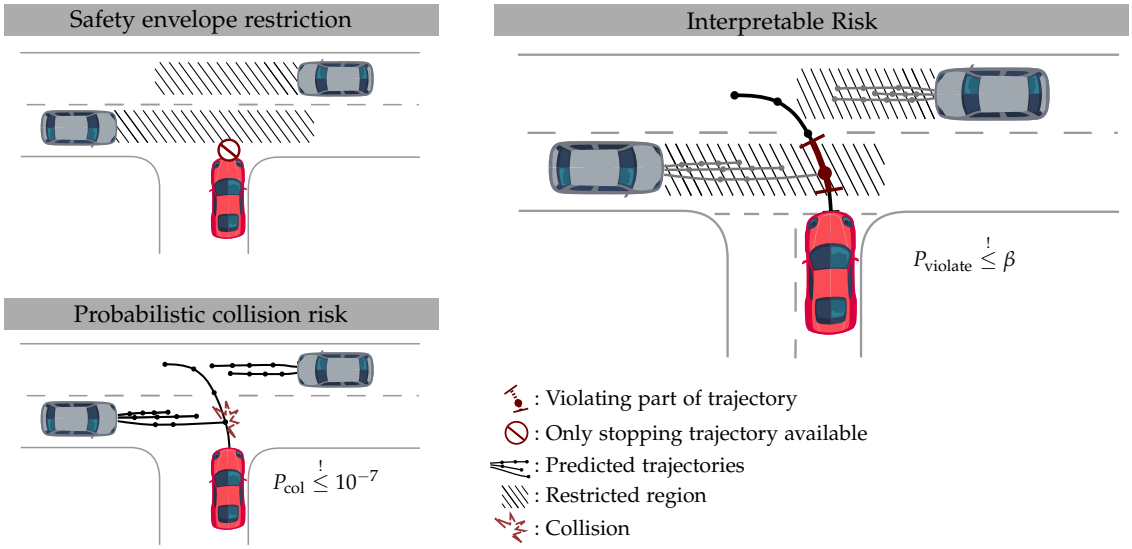


Figure 4.1.: The interpretable risk formalism is motivated by human safety statistics in dense traffic. The interactive planner generates a policy which satisfies a specified maximum risk of violating a safety envelope β. It combines definitions of legal safety, which may cause conservative driving by restricting the allowed planning region, with probabilistic risk estimations. Yet, it avoids employing the collision risk P_{col}, which is infeasible to calculate in an interactive planning approach, to express safety (modified graphic from [37], ©2021 IEEE).

avoid conservative driving in congested traffic. In higher severity traffic, e.g., on highways, they adhere to strict safety ($\beta = 0$).

Based on these considerations, a human-inspired risk concept for interactive behavior planning is stated:

> *"The ego vehicle behaves such that the percentage of time a legal safety definition is violated is smaller than a given risk threshold."*

A time-dependent risk measure is essential to achieve an interpretable risk formalism which is discussed further in the following section.

The severity of violating legal safety is not straightforwardly defined since such a violation does not necessarily lead to a hazardous event, e.g., a collision on which to base the definition of severity. Nonetheless, the term risk is used since it stands for a level of safety given by the maximum probability of violating a safety envelope over time. Fig. 4.1 motivates the proposed risk formalism from a different perspective. The formalism combines legal definitions of safety with inexact probabilistic predictions used in interactive planning into a unique concept to balance safety and efficiency.

4.1.2. Formalizing the Interpretable Risk of Safety Envelope Violations

A time-based formulation of the probability of violating a safety envelope is essential to achieve an interpretable risk formalism. Considering safety violations only at specific states during planning yields a differing safety statistic when the observed states and states visited during planning deviate. To avoid this and achieve a defined mapping between a specified risk level and

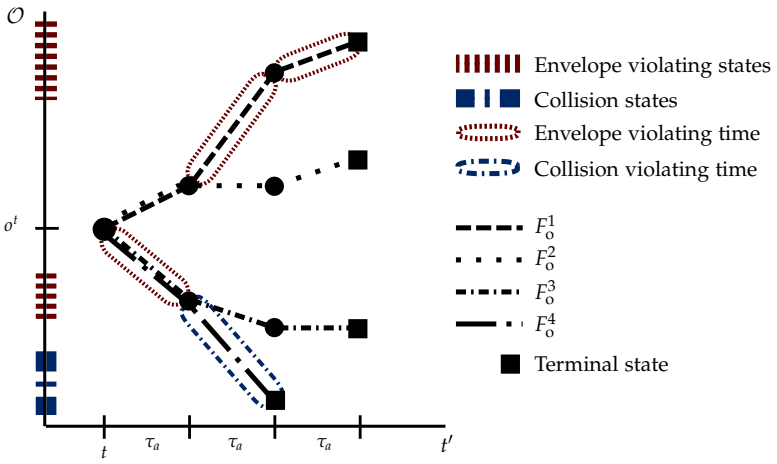


Figure 4.2.: Example for envelope violation and collision risk calculation. The future observation sequences F_o^{1-4} start from state o^t. It is assumed that these sequences occur with probabilities $\mathbb{P}(F_o^{1-3}) = 0.3$ and $\mathbb{P}(F_o^4) = 0.1$. The sequences end in terminal states. Sequence F_o^4 ends in a terminal collision state. When a state violates the safety envelope or collides the respective transition from the previous state in the sequence is marked as envelope or collision violating. Since sequence F_o^1 shows two and sequences F_o^{3-4} show one envelope violation, we obtain $\rho_{\text{env}}(\cdot) = 0.3 \cdot \frac{2\tau_a}{3\tau_a} + 0.3 \cdot \frac{0\tau_a}{3\tau_a} + 0.3 \cdot \frac{1\tau_a}{3\tau_a} + 0.1 \cdot \frac{1\tau_a}{2\tau_a} = 0.35$. The only sequence showing a collision violation is F_o^4 giving $\rho_{\text{col}}(\cdot) = 0.3 \cdot \frac{0\tau_a}{3\tau_a} + 0.3 \cdot \frac{0\tau_a}{3\tau_a} + 0.3 \cdot \frac{0\tau_a}{3\tau_a} + 0.1 \cdot \frac{1\tau_a}{2\tau_a} = 0.05$ (graphic from [37], ©2021 IEEE).

the observed safety statistic, the period between two states evaluated during planning must be incorporated into the risk formalism. Event-based risk formulations [46] represent such a setting by modeling that the harmful event, i.e., the envelope violation, occurs over some time. For the computation of event probabilities, complete trajectories of ego and other participants must be available. This requirement seems to contradict interactive settings in which behavior policies model interactions between participants on a fine-grained time scale. The following definition of a violation risk overcomes these difficulties. It handles interactivity and provides a time-based definition of risk.

Definition 4.1 (Violation risk) *Given the current environment state $o^t \in \mathcal{O}$, behavior policies π_i and π_j, a safety violation indicator $f : \mathcal{O} \to \{0, 1\}$ indicating a violation of a formal safety definition in observation state $o' \in \mathcal{O}$, then the violation risk is defined as:*

$$\rho(o^t, \pi_i, \pi_j, f) = \mathbb{E}_{F_o \sim \mathbb{P}^{o^t, \pi_i, \pi_j}} \left[\frac{\sum_{z=1}^{z=|F_o|-1} f(F_o(z)) \cdot \tau_a}{|F_o| \cdot \tau_a} \right] \tag{4.1}$$

The expectation is defined over the distribution $\mathbb{P}^{o^t, \pi_i, \pi_j}$ over future observation sequences $F_o = (o^t, o^{t+\tau_a}, o^{t+2\tau_a}, \ldots)$ starting from current environment state o^t. This distribution is influenced by ego and other agents' policies. The upper sum represents the violation duration within the observation sequence F_o with $|F_o|$ being the length of the sequence and $F_o(z)$ giving the z-th observation within the sequence. The lower term is the total duration of the sequence. The fraction of these terms yields the percentage of time the safety envelope is violated for one sequence. A sequence ends in a terminal state. The temporal resolution of this fraction is determined by the action duration τ_a. For fixed ego policy π_i, the expectation provides the

time-based violation risk under unknown behavior of other participants π_j.

Calculation of the violation risk can apply an arbitrary safety indicator function, e.g., for detecting safety envelope violations or collisions. Fig. 4.2 provides an example of envelope and collision risk calculations. The next section uses the definition of the violation risk to formalize the problem of risk-constrained interactive safety under behavior uncertainty.

4.1.3. Defining the Problem of Risk-Constrained Interactive Safety

Given the definition of the violation risk in Eq. (4.1), risk-constrained interactive safety under behavior uncertainty is defined using two risk constraints as follows.

Definition 4.2 (Risk-constrained interactive safety) *Given an indicator function for safety envelope violations $f_{envelope}$, the interactive planner generates a goal-directed policy π_i in the current environment state o^t under unknown behavior π_j of other participants which achieves a safety envelope violation risk lower than a specified allowed risk level β*

$$\rho(o^t, \pi_i, \pi_j, f_{envelope}) \mathrel{\widehat{=}} \rho_{env}(o^t, \pi_i, \pi_j) \overset{!}{\leq} \beta. \tag{4.2}$$

Further, given an indicator function for collision detection $f_{collision}$, it achieves a near-zero collision risk

$$\rho(o^t, \pi_i, \pi_j, f_{collision}) \mathrel{\widehat{=}} \rho_{col}(o^t, \pi_i, \pi_j) \overset{!}{\approx} 0 \tag{4.3}$$

The first constraint formalizes the maximum allowed risk of violating a safety envelope over time motivated in Sec. 4.1.1. However, given only this constraint there would exist a gap in the definition of optimality since in certain situations different ego behaviors, π_i and π_i', can provoke an equal envelope violation risk $\rho_{\text{env}}(o^t, \pi_i, \pi_j) = \rho_{\text{env}}(o^t, \pi_i', \pi_j)$ (cf. Fig. 4.3). Therefore, an additional collision risk is introduced to solve ambiguous situations by preferring ego behaviors with less probability of collision over time. Existing approaches applying collision risk (cf. Sec. 1.1) accept that solely a collision constraint expresses safety. In the definition of risk-constrained interactive safety, collision risk is required to be close to zero *only to resolve ambiguities*. The proposed risk formulation is independent of the goal formalism, i.e., the actual reward settings, and does not require a specific definition of the safety envelope, e.g., it could be based on safe distance measures or reachability analysis. The implementations of indicators used in this work are detailed in Sec. 4.5.

An interactive planner satisfying risk-constrained interactive safety must optimize a multi-objective criterion integrating the safety envelope and collision risk constraints and the goal-directed optimality criterion. Thereby, it must correctly approximate the observation sequence distribution given that the behavior of other agents is unknown. The following section extends the RSBG presented in Sec. 3.4 to satisfy the definition of risk-constrained interactive safety.

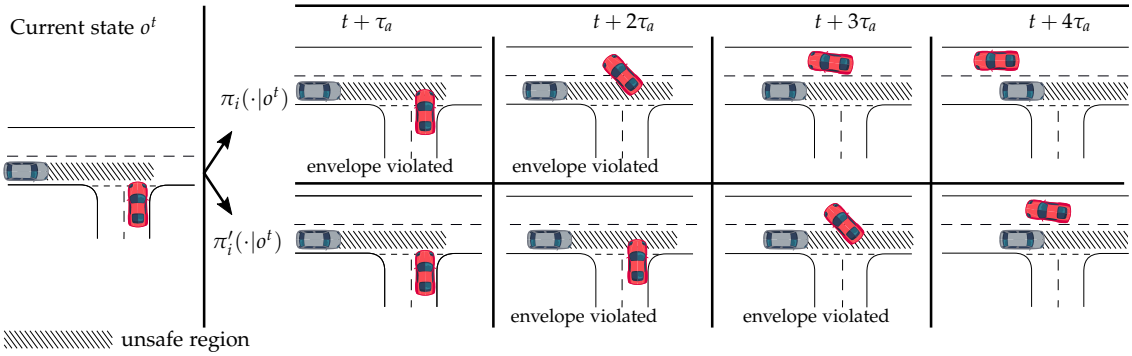


Figure 4.3.: Definition gap without near-zero collision risk constraint. Two policies π_i and π_i' can have an equal envelope violation risk due to violating the unsafe region in two of four predicted future states ($\rho_{\text{env}}(o^t, \pi_i, \pi_j) = \rho_{\text{env}}(o^t, \pi_i', \pi_j) = \frac{2\tau_a}{4\tau_a} = 0.5$). Introducing a near-zero constraint on collision risk resolves this ambiguity. It prefers policy π_i which cuts the safe distance of the oncoming vehicle at larger longitudinal distance and by that achieves lower collision risk ($\rho_{\text{col}}(o^t, \pi_i, \pi_j) < \rho_{\text{col}}(o^t, \pi_i', \pi_j)$). The envelope violation risk expresses the safety aspect whereas the collision risk constraint serves to resolve ambiguities. This combined definition circumvents that solely the collision risk expresses safety.

4.2. Risk-Constrained Robust Stochastic Bayesian Game

To define the Risk-Constrained Robust Stochastic Bayesian Game (RC-RSBG), constraint equations must be added to the optimality definition of the RSBG. The optimal policy of constrained Markov problems is often stochastic [225]. Therefore, the expected return of the RSBG is redefined to be dependent on a stochastic ego policy π, as

$$
\begin{aligned}
E^{\pi}(H_{\text{o}}) = \mathbb{E}_{a_{i,-i}}\left[Q_R^{\pi}(H_{\text{o}}, a_{i,-i})\right] \\
\text{with } a_i \sim \pi(\cdot|H_{\text{o}}), \\
\theta_{-i} \sim \Pr(\cdot|H_{\text{o}}), \\
a_{-i} = (a_{j,\text{wc}}, \ldots)_{\forall j} \;:\; a_{j,\text{wc}} = \underset{\substack{a_j' \in A_k \\ a_{-j}' \sim A_{-j}}}{\operatorname{argmin}} Q_R^{\pi}(H_{\text{o}}, (a_i, a_j', a_{-j}'))
\end{aligned}
\tag{4.4}
$$

The expectation is defined over the distribution of joint actions of the ego and other agents. Similar to the RSBG other agents choose worst-case actions out of the agent's hypotheses action space A_k given by the type combination θ_{-i}.

The Bellman equation integrating the stochastic policy thereby is

$$
Q_R^{\pi}(H_{\text{o}}, a) = u_i(o, a) + \gamma \cdot E^{\pi}(\langle H_{\text{o}}, a, o' \rangle) \tag{4.5}
$$

Next, the risk constraints for safety envelope violation and collision are integrated into the RSBG. For this, the mixture distribution $\widehat{\pi}_{j,\text{wc}}$ is defined as

$$
\widehat{\pi}_{j,\text{wc}}(a_j|H_{\text{o}}) = \sum_{\forall k} \Pr(\theta^k|H_{\text{o}}, j) \cdot \delta(a_j - a_{j,\text{wc}}) \tag{4.6}
$$

with $\delta(a_j - a_{j,\text{worst}})$ being a Dirac function located at the hypothesis-specific worst-case action

$a_{j,\text{wc}}$. The mixture distribution models the unknown behavior of other agents π_j in the constraints defined in Eq. (4.2) and Eq. (4.3). Due to the coverage-based definition of the prediction model over a behavior space, interpretability of the risk formalism remains even when the true behavior of others is not known. The worst-case action selection mechanism within a hypothesis motivated in Sec. 3.4.2 can be improved in a multi-objective problem domain. The calculation of $a_{j,\text{wc}}$ considers only the ego agent's envelope and collision violations without taking into account the ego agent's return. A mean cost action-value function is defined as

$$Q_{\overline{C}}^{\pi,\widehat{\pi}_{j,\text{wc}}}(H_\text{o}, a) = \frac{f_\text{envelope}(o') + f_\text{collision}(o')}{2} + \gamma \cdot \mathbb{E}_{a' \sim \pi,\widehat{\pi}_{j,\text{wc}} \forall j} Q_{\overline{C}}^{\pi,\widehat{\pi}_{j,\text{wc}}}(\langle H_\text{o}, a, o' \rangle, a') \tag{4.7}$$

giving a combined violation value when the agents execute joint action a, transition from state o to o' and from thereon follow policies π_i and $\widehat{\pi}_{j,\text{wc}}$. Given this definition, the worst-case actions are selected based on $Q_{\overline{C}}^{\pi,\widehat{\pi}_{j,\text{wc}}}$. As shown in Chapter 3, worst-case action selection reduces sample complexity for continuous action spaces of other agents. In a constrained optimality setting, it also helps to improve exploration of the joint actions that violate the given risk constraints.

With the above definitions, the optimality of RC-RSBGs is defined as follows.

Definition 4.3 (Optimality of RC-RSBGs) *The RC-RSBG applies similar state, action and agent definitions as the RSBG given in Def. 3.3. Its optimal stochastic policy π_i maximizes the expected cumulative reward, $\pi_i = \text{argmax}_\pi E^\pi(H_o^t)$, defined as*

$$\begin{aligned} E^\pi(H_o) = \mathbb{E}_{a_{i,-i}} \Big[& Q_R^\pi(H_o, a_{i,-i}) \Big] \\ & \textit{with } a_i \sim \pi(\cdot|H_o), \\ & \theta_{-i} \sim Pr(\cdot|H_o) \\ & a_{-i} = (a_{j,wc}, \ldots)_{\forall j} \; : \; a_{j,wc} = \underset{\substack{a'_j \in A_k \\ a'_{-j} \sim A_{-j}}}{\text{argmin}} Q_{\overline{C}}^{\pi,\widehat{\pi}_{j,wc}}(H_o, a_{i,j',-j'}) \end{aligned} \tag{4.8}$$

subject to the risk constraints

$$\begin{aligned} \rho_{env}(o^t, \pi_i, \widehat{\pi}_{j,wc}) &\overset{!}{\leq} \beta \\ \rho_{col}(o^t, \pi_i, \widehat{\pi}_{j,wc}) &\overset{!}{\approx} 0 \end{aligned} \tag{4.9}$$

with ρ_{env} as in Eq. (4.2), ρ_{col} as in Eq. (4.3), $\widehat{\pi}_{j,wc}$ as in Eq. (4.6) and $Q_{\overline{C}}^{\pi,\widehat{\pi}_{j,wc}}$ as in Eq. (4.7).

The RC-RSBG is a constrained stochastic decision problem and its optimal policy is therefore in general stochastic [225]. A stochastic policy may seem counter-intuitive in the context of autonomous driving since its lack of determinism impedes arguing safety. Yet, also human drivers show randomness in their microscopic behavior. Such microscopic variations may help them to resolve dense driving situations. A stochastic policy similarly models such variations while the risk constraints take care of defining safety.

The mutual dependence between the stochastic ego policy and the worst-case actions chosen

by other agents in the RC-RSBG impedes solving it optimally. Tiebraking can, however, be accomplished in an iterative planning procedure by using the violation values $Q_{\overline{C}}$ of the previous search iteration. The next sections extend the RSBG planner presented in Sec. 3.5 and develop such a mechanism to approximately solve the RC-RSBG.

4.3. Planning for the Risk-Constrained Robust Stochastic Bayesian Game

This section starts with reviewing related work on solving constraint decision-theoretic models. It then gives an overview of the RC-RSBG planner and discusses the backpropagation of risk estimates.

4.3.1. Review of Constrained-Based Decision-Theoretic Models and Solvers

This section reviews the suitability of existing single- and multi-objective decision-theoretic models and planning algorithms to integrate risk constraints.

Single-objective approaches are provided with only a single return signal from the environment to model constraints. Risk-sensitive models use risk measures applied to the distribution over the return to base decisions on the probability of worst-case outcomes [43]. Risk-sensitive models are preferable with respect to solution complexity since they can either be solved offline with distributional reinforcement learning [226–229] or online using a combination of parallelized Monte Carlo sampling and stochastic optimal control [230]. Other single-objective decision models guarantee a certain minimum return [231] with certain probability [232]. Solutions to such models are found with variants of Partially Observable Monte Carlo Planning (POMCP) planning and additional linear optimization steps [232]. Overall, single-objective models provide interpretability of constraints only when restricted states are always terminal, e.g., collision states, and provoke a single negative reward. When instead multiple negative and positive returns are accumulated, it becomes challenging to interpret the constraint quantitatively. The interpretable risk formalism (cf. Sec. 4.1.2) relies on accumulated violation costs and can thus not be expressed with single-objective models.

Multi-objective decision models integrate constraints by using mainly one of the following three concepts. Firstly, safe-reachability formulations avoid reward definitions by requiring that the probability of reaching a safe state is above a particular threshold [233]. Secondly, chance-constrained policies maximize the return while restricting the probability of visiting undesired states [234]. Chance-Constrained MDPs are employed in optimization-based trajectory planning [235, 236] or reinforcement learning [57]. For Chance-Constrained POMDPs (CC-POMDPs), Risk-bounded AO* (RAO*), an offline search-based planner, is applied [39, 164, 237] and adaptations of the Adaptive Belief Tree (ABT) planner [165]. Thirdly, in constrained problems, the policy maximizes the return while satisfying constraints on the expected costs [238]. Cost thereby defines an additional information signal coming from the environment. Optimal policies for Constrained MDPs are found iteratively by adapting the weighting of two separate cost and

	RSBG Planner	RC-RSBG Planner	SM-MCTS
Beginning of the Search Iteration	Sampling of hypothesis for each other agent	Sampling of hypothesis for each other agent, Gradient-based update of Lagrange multipliers	-
Selection	Ego action: UCB, Others' actions: Worst-case action within hypothesis concerning the ego return	Ego action: Risk-constrained stochastic policy optimization, Others' actions: Worst-case action within hypothesis concerning the combined ego envelope and collision cost	Ego action: UCB, Others' actions: UCB
Expansion, Rollout	Ego action: Random, Others' actions: Random within hypothesis	Ego action: Random, Others' actions: Random within hypothesis	Ego action: Random, Others' actions: Random
Back-propagation	Update of 1) ego return, 2) others' returns concerning the ego agent	Update of 1) ego return, 2) ego horizon-normalized envelope violation and collision risk and 3) others' combined cost concerning the ego agent	Update of 1) ego return and 2) others' returns

Table 4.1.: Comparison of RC-RSBG, RSBG (cf. Sec. 3.5) and SM-MCTS (cf. Sec. 3.5.1) planners.

return value functions [239] and constrained policy optimization [240]. Planning algorithms for Constrained POMDPs (C-POMDPs) use approximate Linear Programming (LP) in an interpolated belief space [241], Mixed Integer LP (MILP) in the dynamic programming updates [242], or a combination of offline planning with online branch-and-bound tree search [243]. Lee et al. [238] formulate the C-POMDP as POMDP by applying a LaGrange formalism. They extend the action selection of the POMCP planner with LP steps to output a stochastic policy. A Lagrange formalism is also used to solve Constrained Markov games in [244]. Chance-constrained models can be expressed using cost-constrained formulations by defining expected costs over the action-observation histories [234]. A naive approach which assigns a cost of one to risky states to model chance-constraints in a cost-constrained problem holds only when such states are additionally defined as terminal [237].

Summing over envelope violations in a single future observation sequence in the definition of the interpretable risk formalism (cf. Sec. 4.1.2) is related to an accumulation of costs. Approaches from the field of cost-constrained planning are therefore well suited to solve the RC-RSBG. The RC-RSBG planner, presented in the following sections, integrates a risk-constrained ego action selection mechanism based on the C-POMDP planner by Lee et al. [238].

4.3.2. Overview

The RC-RSBG planner is a constrained variant of SM-MCTS suitable for risk-constrained interactive planning. It targets the difficulties of planning under risk constraints when the satisfaction of constraints depends on the ego plan and the reactions of other traffic participants. Tab. 4.1 summarizes the differences between the RC-RSBG, RSBG and SM-MCTS planners.

The definition of the RC-RSBG in Def. (4.3) yields a strong interdependence between risk constraints, optimal ego policy, and worst-case predictions of other agents. The RC-RSBG planner

Algorithm 6 Simulation step of the RC-RSBG planner (with changes to the RSBG planner highlighted).

function SIMULATE($\langle H_o \rangle, \{\theta'_1, \ldots, \theta'_{N_{-i}}\}, d$)
  if $d > d_{\max}$ **or** ISTERMINAL($\langle H_o \rangle$) **then**
    return $[0, 0, 0, 0, 0]$
  if FIRSTNODEVISIT($\langle H_o \rangle$) **then**
    INITNODESTATISTICS()
    return RANDOMROLLOUT($\langle H_o \rangle, \{\theta'_1, \ldots, \theta'_{N_{-i}}\}, d$)
  $a_i \sim$EGOPOLICYOPTIMIZATION($\langle H_o \rangle, \kappa, \upsilon$)
  for $l = 1 \ldots N_{-i}$ **do**
    $a_{jl} \leftarrow$OTHERACTIONSELECTION($\langle H_o \rangle, l, \theta'_l$)
  $\tau_{\text{predict}} \leftarrow d \cdot \tau_a$
  $(o', r) \leftarrow$ ENVIRONMENTMOVE($H_o, (a_i, a_j), \tau_{\text{predict}}$)
  $[R', T'_{\text{env}}, T'_{\text{col}}, T'_{\text{tot}}, \overline{C}'] \leftarrow$SIMULATE($\langle H_o, (a_i, a_j), o' \rangle, \{\theta'_1, \ldots, \theta'_{N_{-i}}\}, d+1$)
  $[R, T_{\text{env}}, T_{\text{col}}, T_{\text{tot}}] \leftarrow [r + \gamma \cdot R',$
    $T'_{\text{env}} + f_{\text{envelope}}(o') \cdot \tau_{\text{predict}},$
    $T'_{\text{col}} + f_{\text{collision}}(o') \cdot \tau_{\text{predict}},$
    $T'_{\text{tot}} + \tau_{\text{predict}}]$ ▷ Backpropagation of violation durations (cf. 4.3.3)
  $N(\langle H_o \rangle) \leftarrow N(\langle H_o \rangle) + 1$
  $N(\langle H_o \rangle, a_i, i) \leftarrow N(\langle H_o \rangle, a_i, i) + 1$
  $Q_R(\langle H_o \rangle, a_i) \leftarrow Q_R(\langle H_o \rangle, a_i) + (R - Q_R(\langle H_o \rangle, a_i))/N(\langle H_o \rangle, a_i, i)$
  $\rho_{\text{env}}(\langle H_o \rangle, a_i) \leftarrow \rho_{\text{env}}(\langle H_o \rangle, a_i) + (T_{\text{env}}/T_{\text{tot}} - \rho_{\text{env}}(\langle H_o \rangle, a_i))/N(\langle H_o \rangle, a_i, i)$ ▷ Envelope risk update
  $\rho_{\text{col}}(\langle H_o \rangle, a_i) \leftarrow \rho_{\text{col}}(\langle H_o \rangle, a_i) + (T_{\text{col}}/T_{\text{tot}} - \rho_{\text{col}}(\langle H_o \rangle, a_i))/N(\langle H_o \rangle, a_i, i)$ ▷ Collision risk update
  for $l = 1 \ldots N_{-i}$ **do**
    $N(\langle H_o \rangle, a_{jl}, l) \leftarrow N(\langle H_o \rangle, a_{jl}, l) + 1$
    $\overline{C} \leftarrow [f_{\text{envelope}}(o') + f_{\text{collision}}(o')]/2 + \gamma \cdot \overline{C}'$
    $Q_{\overline{C}}(\langle H_o \rangle, a_{jl}, l) \leftarrow Q_{\overline{C}}(\langle H_o \rangle, a_{jl}, l) + (\overline{C} - Q_{\overline{C}}(\langle H_o \rangle, a_{jl}, l)/N(\langle H_o \rangle, a_{jl}, l)$ ▷ Mean violation update
  return $[R, T_{\text{env}}, T_{\text{col}}, T_{\text{tot}}, \overline{C}]$

finds an approximately optimal solution by resolving these interdependences in an interactive planning approach. For this, the RC-RSBG planner changes the RSBG planner (cf. Sec. 3.5) by

- **Horizon-normalized backpropagation:** To obtain accurate action-risk estimates, backpropagation additionally maintains the planning horizon reached in the curent search iteration.
- **Risk-based worst-case action selection:** The worst-case action selection is defined over the backpropagated risk estimates.
- **Risk-constrained ego action selection:** Ego actions are selected from a stochastic policy satisfying the risk-constraints. The algorithm is inspired by planning for C-POMDPs presented in [238].

The former two changes are discussed in the following Sec. 4.3.3. Risk-constrained ego action selection is then presented in Sec. 4.4.

4.3.3. Backpropagating Risk Estimates for Worst-Case Action Selection

In addition, to return estimates Q_R, the RC-RSBG planner maintains envelope violation and collision action-risks concerning the ego agent in each stage node. For this, the planner separately backpropagates the violation durations for safety envelope T_{env} and collision T_{col} occurred

Algorithm 7 Random rollout of the RC-RSBG planner (with changes to the RSBG planner highlighted).

function RANDOMROLLOUT($\langle H_o \rangle, \{\theta'_1, \ldots, \theta'_{N_{-i}}\}, d$)
  if $d > d_{\max}$ **or** ISTERMINAL($\langle H_o \rangle$) **then**
    return $[0, 0, 0, 0, 0]$
  $a_i \sim \mathcal{U}(A_i)$
  for $l = 1 \ldots N_{-i}$ **do**
    $a_{jl} \leftarrow$ OTHERACTIONSELECTION($\langle H_o \rangle, l, \theta'_l$)
  $\tau_{\text{predict}} \leftarrow d \cdot \tau_a$
  $[R', T'_{\text{env}}, T'_{\text{col}}, T'_{\text{tot}}, \overline{C}'] \leftarrow$ RANDOMROLLOUT($\langle H_o, (a_i, a_j), o' \rangle, \{\theta'_1, \ldots, \theta'_{N_{-i}}\}, d+1$)
  $[R, T_{\text{env}}, T_{\text{col}}, T_{\text{tot}}, \overline{C}] \leftarrow [r + \gamma \cdot R',$
    $T'_{\text{env}} + f_{\text{envelope}}(o') \cdot \tau_{\text{predict}},$
    $T'_{\text{col}} + f_{\text{collision}}(o') \cdot \tau_{\text{predict}},$
    $T'_{\text{tot}} + \tau_{\text{predict}},$
    $f_{\text{envelope}}(o') + f_{\text{collision}}(o')]/2 + \gamma \cdot \overline{C}']$
  return $[R, T_{\text{env}}, T_{\text{col}}, T_{\text{tot}}, \overline{C}]$

Algorithm 8 Worst-case action selection for other agents in the RC-RSBG planner (with changes to the RSBG planner highlighted).

function OTHERACTIONSELECTION($\langle H_o \rangle, j, \theta_j$)
  if $|A_j(\langle H_o \rangle)| \leq k_0 N_j(\langle H_o \rangle)^{\alpha_0}$ **then**
    $a_j \sim \pi_{\theta_j}(a_j | H_o)$
    $A_j(\langle H_o \rangle) \leftarrow A_j(\langle H_o \rangle) \cup \{a_j\}$
    return a_j
  else
    $a_j \leftarrow \text{argmax}_{a \in A_j(\langle H_o \rangle)} Q_{\overline{C}}(\langle H_o \rangle, a, j)$ ▷ Worst-case action selection over combined risk estimate
    return a_j

within the current iteration's selection, expansion (cf. Alg. 6) and rollout step (cf. Alg. 7). Further, it backpropagates the absolute planned horizon T_{tot} of the current iteration. These values correspond to the upper and lower term of the ratio defined in Eq. (4.1) and are used to update the ego-action risk estimates $\rho_{\text{env}}(\langle H_o \rangle, a_i)$ and $\rho_{\text{col}}(\langle H_o \rangle, a_i)$ during the backpropagation step in each traversed node. The other agents individually maintain a mean action-cost estimate $Q_{\overline{C}}$ defined according to Eq. (4.7). It represents the combined envelope and collision costs of the ego agent for the action a_{jl} selected by another agent. Note that the prediction time τ_{predict} increases with prediction depth, and therefore the backpropagated violation risk can become less accurate with increasing search depth. However, it rarely occurs that a tree state is *not violating* while an intermediate state between the tree states *is violating* the safety envelope. Due to using a time-based risk estimate, the violation risk is therefore likely being *overapproximated* which is in favor of the target of constraining the risk. This consideration again strengthens the importance of time-based risk estimates in interactive planning. A state-based risk estimate would not take into account differences due to varying prediction time steps. The return estimates Q_R are updated as usual.

Other agents select actions (cf. Alg. 8) similar to the RSBG planner by using a combination of worst-case selection and progressive widening. Progressive widening ensures that new actions are explored. In the other case, other agents select actions maximizing the combined envelope violation and collision cost of the ego agent $Q_{\overline{C}}$.

4.4. Selecting Ego-Actions using Risk-Constrained Stochastic Policy Optimization

This section presents the risk-constrained ego action selection of the RC-RSBG planner. The algorithm is inspired by the C-POMDP solver presented in [238].

4.4.1. Background on Solving Constrained Partially Observable Markov Decision Processes (C-POMDPs)

An optimal policy π of a Constrained POMDP (C-POMDP) [238, 242] maximizes the expected return V_R^π while with satisfying M constraints $\widehat{\boldsymbol{c}} = \{\widehat{c}_m\}_{m=1,\dots,M}$ for the expected costs $\mathbf{V}_C^\pi = \{V_{C_m}^\pi\}$. A belief-state Markov Decision Process (MDP) formulation of C-POMDPs can be given as [238]

$$\begin{aligned} \max_\pi V_R^\pi(\mathfrak{b}^0) &= \mathbb{E}_\pi\Big[\sum_{t=0}^{\infty}\gamma^t r(\mathfrak{b}^t, a^t)\Big|\mathfrak{b}^0\Big] \\ s.t.\ V_{C_m}^\pi(\mathfrak{b}^0) &= \mathbb{E}_\pi\Big[\sum_{t=0}^{\infty}\gamma^t c_m(\mathfrak{b}^t, a^t)\Big|\mathfrak{b}^0\Big] \le \widehat{c}_m\ \forall m \end{aligned} \tag{4.10}$$

with $c_m(\mathfrak{b}^t, a^t)$ being the immediate cost of action a^t in belief-state $\mathfrak{b}^t$ at time t.

Lee et al. [238] adapt the POMCP algorithm [203] to solve C-POMDPs. The following explanation is based on their work. They replace the constrained problem with the problem of minimizing Lagrange multipliers $\boldsymbol{\lambda} = \{\lambda_m\}_{m=1,\dots,M}$ in a dual LP formulation of the C-POMDP. When thinking of Lagrange multipliers as constants, this formulation is equal to finding an optimal policy maximizing a scalarized reward function

$$r(\mathfrak{b}^t, a^t) - \boldsymbol{\lambda}^\mathrm{T}\boldsymbol{c}(\mathfrak{b}^t, a^t) \tag{4.11}$$

in an *unconstrained* belief-state MDP. The optimal value function $V_{\boldsymbol{\lambda}}^*$ of this unconstrained POMDP can be solved for any $\boldsymbol{\lambda}$ using a standard POMDP solver, e.g., POMCP. The optimal lambda must thereby satisfy

$$\min_{\boldsymbol{\lambda}\ge 0}\left[V_{\boldsymbol{\lambda}}^*(\mathfrak{b}^0) + \boldsymbol{\lambda}^\mathrm{T}\widehat{\boldsymbol{c}}\right]. \tag{4.12}$$

By decomposing the combined value function $V_{\boldsymbol{\lambda}}^*$, the previous Eq. (4.12) can be rewritten as

$$\min_{\boldsymbol{\lambda}\ge 0}\left[V_R^{\pi_{\boldsymbol{\lambda}}^*}(\mathfrak{b}^0) - \boldsymbol{\lambda}^\mathrm{T}\mathbf{V}_C^{\pi_{\boldsymbol{\lambda}}^*} + \boldsymbol{\lambda}^\mathrm{T}\widehat{\boldsymbol{c}}\right] \tag{4.13}$$

with $\pi_{\boldsymbol{\lambda}}^*$ being the optimal stochastic policy for the scalarized reward function for a specific value of $\boldsymbol{\lambda}$.

Lee et al. [238] propose an iterative procedure to solve the dual problem defined by Eq. (4.13). They update the Lagrange multipliers iteratively at the beginning of each Monte Carlo Tree Search (MCTS) iteration using gradient descent $\Delta\lambda_m \sim Q_{C_m} - \widehat{c}_m$, $\forall m$, treating the optimal policy $\pi_{\boldsymbol{\lambda}}^*$ to be constant. In selection steps actions are sampled from a stochastic policy which

is calculated by solving a linear program given the current estimate of $\boldsymbol{\lambda}$. Further theoretical details on the algorithm are found in Lee et al. [238].

4.4.2. Comparison Between Solving C-POMDPs and RC-RSBGs

The C-POMDP planning approach outlined previously inspires the ego action selection mechanism of the RC-RSBG planner. Three differences arise in the context of multi-agent planning under risk constraints:

- **QMDP approximation:** With a similar motivation as discussed for the RSBG planner in Sec. 3.5.2, the RC-RSBG planner applies a QMDP approximation. Lee et al. [238] evaluate the C-POMDP planner in classical POMDP benchmarks and Atari game play. This work extends the approach to continuous action-observation spaces and applies it to interactive planning for AVs.

- **Normalization over the planning horizon:** Calculating the violation risk in Eq. (4.1) requires the distribution of future observation sequences and normalization with the respective length of each observation sequence. The required information is available in an MCTS planning approach. Each sequence of visited tree states during a single planning iteration corresponds to a future observation sequence. With an increasing number of iterations, these sequences of visited states approximate the distribution of future observation sequences. The respective sequence lengths are obtained during backpropagation. The C-POMDP planner in [238] is based on MCTS and can therefore be adapted straightforwardly to planning under violation risk constraints.

- **Avoidance of constraint updates:** In [238], the cost constraints are updated during interaction in the environment before each planning step based on the cost estimates and action probabilities of the previous time step. The authors argue that these updates ensure consistency of specified constraints and observed expected costs over an episode with multiple planning steps. However, when an executed action has a small probability in the planned stochastic policy, an extensive constraint update occurs with this approach, which causes sudden changes of the planned behavior in real-world applications. The advantage of time-normalized constraints, as given in the definition of the violation risk, is that the time-dependency cancels out, avoiding the need for constraint updates.

With these differences in mind, the envelope violation risk ρ_{env} and collision violation risk ρ_{col} in the RC-RSBG correspond to two cost terms ($M = 2$) with constraints β and 0, and the Lagrange multipliers λ_{env} and λ_{col}, in the C-POMDP formulation of Sec. 4.4.1.

4.4.3. Updating Lagrange Multipliers Using Gradient Estimates

Similar to [238], the search method of the RC-RSBG planner (cf. Alg. 9) performs gradient updates of the Lagrange multipliers λ_{env} and λ_{col} at the beginning of each search iteration. An ego action a_i is sampled from the root stochastic policy with $a_i \sim \textsc{EgoPolicyOptimization}(\langle o^t \rangle, 0, 0)$.

Algorithm 9 Gradient-updates of Lagrange multipliers in the RC-RSBG planner (with changes to the RSBG planner highlighted).

function SEARCH($o^t, \Pr(\theta^k | H_o^t, \cdot)$)
  $\lambda_{\text{env}}, \lambda_{\text{col}} \leftarrow [1, 1]$
  repeat
    for $j = 1 \dots N_{-i}$ **do**
      $\theta'_j \sim \Pr(\theta^k | H_o^t, j)$
    SIMULATE($\langle o^t \rangle, \{\theta'_1, \dots, \theta'_{N_{-i}}\}, 1$)
    $a_i \sim$ EGOPOLICYOPTIMIZATION($\langle o^t \rangle, 0, 0$)
    $\lambda_{\text{env}} \leftarrow \lambda_{\text{env}} + \alpha_n[\rho_{\text{env}}(\langle o^t \rangle, a_i) - \beta]$
    $\lambda_{\text{col}} \leftarrow \lambda_{\text{col}} + \alpha_n[\rho_{\text{col}}(\langle o^t \rangle, a_i) - 0]$
    Clip λ_{env}, λ_{col} to range $[0, 10]$
  until MAXITERATIONS()
  $a_i \sim$ EGOPOLICYOPTIMIZATION($\langle o^t \rangle, 0, v$)
return a_i

The calculation of the stochastic policy is described in the next Sec. 4.4.4. The action-risk estimates $\rho_{\text{env}}(\langle o^t \rangle, a_i)$ and $\rho_{\text{col}}(\langle o^t \rangle, a_i)$ at the root node are evaluated under this action sample and a gradient descent is performed to update the Lagrange multipliers with

$$\begin{aligned} \lambda_{\text{env}} &\leftarrow \lambda_{\text{env}} + \alpha_n[\rho_{\text{env}}(\langle o^t \rangle, a_i) - \beta] \\ \lambda_{\text{col}} &\leftarrow \lambda_{\text{col}} + \alpha_n[\rho_{\text{col}}(\langle o^t \rangle, a_i) - 0]. \end{aligned} \tag{4.14}$$

The gradient update takes into account the difference between the expected violation risks under the current stochastic policy and the desired envelope violation constraint β and collision violation constraint zero. The gradient step size α_n is chosen inversely proportional to the current iteration number $\alpha_n \sim 1/\text{ITERATIONNUM}()$. After the gradient update, the Lagrange multipliers are clipped as proposed in [238] to the range $[0, \frac{R_{\max} - R_{\min}}{\epsilon(1-\gamma)}]$. Using $\epsilon = 1$ [238] and, e.g., a discount factor $\gamma = 0.9$, and reward bounds, $R_{\max} = 1.0$ and $R_{\min} = 0.0$, the clipping range reduces to [0, 10].

4.4.4. Risk-Constrained Stochastic Action Selection

The RC-RSBG planner samples from a stochastic policy π_i in the selection, expansion and rollout steps, and in the main search method (cf. Alg. 6, 7 and 9). The calculation of the stochastic policy is similar to the algorithm for C-POMDPs presented in [238]. It is summarized in Alg. 10 and detailed in the following.

The stochastic policy maximizes a combined action-value including the current estimate of Lagrange multipliers and an UCT exploration term (cf. Eq. (3.18)):

$$Q_\lambda^\oplus(\langle H_o \rangle, a) = Q_R(\langle H_o \rangle, a, i) - \lambda_{\text{env}} \cdot \rho_{\text{env}}(\langle H_o \rangle, a) - \lambda_{\text{col}} \cdot \rho_{\text{col}}(\langle H_o \rangle, a) + \kappa \sqrt{\frac{\ln N(\langle H_o \rangle)}{N(\langle H_o \rangle, a, i)}} \tag{4.15}$$

with exploration parameter κ.

To account for inaccuracies in return and risk estimates, not only the maximizing action, but

Algorithm 10 Stochastic ego-policy optimization of the RC-RSBG planner
(with changes to the RSBG planner highlighted).

function EGOPOLICYOPTIMIZATION($\langle H_o \rangle, \kappa, \upsilon$)
  if NOTALLACTIONSEXPANDED() **then**
    return UNIFORMPOLICY()

  $Q^{\oplus}_{\lambda}(\langle H_o \rangle, a) \leftarrow Q_R(\langle H_o \rangle, a) - \lambda_{\text{env}} \cdot \rho_{\text{env}}(\langle H_o \rangle, a)$
    $\lambda_{\text{col}} \cdot \rho_{\text{col}}(\langle H_o \rangle, a) + \kappa \sqrt{\ln N(\langle H_o \rangle) / N(\langle H_o \rangle, a, i)}$
  $a^* \leftarrow \arg\max_a Q^{\oplus}_{\lambda}(\langle H_o \rangle, a)$
  $A^* \leftarrow$ Add other actions to a^* to consider exploration differences using Eq. (4.16)
  $\pi_i \leftarrow$ Solve linear program defined in Eq. (4.19) over A^* to obtain stochastic policy
  return π_i

an extended action set

$$A^* = \left\{ a^*_z \middle| |Q_\lambda(\langle H_o \rangle, a^*_z) - Q_\lambda(\langle H_o \rangle, a^*)| \leq \upsilon \cdot \left(\sqrt{\frac{\ln N(\langle H_o \rangle, a^*_z, i)}{N(\langle H_o \rangle, a^*_z, i)}} + \sqrt{\frac{\ln N(\langle H_o \rangle, a^*, i)}{N(\langle H_o \rangle, a^*, i)}} \right) \right\} \tag{4.16}$$

serves as support for the stochastic policy. Building the action set uses tolerance parameter υ to include, based on action selection counts, further actions apart from the return-maximizing action $a^* = \arg\max_a Q^{\oplus}_{\lambda}(\langle H_o \rangle, a)$. The value differences are thereby evaluated over the action-values

$$Q_\lambda(\langle H_o \rangle, a) = Q_R(\langle H_o \rangle, a, i) - \lambda_{\text{env}} \cdot \rho_{\text{env}}(\langle H_o \rangle, a) - \lambda_{\text{col}} \cdot \rho_{\text{col}}(\langle H_o \rangle, a) \tag{4.17}$$

without added exploration term.

The stochastic ego-policy with support A^* must satisfy the risk constraints with

$$\begin{aligned} \sum_{a_i \in A^*} \pi_i(a_i | \langle H_o \rangle) \cdot \rho_{\text{env}}(\langle H_o \rangle, a_i) &\overset{!}{\leq} \beta, \text{ and} \\ \sum_{a_i \in A^*} \pi_i(a_i | \langle H_o \rangle) \cdot \rho_{\text{col}}(\langle H_o \rangle, a_i) &\overset{!}{=} 0. \end{aligned} \tag{4.18}$$

Inaccuracies in the Monte Carlo value estimates and the collision constraint of zero prevent calculation of a policy always exactly satisfying these constraints. The following linear program

$$\begin{aligned} \min_{\{\epsilon^+_{\text{env}}, \epsilon^-_{\text{env}}, \epsilon^+_{\text{col}}, \epsilon^-_{\text{col}}\}} \quad & \lambda_{\text{env}} \cdot (\epsilon^+_{\text{env}} + \epsilon^-_{\text{env}}) + \lambda_{\text{col}} \cdot (\epsilon^+_{\text{col}} + \epsilon^-_{\text{col}}) \\ s.t. \quad & \sum_{l: a^*_l \in A^*} w_l \cdot \rho_{\text{env}}(\langle o^t \rangle, a_i) = \beta + (\epsilon^+_{\text{env}} - \epsilon^-_{\text{env}}) \\ & \sum_{l: a^*_l \in A^*} w_l \cdot \rho_{\text{col}}(\langle o^t \rangle, a_i) = 0 + (\epsilon^+_{\text{col}} - \epsilon^-_{\text{col}}) \\ & \sum_{l: a^*_l \in A^*} w_l = 1 \\ & \epsilon^+_{\text{env}}, \epsilon^-_{\text{env}}, \epsilon^+_{\text{col}}, \epsilon^-_{\text{col}}, w_l \geq 0 \end{aligned} \tag{4.19}$$

proposed in [238] accounts for estimation errors. It introduces the error variables ϵ^+_{col}, ϵ^-_{col}, ϵ^+_{env} and ϵ^-_{env}, in addition to the variables w_l representing the action probabilities of the stochastic

policy. It is solved in each call to EgoPolicyOptimization. The returned stochastic policy then approximately satisfies the envelope violation and collision risk constraints defined by the RC-RSBG. In rare cases, the linear program is not feasible given a node's current risk estimates. In this case, a deterministic policy is returned favoring the action with lowest collision risk.

4.5. Defining Safety Envelopes For Interactive Planning

The policy generated by the RC-RSBG planners satisfies the problem of risk-constrained interactive safety. This section presents implementations of indicator functions f_{envelope} to detect the violation of a safety envelope. The indicators should support computationally fast evaluation due to being repeatedly called in expansion and rollout steps. First, an envelope violation indicator for lane changing is defined, separately evaluating longitudinal and lateral violations. The following section then extends the indicator to support the evaluation of turning scenarios in intersections.

4.5.1. Envelope Violation Indicator for Lane Changing Scenarios

Longitudinal Violation

Rizaldi et al. [223] propose a model to guarantee longitudinal safety between a rear vehicle V_{R} at longitudinal position s_{r}^t and another vehicle V_{F} located in front at longitudinal position $s_{\text{f}}^t > s_{\text{r}}^t$, driving with velocities vel_{f}^t and vel_{r}^t at time t. The model assumes that the front vehicle performs a sudden emergency brake with maximum deceleration $\text{acc}_{\text{br,max}}^{t'}$, $\forall t' \geq t$ and the rear vehicle reacts after response time $T_{\text{r, react}}$ by applying maximum deceleration $\text{acc}_{\text{br,max}}^{t'}$, $\forall t' \geq t + T_{\text{r, react}}$. The authors distinguish four cases of relative stopping positions between vehicles by comparing braking durations to full standstill between both vehicles. If all four cases are satisfied at time t, the model guarantees longitudinal safety given that the above assumptions on response time and deceleration limits hold.

The formulation of Rizaldi et al. [223] is related to the Responsibility-Sensitive Safety (RSS) model [24], but less restrictive in the given application. RSS allows the ego vehicle to apply maximum acceleration during the response period giving more restrictive envelopes. In contrast, Rizaldi et al. [223] include the case that the front vehicle brakes with less acceleration than the ego vehicle, potentially allowing to maintain a smaller safe distance. Both approaches are similar when assuming no ego acceleration during the response time and equal minimum and maximum braking accelerations. Planning for the ego vehicle gives controllability of the ego action and therefore allows to apply the less restrictive formulation of Rizaldi et al. [223].

A function $v_{\text{lon. safe}}(a, b, o^t)$ is defined to represent the longitudinal envelope violations in the indicator definition. It returns true if vehicle a violates longitudinal safe distance $s_{\text{safe,ab}}$ with respect to its front vehicle b, or vice versa vehicle b violates longitudinal safe distance $s_{\text{safe,ba}}$ with respect to its front vehicle a in observation state o^t according to the model of Rizaldi et al. [223].

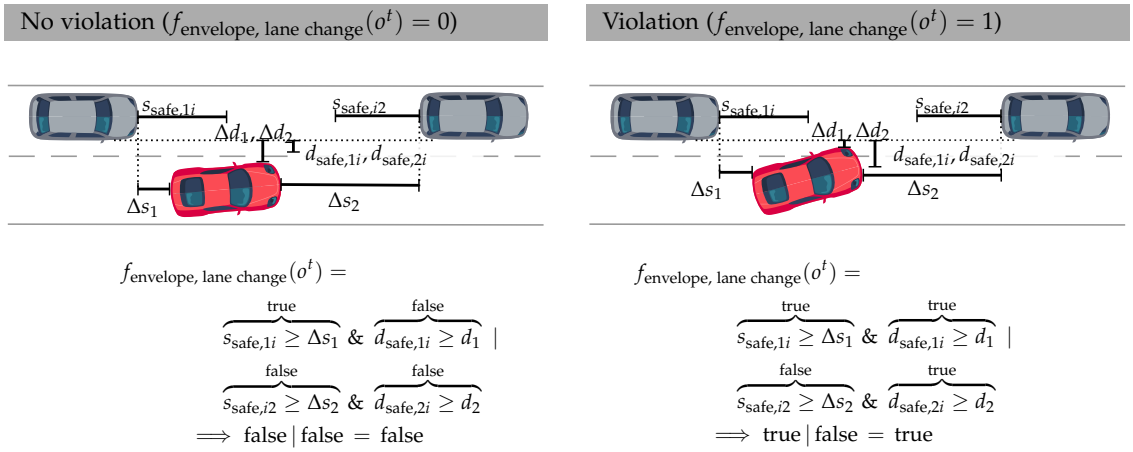


Figure 4.4.: Example calculation of the indicator function for a lane changing scenario. A violation is indicated if longitudinal and lateral safe distances, $s_{\text{safe},xi}$ and $d_{\text{safe},xi}$, exceed relative longitudinal and lateral distances, Δs_x and Δd_x, for a single other vehicle.

Lateral Violation

The formulation of Rizaldi et al. [223] does not include lateral safety. Therefore, the lateral safe distance definition of RSS [24] serves to check lateral envelope violations. For two vehicles V_1 and V_2, with lateral velocities v_1 and v_2 defined with respect to the center line of V_1's lane, the safe lateral distance according to RSS* is

$$d_{\text{safe, 12}} = \left[\frac{v_1 + v_{1,T}}{2} \cdot T_{1,\text{ react}} + \frac{v_{1,T}^2}{2 \cdot \text{acc}^t_{\text{lat, min}}} - \left(\frac{v_2 + v_{2,T}}{2} \cdot T_{2,\text{ react}} - \frac{v_{2,T}^2}{2 \cdot \text{acc}^t_{\text{lat, min}}} \right) \right]_+ \tag{4.20}$$

Thereby, V_1 is located in driving direction on the left side of vehicle V_2, and $v_{1,T} = v_1 + T_{1,\text{ react}} \cdot \text{acc}^t_{\text{lat, max}}$ and $v_{2,T} = v_2 + T_{2,\text{ react}} \cdot \text{acc}^t_{\text{lat, max}}$. This safe distance formulation applies if the two vehicles accelerate within the response times $T_{1,\text{ react}}$ and $T_{2,\text{ react}}$ towards each other laterally with maximum accelerations of $\text{acc}^t_{\text{lat, max}}$. After the response time, they must brake laterally with at least $\text{acc}^t_{\text{lat, min}}$. The assumption of RSS that vehicles accelerate towards each other with $\text{acc}^t_{\text{lat, max}}$ yields restrictive envelopes. Similar to the longitudinal safe distance formulation, it is meaningful to assume zero acceleration during the response period, giving $\text{acc}^t_{\text{lat, max}} = 0$ for both the ego and other vehicles. Planning for the ego vehicle gives controllability over its acceleration. For other vehicles, it is assumed that they do not deviate laterally from their center lines.

A function $v_{\text{lat. safe}}(a, b, o^t)$ is defined to represent the lateral envelope violations in the indicator definition. It returns true if vehicle a violates the lateral safe distance defined in Eq. (4.20) in observation state o^t under the above assumptions.

*The μ-lateral velocity used for the definition in [24] is not relevant for the definition of the safety envelope in this work.

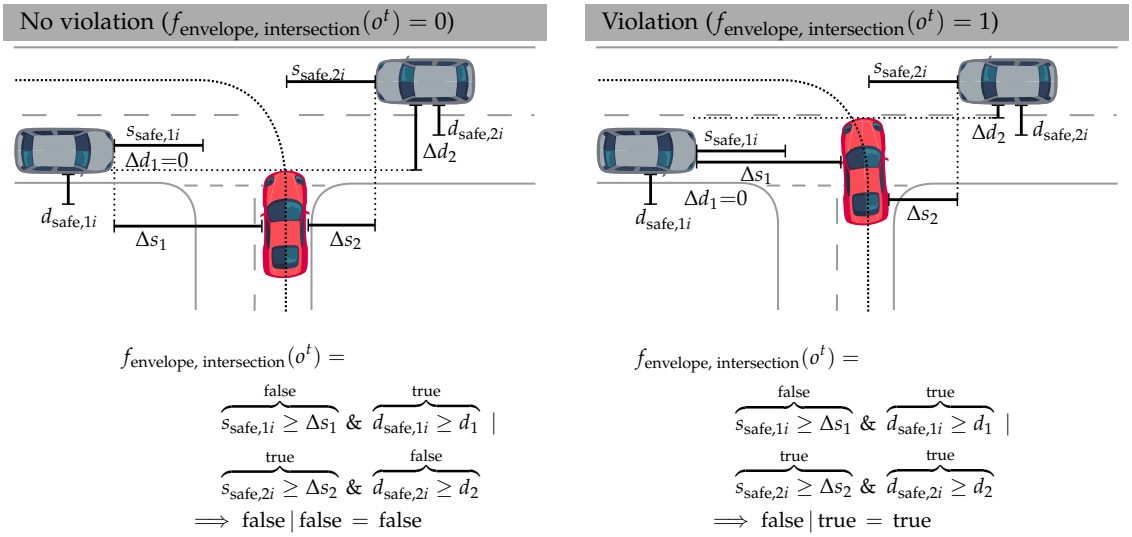


Figure 4.5.: Example calculation of the indicator function for an intersection scenario. A violation is indicated if longitudinal and lateral safe distances, $s_{\text{safe},xi}$ and $d_{\text{safe},xi}$, exceed relative longitudinal and lateral distances, Δs_x and Δd_x, for a single other vehicle.

Indicator Definition

An indicator function checking the violation of the safety envelope at time t in lane changing is then defined as

$$f_{\text{envelope, lane change}}(o^t) = \begin{cases} 1 & \text{if } \exists j \in \{1, \dots, N_{-i}\},\ v_{\text{lon. safe}}(j, i, o^t) \quad \& \quad v_{\text{lat. safe}}(j, i, o^t) \\ 0 & \text{else} \end{cases} \tag{4.21}$$

An envelope violation occurs if both longitudinal and lateral violations coincide with any of the other vehicles. Fig. 4.4 gives examples of violating and non-violating envelopes in lane changing. Parameters of this envelope definition are the maximum acceleration and deceleration of the ego vehicle and other vehicles $\text{acc}_{\text{max, ego}}$ and $\text{acc}_{\text{max, other}}$, and the response times of ego $T_{\text{ego, react}}$ and other vehicles $T_{\text{other, react}}$. The lateral and longitudinal maximum accelerations required for calculating lateral and longitudinal safe distances are obtained from $\text{acc}_{\text{max, ego}}$ and $\text{acc}_{\text{max, other}}$ by Frenet transformation and assuming a single-track vehicle model.

4.5.2. Envelope Violation Indicator for Intersection Scenarios

Fig. 4.5 depicts a left-turning situation of the ego vehicle at an intersection with the main road being occupied by oncoming cars. The safe distance formulation of Rizaldi et al. [223] is only valid when the front and rear vehicle partially occupy the same lane. This condition does not hold in an intersection scenario when the ego vehicle has not yet entered the intersection. This section extends the indicator definition of the previous section to this situation.

In intersections, different longitudinal orderings between the crossing vehicles exist. These depend on the time points of passing the intersecting area and can be used to extend the longitudinal safe distance formulations presented previously to intersections [24]. However,

determining such orderings is not meaningful in interactive planning since the reactions of the vehicles onto each other determine the actual ordering. Shalev-Shwartz et al. [24] define a safe distance formulation for the unstructured situation based on a comparison of physically feasible trajectories. Such concepts are related to reachable set formulations (cf. Sec. 1.1.1) and are computationally too demanding to be applied with the RC-RSBG planner.

An efficient approach for longitudinal safe distance calculation avoiding comparison of longitudinal distances to the crossing point to estimate the ordering of vehicles is to assume that the ego vehicle, turning into the main road, virtually occupies the lane of another oncoming vehicle. The virtual ego vehicle has a longitudinal distance and velocity obtained via a Frenet transformation with respect to the center line of the other car. A safe longitudinal distance is then calculated as if both the ego and the other vehicle's route are in complete contact as given in the lane-changing case. The lateral safe distance formulation remains unchanged.

Apart from this adaptation of the longitudinal safe distance formulation, the indicator function for evaluating violation of the safety envelope in intersection scenarios $f_{\text{envelope, intersection}}$ is similarly defined as in the lane changing case. Examples of violating and non-violating envelopes in the intersection scenario are given in Fig. 4.4.

5 Experience-Based and Parallelized Risk-Constrained Planning

This chapter presents enhancements of the Risk-Constrained Robust Stochastic Bayesian Game (RC-RSBG) planner that enable online planning. Existing concepts accelerating single-objective planning with learned prior experience and parallelization are transferred and adapted to the multi-objective case. Specifically, this chapter contributes

- value-guided exploration which integrates prior experience into the RC-RSBG planner and benefits from the increase of information available in multi-objective planning,
- an input feature representation and architecture of neural networks to learn prior value experiences for interactive planning in a supervised training process,
- a data generation process which distributes the demonstration data for supervised learning similarly to the exploration distribution confronted with during online planning,
- a parallelized implementation of the RC-RSBG planner to additionally reduce computational demands.

The work presented in this chapter is based on [245] proposing experience-based planning for static environments and [246] evaluating neural network input representations for value function learning. It starts with a review on accelerating online planning with prior experiences in Sec. 5.1. Integration of prior experiences into the RC-RSBG planner is described in Sec. 5.2. Sec. 5.3 discusses training and representation of prior experiences using neural networks. The data generation process is developed in Sec. 5.4. Finally, Sec. 5.5 outlines the benefits of a parallelized implementation of the RC-RSBG planner.

5.1. Review on Accelerating Online Planning with Prior Experience

This section gives an overview of selected related work in domains other than autonomous driving that use offline experiences to improve and accelerate exploration in online planning. The focus is thereby on methods that separate offline experience collection and online planning.

Early work in this domain focuses on manually tuned heuristics. Knowledge about the probability of expert moves is incorporated into UCT action selection with progressive bias [247, 248]. It prioritizes expert moves in the UCT formula. Prioritization degrades with increasing node visit count. Progressive history [249] additionally includes information about the reliability of the domain knowledge. Both concepts assume that the optimality of moves is independent of the environment state. Pattern matching prioritizes moves that agree with patterns of the environment state [250]. Follow-up work by Gelly and Silver [251] showed that using a linear function approximator, predicting action-values and learned from self-play, improves playing strength over Go programs with only manually tuned heuristics. It integrates learned action values either as default selection policy or for initialization of node action-values. Silver et al. [252] present Dyna-2, which combines offline learning of a permanent experience memory with online correction using a transient memory for sample-based search.

This early work on combining Monte Carlo Tree Search (MCTS) with learned policy and value networks supported one of the breakthroughs in artificial intelligence. AlphaGo, which uses a combination of supervised learning and self-play, managed to beat the best human players in the game of Go [169]. AlphaGo's successors learn completely from self-play [170] and also master other games such as Chess and Shogi. Related ideas using learned value and policy estimates predominantly to adapt the UCT selection formula are also applied to Atari gameplay [253]. The authors in [254] express the combination of UCT action selection with learned stochastic policies as regularized policy optimization and, based on this, derive several improvements to the action selection procedure of AlphaZero.

Other forms to integrate prior experiences into sampling-based online planning are presented next. In [255], the authors compare various machine learning approaches such as linear regression, decision trees, and neural networks for predicting the solvability of card games in order to accelerate a depth-first search planner. Offline learned winning probabilities are used in [256] to switch between action selection and backpropagation in MCTS and in [257] to predict the return at the end of a random rollout. Li et al. [258] trained a neural network with supervised learning to estimate a correction factor for a standard heuristic. Pareekutty et al. [259] use value iteration to iteratively create a quality grid map online during planning to guide the node expansion of a Rapidly Exploring Random Tree (RRT) planner. Online experience creation is also used in [218] to approximate value functions for belief-state planning in Bayes-Adaptive MDPs (BAMDPs). The approach extends Bayes-Adaptive Monte Carlo Planning (BAMCP) discussed in Sec. 3.5.1 to continuous state spaces. The authors combine beliefs with function approximation by using belief particles as feature inputs to the value function. In discrete imperfect information environments, e.g., the game of heads-up no-limit poker, an abstracted representation of the

probability distribution over others players' cards can be directly used as a feature input for a neural network function approximator [216].

Previous work on data generation for learning experiences is sparse. Self-play is meaningful in domains of game playing [170, 257], yet, not straightforwardly applicable to autonomous driving. The authors in [260] present a data collection process that samples initial states from distributions of potential environments and lets the planning agent act under mixtures of a random and learned policy to collect action values of each encountered environment state. The approach is related to the approach presented in Sec. 5.4, yet, has been evaluated only in discrete, grid environments.

Overall, existing work focuses on single-objective planning domains. In these domains, prior experience is straightforwardly integrated into UCT action selection by weighting the exploration term [117, 118, 171, 251]. Interactive planning for autonomous driving also applies this concept [117, 118, 171]. This is reasonable when the stochastic policy is obtained from a softmax calculation over action-values [251] or based on action-counts [254] and therefore expresses count-based or value-based action preferences from prior search runs. In contrast, the RC-RSBG planner outputs a truly stochastic policy not associated with action-counts but obtained from solving a linear optimization problem to fulfill envelope and collision risk constraints. It is thus not reasonable to combine the stochastic policy of the RC-RSBG planner with the exploration term of the combined reward and cost action values in procedure EgoPolicyOptimization. It remains unclear what concepts to use for the RC-RSBG planner, and generally in multiobjective planning, to benefit from offline learned experiences. Further, existing work misses a data generation process for offline experience learning tailored to the domain of autonomous driving.

5.2. Value-Guided Risk-Constrained Planning

This section presents an approach to guide the search of the RC-RSBG planner by prior experiences available in the form of value estimates. Warm starting [159] is a concept to guide the exploration of an MCTS algorithm with prior knowledge by initializing the node's action values at node creation. This concept is applied to UCT action selection in [251]. Warm starting is also meaningful with the RC-RSBG planner. It provides valid information to calculate the risk-constrained policy in procedure EgoPolicyOptimization directly after node initialization.

In the RC-RSBG planner, given prior experience in form of action-returns $Q_R^{\text{prior}}(o, a_i)$, envelope action risks $\rho_{\text{env}}^{\text{prior}}(o, a_i)$ and collision action risks $\rho_{\text{col}}^{\text{prior}}(o, a_i)$ for a search state o, warm starting initializes a node $\langle H_{\text{o}} \rangle$ with

$$\begin{aligned} Q_R(\langle H_{\text{o}} \rangle, a_i) &\leftarrow Q_R^{\text{prior}}(o, a_i) \\ \rho_{\text{env}}(\langle H_{\text{o}} \rangle, a_i) &\leftarrow \rho_{\text{env}}^{\text{prior}}(o, a_i) \\ \rho_{\text{col}}(\langle H_{\text{o}} \rangle, a_i) &\leftarrow \rho_{\text{col}}^{\text{prior}}(o, a_i) \end{aligned} \tag{5.1}$$

with o being the last state in action observation history H_{o}. This initialization is performed in procedure InitNodeStatistics (cf. Alg. 6).

The initialization with experiences replaces the role of the rollout heuristics guiding the search. The rollout step can thus be dropped to reduce computational demands. The newly initialized risk and return values are not included in the backpropagation. Backpropagation of the prior knowledge information would lead to an incorrect normalized risk estimate. The backpropagated planning horizon length does not fit the normalization already inserted in the prior risk estimates. Therefore, the prior knowledge is only used for warm starting such that action selection in subsequent node visits benefits from this information. That is a potential limitation of warm starting in the case of the RC-RSBG planner.

5.3. Offline Training of Value Experiences

This section presents supervised learning of value experiences with Neural Networks (NNs), designs meaningful input features for the neural network, and discusses relevant properties of NN architectures used for experience learning.

5.3.1. Supervised Learning and Loss Function Definition

The prior experience about return and risk values is represented by a NN trained using supervised learning. The training data set $\mathbb{D}_{\text{train}}$ consists of experience tuples

$$\mathbf{e}(o) = \big(\underbrace{(\mathbf{i}^i, \mathbf{i}^1, \mathbf{i}^2, \ldots, \mathbf{i}^{N_{-i}})}_{\text{NN input features}}, \tag{5.2}$$

$$\underbrace{(q^1, q^2, \ldots, q^{W_i})}_{\text{Return action values}}, \tag{5.3}$$

$$\underbrace{(\rho^1_{\text{env}}, \rho^2_{\text{env}}, \ldots, \rho^{W_i}_{\text{env}})}_{\text{Envelope violation risks}}, \tag{5.4}$$

$$\underbrace{(\rho^1_{\text{col}}, \rho^2_{\text{col}}, \ldots, \rho^{W_i}_{\text{col}})}_{\text{Collision violation risks}} \big) \tag{5.5}$$

which contain 1) the agent-specific neural network input features extracted from state o and 2) the action value estimates for the three value types, the return values, the envelope violation risks and the collision violation risks which are represented individually for each of the $W_i = |A_i|$ ego actions. The creation of the training data is explained in Sec. 5.4.

The value function network g_v predicts all three value types

$$g_v(\mathbf{i}^i, \mathbf{i}^1, \mathbf{i}^2, \ldots, \mathbf{i}^{N_{-i}}) \rightarrow \left[Q_R^{\text{prior}}, \rho_{\text{env}}^{\text{prior}}, \rho_{\text{col}}^{\text{prior}} \right] \tag{5.6}$$

based on the NN input features. The supervised training of the NN uses maximum log-likelihood estimation and stochastic gradient descent over batches sampled from the training data set [261]. Specifically, the training applies a mean squared error loss $MSE(p,q) = \frac{1}{N}\sum_{i=0}^{N}(p(i)-q(i))^2$ calculated between the outputted return action values Q_R^{prior}, envelope violation action risks $\rho_{\text{env}}^{\text{prior}}$ and collision action risks $\rho_{\text{col}}^{\text{prior}}$, and the respective value information in the experience

tuple, $(q^1, q^2, \ldots, q^{W_i})$, $(\rho^1_{\text{env}}, \rho^2_{\text{env}}, \ldots, \rho^{W_i}_{\text{env}})$ and $(\rho^1_{\text{col}}, \rho^2_{\text{col}}, \ldots, \rho^{W_i}_{\text{col}})$.

5.3.2. Neural Network Input Features

A large body of work exists on the definition of appropriate neural network feature representations encoding traffic environments. Commonly included features are the absolute state of the ego vehicle and relative states to other vehicles [134, 144]. Position information is either explicitly encoded using separate features [134, 144] or implicitly using a grid-based environment representation [141, 145]. Generalizing to different maps requires features that represent lane and road information [141]. Relational [141] or graph-based representations [166] aim at better generalization. Since there is a trade-off in experience learning between achieving a low inference time of the neural network and good generalization for states not in the training data, the feature representation defined in the following relies on absolute and relative state properties. Graph-based and grid-based features require specific neural network architectures, increasing inference time. The influence of architecture on inference time is further discussed in the following Sec. 5.3.3.

The RC-RSBG planner employs a belief-based prediction of other traffic participants, also suggesting to condition the prediction of prior experiences onto belief information. Though adding beliefs can increase the accuracy of learned experiences, generalization is impeded since adding beliefs increases the size of the NN input space. For instance, given a hypotheses set size of $K = 16$ and considering $N_{-i} = 4$ other agents in the input representation, leads to an additional number of $K \cdot N_{-i} = 64$ input features to represent all belief information. Taking beliefs as neural network input features thus rather worsens the prediction capability due to the curse of dimensionality. Instead of using belief information, the uncertainty over the behavior of other drivers is thus incorporated into the data generation process presented in Sec. 5.4.

Therefore, the input of the neural networks consists of a non-belief-based feature representation extracted from predicted environment state o^t. The ego agent features are defined as

$$\mathbf{i}^i = (s_i, d_i, \alpha_i^F, vel_i^{\text{lon}}, vel_i^{\text{lat}}, lane_i) \tag{5.7}$$

and consist of the absolute Frenet state with s_i and d_i being the longitudinal and lateral coordinates, and vel_i^{lon} and vel_i^{lat}, being the longitudinal and lateral velocity with respect to the center line of the ego vehicle's current lane. Further, the representation includes the orientation α_i^F with respect to the center line and information about the current lane with $lane_i$ being a lane index. The lateral ego coordinate has a positive sign when being on the left side of the center line in travel direction of the lane and a negative sign on the opposite side. The lateral velocity is positive when the ego vehicle steers from right to left in driving direction and negative when steering from left to right.

The representation of other agents includes the differences between the Frenet coordinates and velocities of the respective agent j and the ego vehicle obtained with respect to the ego center line:

$$\mathbf{i}^j = (\Delta s_j, \Delta d_j, \Delta v_j^{\text{lon}}, \Delta v_j^{\text{lat}}) \tag{5.8}$$

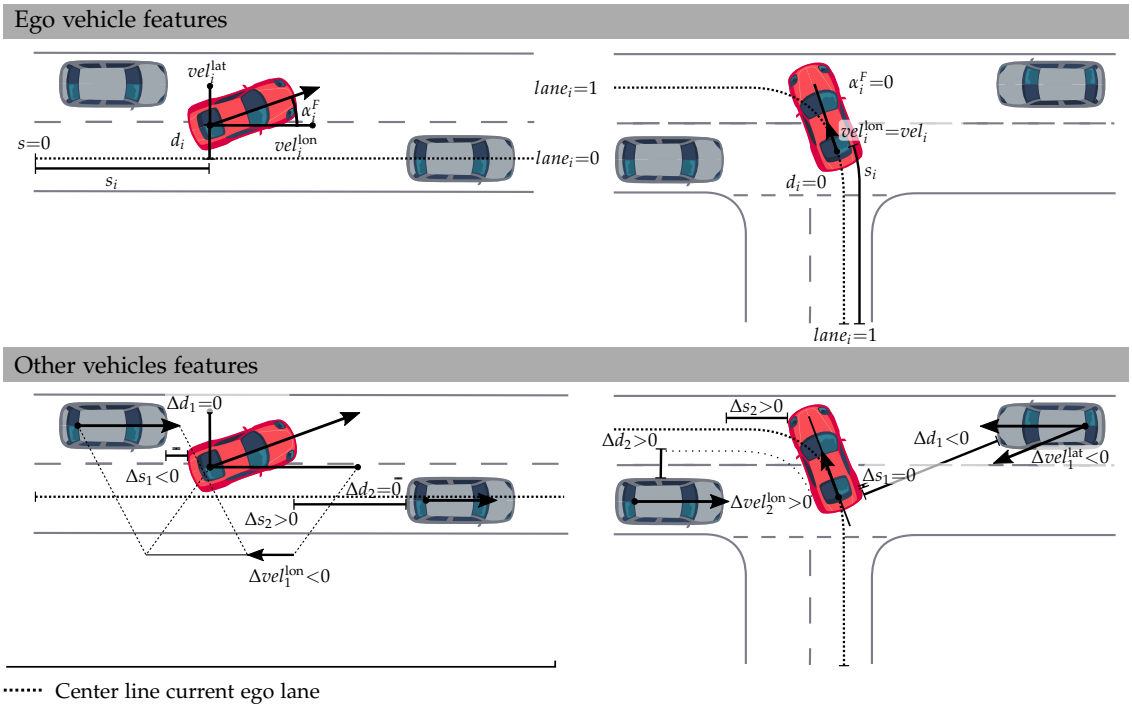


Figure 5.1.: Extraction of neural network input features from environment states.

The coordinate differences are calculated between the closest points of the ego and other vehicles' shapes, thereby also considering the shape orientations. If an overlap of the shapes in either longitudinal or lateral direction exists, the respective coordinate is zeroed.

The extraction of features is visualized in Fig. 5.1. All features are normalized to the range [-1.0, 1.0] before being passed to the neural network.

5.3.3. Compromising Generalization and Inference Time in Experience Learning

The inference of the neural network g_v, i.e., a full forward pass through the network given a feature vector as input, occurs at each initialization of a newly expanded node in the RC-RSBG planner. Since the total available search time is limited, the inference time restricts the number of achievable iterations. On the other hand, a higher inference time, i.e., a lower number of iterations, can be acceptable if the prior experience significantly improves exploration. This compromise must be appropriately considered when defining a neural network architecture for g_v. In the field of autonomous driving, classical network architectures used to predict behavior policies and action values in, e.g., reinforcement, experience, or imitation learning, can be decomposed into layers processing the input representation to generate an intermediate representation, and output layers generating the policy and value information. Input representations mostly consist of either fully connected layers [134, 144] or convolutional layers processing grid features [142, 143, 262] or separately convoluting individual agent features [171]. The output processing mostly consists of fully connected layers [134, 142–144] or head networks to separately predict

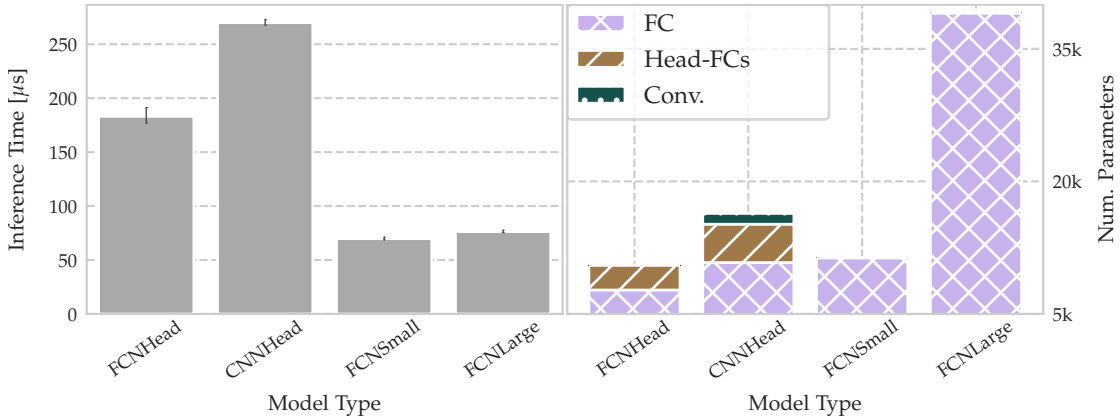


Figure 5.2.: Inference times and the number of network parameters for characteristic neural network architectures for experience-learning. For the application of experience learning with single batch evaluations on a CPU, a larger fully connected network ("FCNLarge") is superior to architectures with additional head networks (suffix "Head") and convolutional input processing (prefix "CNN") regarding inference time and number of available trainable parameters.

values and policies [171].

The suitability of different architectures for experience learning is evaluated in the following by comparing inference times and the number of trainable parameters for four types of neural network architectures*. They all represent the value network g_v, yet, each has a specific architectural characteristic. The input dimensions are set according to the feature definitions of the previous sections. The input space consists of 6 ego features and 4×4 other agent features. The network predicts three value functions, which yields for an action space of size 8, a total number of $3 \times 8 = 24$ linear network outputs. Specifically, the evaluation compares the architectures

- **FCNSmall**/**FCNLarge**: Fully connected network with 3×64 (Small) or 3×128 (Large) ReLU layers.
- **FCNHead**: Equal to **FCNLarge** except it adds three separate head networks with 2×16 fully connected ReLU layers to separately process action-value functions.
- **CNNHead**: Equal to the **FCNHead** with convolutional processing of other agents' input features using two convolutional layers of size 32. The architecture is related to [171].

The inference is performed using the C++ CPU-Pytorch-API v1.9.0 on an Intel 3.2 Ghz CPU with computations restricted to a single core. The inference times are averaged over 10000 network passes with uniformly drawn input data in the range 0 to 1. The results are depicted in Fig. 5.2.

The inference times of fully connected architectures are significantly lower. Adding separate head networks at least doubles the inference time. Adding convolutions requires at least three times the inference time given by the fully connected architectures. It seems that single batch evaluation of fully connected networks on CPUs† better benefits from hardware optimizations

*The architectures are based on the variants developed within the supervised thesis [263].

†GPU computation would require a multi-threaded implementation queuing newly explored states, which are then processed in batches. These implementations are reasonable in applications without strict computational limits, e.g., in Game playing [170]. Yet, at a low computational budget, batch processing may waste time in filling the queue without guiding exploration.

than more complex architectures though having to process a more significant number of parameters. The minor difference in inference time between FCNSmall to FCNLarge underlines the benefits of hardware optimization, given that FCNSmall has around one fourth of the parameters.

Overall, an increase in the number of available trainable parameters should, in theory, allow generalizing better assuming an optimal training process. Therefore, the low inference time of the FCNs and a large number of trainable parameters make these kinds of architectures superior for the application of learning experiences. The architecture FCNLarge is, therefore, chosen to represent the value network g_v.

5.4. Collecting Exploration-Distribution-Aligned Offline Experiences

This section presents an offline data collection process for supervised learning of experiences to achieve equal distributions over environment states during offline training and online exploration.

The principle of likelihood maximization in supervised learning relies on the training data $\mathbb{D}_{\text{train}}$ being similarly distributed as the validation data $\mathbb{D}_{\text{online}}$ confronted with during application [261]. This requirement is fulfilled in the context of experience learning by ensuring that

$$f_{\text{train}} \overset{!}{\approx} f_{\text{online}}. \tag{5.9}$$

The states in the training data should be similarly distributed as the states for which an experience is inferred during online planning. The exploration distribution f_{online} is a combination of the distribution over environment states $s^t \sim f_{\text{env}}$ visited during online planning and interacting with the environment, and the distribution of predicted search states $o' \sim f_{\text{explore}}(\cdot|o^t)$ which is conditioned on the current observation o^t.

Thereby, the environment state $s^t = (s_i^t, s_1^t, s_2^t \ldots, s_{N_{-i}}^t)$ consists of the agent states $s_j^t = (o_j^t, \mathcal{B}_j^t)$. As defined in Sec. 3.2.1, the observation states o_j^t comprise the observable dynamic properties of an agent. The behavior spaces $\mathcal{B}_j^t$ are estimated using belief tracking. As discussed in Sec. 5.3.2 belief information is not part of the neural network input representation. Instead, by varying $\mathcal{B}_j^t$ in the set of simulated states s_j^t in the generated dataset, the distribution over unknown behavior spaces is modeled and implicitly considered in the learned experiences.

A two step process is proposed to collect a data set $\mathbb{D}_{\text{online}}$ approximately distributed according to f_{online}:

- **Distribution Approximation:** Sec. 5.4.1 presents a sampling-based process to obtain a collection of fully observable states approximately distributed according to f_{online}.

- **Experience Planning:** Sec. 5.4.2 explains how a variant of the RC-RSBG planner exploits full observability of states to obtain accurate return value, envelope violation and collision risk estimates for each collected state.

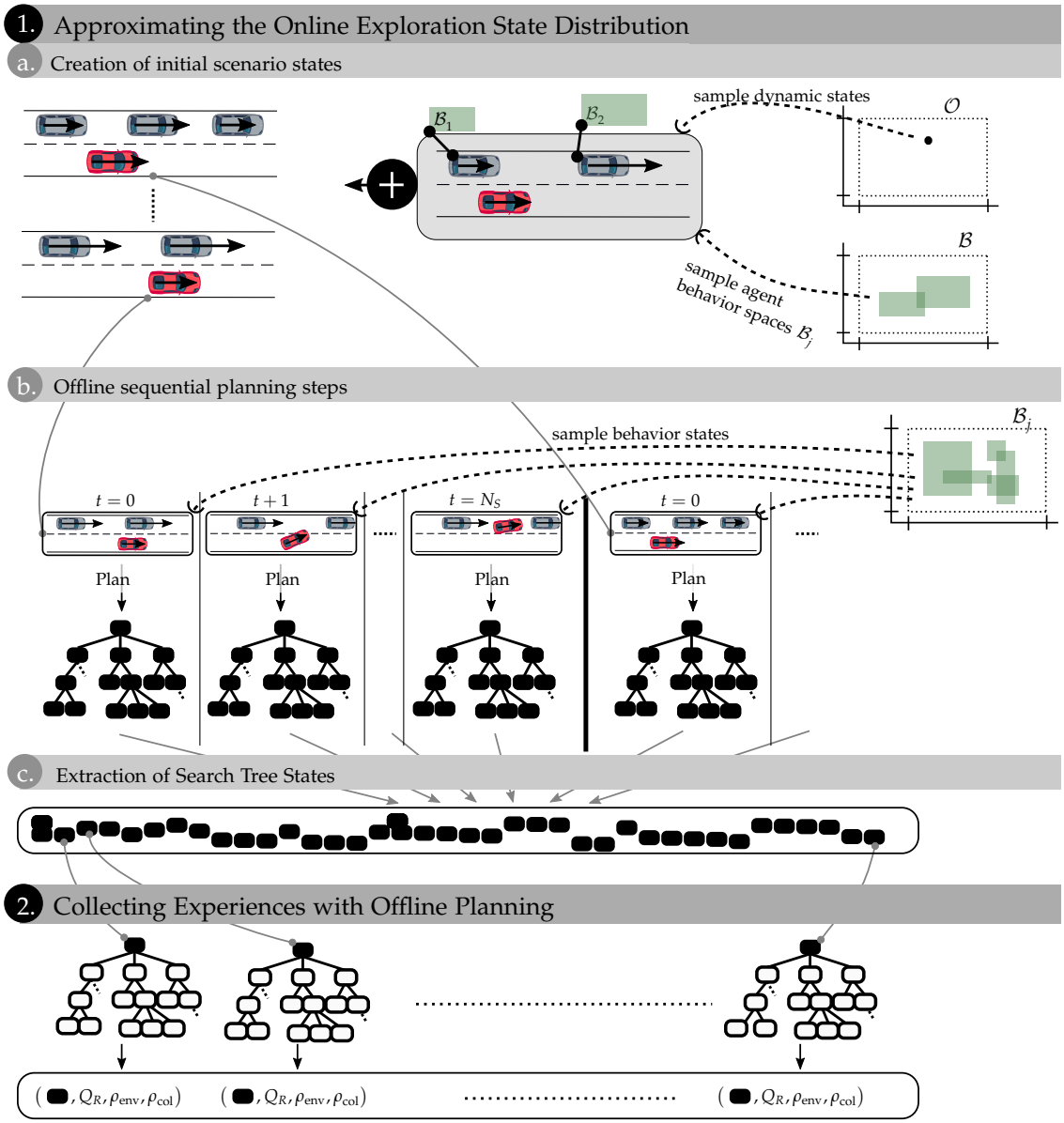


Figure 5.3.: Overview of the data generation process. First, to approximate the exploration state distribution sequential online planning steps are performed for a set of sampled scenarios. Secondly, the visited tree states are extracted and additional offline planning approximates the value experiences of these states. Behavior variations are represented in the data set by randomly sampling agent behavior spaces and behavior states.

The resulting data set is split using percentage p_{train} into a training $\mathbb{D}_{\text{train}}$ and test data set $\mathbb{D}_{\text{test}}$. The above steps are detailed in the following and visualized in Fig. 5.3.

5.4.1. Approximating the Online Exploration Distribution

Two sampling processes are interwoven to collect samples approximately distributed according to f_{online}. The first process samples scenarios from a scenario distribution. The second performs sampling-based online planning steps within these scenarios. The set of all expanded tree states

is then approximately distributed according to f_{online}.

Specifically, the distribution over visited states $s^t \sim f_{\text{env}}$ is approximated by sampling multiple initial scenario states $s_m^{t=0} \sim f_{\text{env}}(\cdot|t=0), m = \{1,\dots,M\}$ from a manually defined initial state distribution. Apart from sampling physical properties, e.g., initial positions and velocities of vehicles, random subsets of the full behavior space $\mathcal{B}$ are sampled as unknown behavior spaces of other agents $\mathcal{B}_j$ (cf. Sec. 3.3). After beginning from the initial scenario state, the ego vehicle repeatedly performs belief updates with subsequent RC-RSBG planning steps. The planner thereby applies a fixed number of search iterations $\mathsf{N}_{\text{iters}}^{\text{col}}$ in each planning step. Given the planned action, the simulation proceeds to the next environment state s_m^t until reaching the terminal criterion of the scenario, e.g., the goal or a maximum number of steps N_S. Uncertainty in the behavior of other traffic participants is simulated by letting them sample a behavior state from their behavior space in each time step and then choose an action according to the hypothetical policy. This generates around $M \cdot N_S$ visited state samples approximately distributed according to f_{env}. Note that each state s_m^t also includes information about behavioral variations in form of the agents' behavior spaces $\mathcal{B}_j^t$.

After a planning step in state s_m^t, the expanded states are retrieved from the MCTS search tree. The expanded and visited states of all scenarios form together with the behavior spaces of other agents given from s_m^t, a collection of fully observable states $\mathcal{S}^{col} = \{s_{t,l,m}\}$, $s_{t,l,m} = (o_{t,l,m}, \mathcal{B}_1, \dots, \mathcal{B}_{N_{-i}})$, $0 \le t \le N_S$, $l \in \{1,\dots,\mathsf{N}_{\text{iters}}^{\text{col}}\}$, $m \in \{1,\dots,M\}$. The tree state expanded in iteration l during planning for visited environment state s_m^t is denoted with $o_{t,l,m}$.

The combination of scenario sampling and planning can be considered as marginalizing the conditional dependence on the environment states out of the exploration distribution:

$$f_{\text{online}}(o) = \int_{\mathcal{O}} f_{\text{explore}}(o|o') f_{\text{env}}(o') \mathrm{d}o'. \tag{5.10}$$

With this simplified consideration, the collected set of states $\mathcal{S}^{col}$ can be seen as being approximately distributed according to f_{online}.

5.4.2. Collecting Experiences With Offline Planning

For each state $s_{t,l,m}$ in the set of collected states $\mathcal{S}^{col}$, value experiences are obtained by applying a second offline planning step. Given that the collected states $s_{t,l,m}$ include the true behavior spaces of other participants, a variant of the RC-RSBG planner is used predicting other participants using their true behaviors π_j. After planning for $\mathsf{N}_{\text{iters}}^{\text{est}}$ iterations, the return and violation risk values, $Q_R(\langle o_{t,l,m}\rangle, a_i), \rho_{\text{env}}(\langle o_{t,l,m}\rangle, a_i)$ and $\rho_{\text{col}}(\langle o_{t,l,m}\rangle, a_i)$ for all $a_i \in A_i$ are extracted at the root node. Together with the NN input features of the observable state $o_{t,l,m}$ within $s_{t,l,m}$ this defines an experience tuple according to Sec. 5.3.1.

As motivated in Sec. 5.3.2, the input representation of the NN used for experience prediction excludes belief information. Nevertheless, the generated experiences encode uncertainty about the behavior variations as outlined in the following. The set of collected states likely contains tuples s and s' with similar observations ($o \approx o' : o \in s, o' \in s'$), but differing behavior spaces of other participants ($\mathcal{B}_j \neq \mathcal{B}_j' : \mathcal{B}_j \in s, \mathcal{B}_j' \in s'$). Therefore, experiences are generated for

differing behavior of another participant at near equal observed states. In the training process, likelihood maximization results in learning the expected value over the variations in behavior. This approach serves as a meaningful approximation to an eventually more accurate but tedious training and inference of experiences integrating belief information.

There are various parameters to adjust in the data generation process. The parameters N_S and M affect the size of the dataset. Choosing a higher number of initial states M and a lower number of search iterations $\mathsf{N}_{\text{iters}}^{\text{col}}$ more correctly approximates the environment distribution. In contrast, choosing a larger number of iterations leads to a more accurate approximation of the exploration distribution. The number of iterations for calculating the experiences is in general larger than the number of iterations to collect states, $\mathsf{N}_{\text{iters}}^{\text{est}} > \mathsf{N}_{\text{iters}}^{\text{col}}$ since the parameter $\mathsf{N}_{\text{iters}}^{\text{est}}$ tunes the accuracy of the estimates.

5.5. Parallelized Implementation of Risk-Constrained Planning

Parallelized planning implementations use multiple threads or cores to perform the planning task concurrently, which is especially meaningful in sampling-based planning approaches such as MCTS. Parallelization of the RC-RSBG planner can be employed in addition to value-guided exploration to improve online planning capability further.

Several approaches exist to parallelize the implementations of MCTS. An overview is given in [159]. In root-parallelization, each thread runs a separate search on an individual tree. After searches have finished, the root statistics are averaged to extract the final plan. Leaf-parallelization applies separate threads to obtain a more accurate roll-out estimate. In tree-parallelization, a single search tree is shared among threads. Mutex-based locking mechanisms are required to avoid inconsistency in the node statistics due to parallel updates. There exist also tree-parallelization variants with a lock-free mechanism [264] which require specifically adapted implementations.

Overall, the scalability of parallel implementations to a large number of threads is limited as, e.g., analyzed for gameplay in [265]. In the case of tree- and leaf-parallelization, available search time is wasted when waiting for other threads to finish their roll-out task or to unlock mutexes. Root-parallelization avoids such computational overhead allowing to spend all available computational resources on actual planning. This efficiency is beneficial in applications of online planning for AVs.

However, root-parallelization suffers from degradation of the averaged plan, as explained in the following. It is not straightforward to tune the amount of exploration when applying root parallelization. When limiting the search time available to each thread, a sufficient search depth is only achieved by reducing exploration. Reduced exploration can lead to each thread finding only a locally optimal root policy. Averaging these policies' action returns to obtain the final policy does not necessarily yield a more optimal plan. Rather the resulting policy can become suboptimal. An example in the context of interactive planning for AVs is as follows. One MCTS primarily explored goal states, e.g., reaching the target lane. Its resulting policy thus represents a goal-driven plan. Another MCTS explored collision states. Its resulting policy thus encodes

passive driving. If the averaged policy passively tries to be goal-directed, it may be more unsafe than the individual policies.

Additional information in a multi-objective planning formulation can potentially overcome the degradation with root-parallelization. Therefore, it is promising to analyze how a root-parallelized variant of the RC-RSBG planner performs under limited search time. It uses N_{th} parallel threads each running an RC-RSBG planner. Each thread $\mathfrak{t}$ provides an estimate of return values, envelope violation risks and collision violation risks, $Q_R^{\mathfrak{t}}(\langle o^t \rangle, a_i), \rho_{\text{env}}^{\mathfrak{t}}(\langle o^t \rangle, a_i)$ and $\rho_{\text{col}}^{\mathfrak{t}}(\langle o^t \rangle, a_i)$ for all $a_i \in A_i$ extracted from the root node $\langle o_t \rangle$ after its search has finished. After calculating action-wise the mean over the root node estimates of each thread

$$\begin{aligned} \overline{Q}_R(\langle o^t \rangle, a_i) &\leftarrow 1/\mathsf{N}_{\text{th}} \sum_{\forall \mathfrak{t}} Q_R^{\mathfrak{t}}(\langle o^t \rangle, a_i) \\ \overline{\rho}_{\text{env}}(\langle o^t \rangle, a_i) &\leftarrow 1/\mathsf{N}_{\text{th}} \sum_{\forall \mathfrak{t}} \rho_{\text{env}}^{\mathfrak{t}}(\langle o^t \rangle, a_i) \\ \overline{\rho}_{\text{col}}(\langle o^t \rangle, a_i) &\leftarrow 1/\mathsf{N}_{\text{th}} \sum_{\forall \mathfrak{t}} \rho_{\text{col}}^{\mathfrak{t}}(\langle o^t \rangle, a_i) \end{aligned} \tag{5.11}$$

the final planned stochastic policy is obtained by applying Alg. 10 to the mean estimates $\overline{Q}_R$, $\overline{\rho}_{\text{env}}$ and $\overline{\rho}_{\text{col}}$.

6 Evaluation

This chapter contributes an extensive simulative evaluation of the Robust Stochastic Bayesian Game (RSBG) and Risk-Constrained Robust Stochastic Bayesian Game (RC-RSBG) interactive planners. It starts with a description of the experiment setup in Sec. 6.1. The setup is based on the OpenSource benchmarking framework BARK contributed alongside this thesis. The section presents scenarios, baseline planners, and a concept to systematically benchmark interactive planning under unknown microscopic behavior variations of other participants. Each of the following sections then evaluates contributions of one of Chapters 3, 4, and 5. First, the benefits of interactive and robustness-based planning in behavior spaces with the RSBG planner are analyzed in Sec. 6.2. Sec. 6.3 evaluates the interpretability of the risk formalism integrated into the RC-RSBG planner. Finally, the benefits of experience-based parallelized planning are investigated in Sec. 6.4. Parts of this chapter are based on previously published work in [37, 173, 266].

6.1. Experiment Setup

The experiment setup is based on the simulation framework *BARK*, an acronym for *B*ehavior Benchm*ARK*. BARK was developed in the context of this thesis as joint work with colleagues and is available as open-source software*. The next section highlights some of the core features of BARK as published in [266]. Sec. 6.1.2 presents the evaluation scenarios. The simulation of inaccurate microscopic behavior prediction is discussed in Sec. 6.1.3 followed by a description of the baseline interactive planners in Sec. 6.1.4.

6.1.1. Benchmarking Interactive Planning using BARK

BARK provides a software framework for the systematic evaluation and improvement of behavior models. BARK defines the behavior $\mathtt{B}_t^i$ of an agent i at time t as *its desired future* sequence of physical states encoding the agent's strategy to reach a short-term goal, e.g., changing lane. A behavior may deviate from the executed motion in the environment due to errors in trajectory

*https://github.com/bark-simulator/bark/

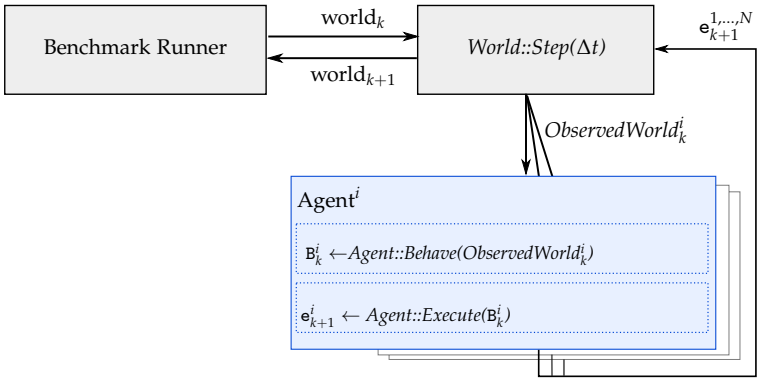


Figure 6.1.: BARK's simulation loop is handled by the benchmark runner holding the current world state at discrete world time k. In each iteration, the benchmark runner calls *World::Step(Δt)*. This function generates an *ObservedWorld* for each agent and passes it to the agent's internal *BehaviorModel* which generates a behavior B^i_k. The behavior is passed to the agent's *ExecutionModel* calculating the next executed agent state e^i_{k+1}. The next world state at time $t_{k+1} = t_k + \Delta t$ integrates the updated agent states for all agents N and is returned to the *Benchmark Runner* (graphic from [266], ©2020 IEEE).

tracking or environmental influences. The core idea of BARK is to apply the same behavior model implementation to

1. plan the ego-motion of the Autonomous Vehicle (AV),

2. predict the motion of other, potentially human-driven, vehicles,

3. forward simulate an agent in a purely virtual environment.

For example, a traffic model, such as the Intelligent Driver Model (IDM) [186] can, on the one hand, be used to populate a simulation with agents but also as a generative model to predict other agents' motion from the viewpoint of the ego vehicle. Experiments in BARK shall be fully reproducible, independent of the frequency at which the simulation runs. To ensure this, BARK models the world as a multi-agent system with agents performing *simultaneous movements* in the simulated world. At fixed, discrete world time-steps, each agent plans its behavior using an agent-specific behavior model, which only has access to the agent's observed world but not the simulator's simulated world. The concept of simultaneous movement ensures that a behavior model can plan based on reproducible input information. It avoids timing artifacts that may occur in message-passing, middleware-based simulation architectures. Fig. 6.1 visualizes the core concept of BARK's simulation model.

BARK uses behavior models not only for behavior planning but also for predicting other agents in the world. For example, the observed world of each planner derives from the actual world. All agents in this observed world behave according to their prediction configuration. Fig. 6.2 visualizes BARK's observed world model. The concept of an observed world allows configuring the planner's prediction differently from the actual environment to systematically examine planning failures caused by inaccurate internal predictions of other traffic participants.

The following sections take a more detailed look at the BARK components.

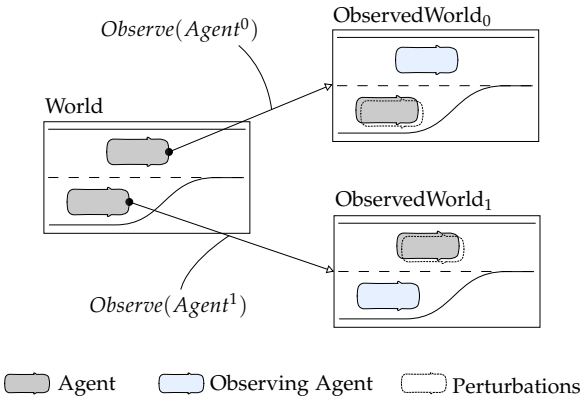


Figure 6.2.: Each agent in BARK uses an observed world for defining its behavior. In each observed world, there is one observing agent (depicted in blue) from whose perspective the observation is being made. Perturbations can be introduced by, e.g., exchanging the other agents' behavior models and model parameters in the observed world (graphic from [266], ©2020 IEEE).

World and ObservedWorld Model The BARK *World* model contains the map, all objects, and agents. Static and dynamic objects are represented in the form of object lists. The *ObservedWorld* model, on the other hand, reflects the world that an agent perceives. BARK's *ObservedWorld* model accounts for the fact that the observing agent has no access to the actual (world) behavior model of other agents. BARK can model different degrees of observability by either restricting access to the world behavior model or by only perturbing the parameters of the world behavior model.

Agent Models As shown in Fig. 6.1, an agent i in BARK provides two main interfaces:

- $\mathtt{B}_k^i \leftarrow$*Agent::Behave(ObservedWorld*$_k^i$*)*: calls the agent-specific behavior model, which generates a behavior trajectory $\mathtt{B}_k^i = (\mathtt{b}_1^i, \mathtt{b}_2^i, \ldots, \mathtt{b}_L^i)$, being a sequence of desired future physical agent states $\mathtt{b}_l^i = (t, x, y, \theta, v)$ between current simulation world time t_k and at least the end time of the simultaneous movement $\mathtt{b}_L^i(t) \geq t_k + \Delta t$. The time discretization of the behavior trajectory can be arbitrary.

- $\mathtt{e}_{k+1}^i \leftarrow$ *Agent::Execute(*$\mathtt{B}_k^i$*)*: calls the agent-specific execution model determining the agent's next state $\mathtt{e}_{k+1}^i$ in the world based on the generated behavior trajectory $\mathtt{B}_k^i$. An additional execution model allows to systematically examine the robustness of behavior planners against execution errors. This is not the focus of this thesis (cf. Sec. 3.2.4) and thus perfect execution is simulated using BARK's interpolation-based execution model.

To implement these two main interfaces, an agent holds the following additional agent-specific information:

- *GoalDefinition*: BARK considers agents to be goal-driven. It provides an abstract goal specification with various inherited agent goals, e.g., geometric goal regions or lane-based goals. Each agent contains a single goal definition instance.

- *RoadCorridor*: When an agent is initialized, it computes the set of roads and corresponding lanes required to reach its goal. The topology information on how roads are connected is extracted from the map. Also, the geometric information of the map, such as lane boundaries, is being discretized. This precomputation avoids computational overhead during simulation.

- *Polygon*: A 2D polygon defines the shape of the agent.

Scenario and Scenario Generation A BARK scenario contains a list of agents with their initial states, behavior, and execution models and a goal definition for each agent. Further, it contains a map file in the OpenDrive[†] format. Behavior benchmarking is supported by specifying for each scenario which agent is considered the 'controlled' agent during the simulation. A BARK scenario does not explicitly specify how agents will behave over time, e.g., by predefined maneuvers or trajectories. This concept allows simulating interactive scenarios in which other agents react to the controlled agent's behavior.

BARK provides a scenario generation module for configuring sets of scenarios. Different scenario sets can be constructed specifying the distribution of agent states, their behavior and execution models, and goal definitions.

Benchmarking For the systematic development of behavior models, BARK supports large-scale evaluation of behavior models over a collection of scenario sets contained in a *benchmarking database*. Serializing the database before starting benchmarks ensures that scenarios remain reproducible across different systems[‡].

BARK provides a *BenchmarkRunner* to evaluate specific behavior models with different parameter configurations over the entire benchmarking database. The evaluation is based on an abstract evaluator interface calculating a Boolean, integer, or real-valued metric based on the current simulation world state. Evaluators can be used not only for benchmarking but also internally by the behavior models, e.g., for the reward calculation in search or reinforcement learning-based planners.

The *BenchmarkRunner* runs each scenario of the database evaluating world states of the simulation. It terminates the scenario run based on criteria defined with respect to the evaluators. The quantitative results are dumped into a database allowing to query evaluation results for individual scenarios and parameter settings. BARK also provides a *distributed benchmark runner* to perform evaluations in parallel on multiple cores and clusters. Distributed processing is especially beneficial for computationally expensive planning algorithms such as the variants of Simultaneous-Move MCTS (SM-MCTS) presented in this thesis.

Software Design BARK has a monolithic, single-threaded core written in C++. The core of BARK is wrapped in Python, including pickling support for all C++ types. Scenario generation

[†] `http://www.opendrive.org/`

[‡] BARK supports adjusting random seeds in the scenario generation. However, differing implementations and versions of pseudorandom number generators across systems may yield differing scenarios using the same random seed.

and benchmarking, and service methods for parameter handling and visualization are implemented in Python. A simulation cycle is entirely deterministic, which enables the simulation and experiments to be reproducible and is essential to conduct systematic research in interactive planning.

6.1.2. Evaluation Scenarios

This thesis proposes an interpretable risk concept for interactive planning. The main application and motivation of the approach are to navigate efficiently in low speed ($30\,\mathrm{km\,h^{-1}}$ to $50\,\mathrm{km\,h^{-1}}$), low severity, dense traffic. The following two scenarios have complementary difficulties and thus serve well to analyze the interactive planners proposed in this thesis:

- **Freeway enter:** In a double merge, the ego vehicle wants to enter the freeway on the left occupied lane. Other vehicles drive only on the left lane. The merging scenario evaluates if the interactive planner appropriately coordinates both lateral and longitudinal actions simultaneously. The number of traffic participants to consider during planning is low, and it is mostly sufficient to consider two homotopies, merging behind or before another car. The scenario successfully terminates when the ego vehicle is close to and oriented along the centerline of the left lane with the ego velocity being larger than $18\,\mathrm{km\,h^{-1}}$. The allowed duration to successfully change the lane is $\overline{T}_{\mathrm{max}} = 6.0\,\mathrm{s}$.

- **Left turn:** The ego vehicle wants to turn left from a side into the main road and has to cross two occupied lanes. The left turn scenario evaluates if the interactive planner predicts a notable gap between vehicles coming from both sides. The amount of homotopies, i.e., gaps to take, is generally much larger than for freeway enter. More vehicles must be considered, which increases exploration complexity. However, the ego agent chooses from a reduced action set with only longitudinal accelerations in this scenario. The scenario successfully terminates when the ego vehicle passes the turning lane with the ego velocity being larger than $18\,\mathrm{km\,h^{-1}}$. The allowed duration to successfully perform the left turn is $\overline{T}_{\mathrm{max}} = 10.0\,\mathrm{s}$.

The simulation applies a step time of $\Delta t = \tau_{\mathrm{sc}} = 0.2\,\mathrm{s}$ for both scenario types. The scenario properties, e.g., initial distances between vehicles and velocities, are sampled uniformly. Adjusting the sampling ranges results in varying types of traffic densities and difficulties of the scenarios. The parameters are given in App. A.5. Introducing randomness in behavior into the scenarios is explained in the following section.

6.1.3. Benchmarking Effects of Inaccurate Microscopic Behavior Prediction

By using behavior models both for simulation and prediction, BARK allows to systematically analyze how inaccurate predictions affect the performance of the RSBG and RC-RSBG planners. Specifically, inaccurate predictions are modeled in simulation by calling the BARK simulation function World::Step(Δt) in the

1. **BenchmarkRunner** to progress to the next scenario state with $\Delta t = \tau_{\mathrm{sc}}$, and in the

2. **EnvironmentMove** function used in expansion and rollout steps (cf. Alg. 2 and Alg. 6) to predict the next state with $\Delta t = \tau_{\text{predict}}$. In this case, multiple forms of predictions are modeled by replacing the behavior models of observed agents before the call to WORLD::STEP(τ_{predict}):

   - **Macro-Actions:** The agent performs a macro action, e.g., lane changing or constant acceleration, selected from a set of predefined actions. The ego behavior is always of this form. For other agents, this behavior is used in the case of the *Cooperative* baseline defined in Sec. 6.1.4.
   - **Full Information:** The next agent state is predicted with the behavior model used in simulation as in case 1). This approach models prediction with full information about the behavior of other participants.
   - **Hypotheses-based:** The next agent state is predicted with a behavior hypothesis of the respective agent. This approach simulates interactive prediction using stochastic behavior hypotheses.

The replacement of behavior models does not affect the prediction's dynamic, geometric, or other properties. Thus, this approach allows to *systematically analyze* effects of inaccurate prediction of only the behavior of other participants.

Inaccurate prediction of intra- and inter-driver behavior is analyzed by determining a *single* behavior model for simulation, which also defines the hypothetical policy π^* of the behavior hypotheses. The simulation then uses the full range of model parameters to simulate variability in driving behavior. The hypotheses-based prediction, in turn, is defined only over a reduced range of parameters. Specifically, simulation and prediction use the IDM (cf. App. A.1). In the simulation, microscopic variations are simulated using two types of sampling processes applied 1) during the scenario generation and 2) during the simulation as follows:

- **Inter-driver variations:** Boundaries of behavior variations $[b^l_{j,\text{min}}, b^l_{j,\text{max}}]$, $l \in \{1, \dots, 5\}$ are sampled from uniform distributions differently for each agent and scenario ($\mathcal{B}_j \subseteq \mathcal{B}^*_{5D}$) from a 5-dimensional true behavior space $\mathcal{B}^*_{5D}$ defined over the IDM parameters. Thus, the range of allowed IDM parameters is different for all agents and scenarios. The parameters minimum and maximum boundary widths $\Delta_{\text{min}}/\Delta_{\text{max}}$ specify the minimum and maximum allowed widths of the sampled parameter ranges. This avoids unrealistic large intra-driver variations simulated according to the following concept.

- **Intra-driver variations:** In each simulation step, a true behavior state b is sampled uniformly from the boundaries of behavior variations $b \sim \mathcal{U}(\mathcal{B}_j)$. Thus, the true behavior state, i.e., the parameters of the IDM of other agents, constantly change during simulation. The ego agent can not observe these changing behavior states.

This sampling concept simulates the relationship of behavior spaces and behavior state given by the causal model of behavior spaces (cf. Sec. 3.3.2). The *prediction* of the RC-RSBG planner is then hypotheses-based. It uses the same behavior model as in simulation, i.e., the IDM to

	$\mathcal{B}^*_{5D}$		$\mathcal{B}_{1D,\text{Head.}}$	$\mathcal{B}_{1D,\text{Vel.}}$	$\mathcal{B}_{2D}$
Param b^l	$[b^l_{\min}, b^l_{\max}]$	$\Delta_{\min}/\Delta_{\max}$			
v_{desired} [m/s]	see App. A.5	0.5 / 1.0	9.5	[5.0, 15.0]	[5.0, 15.0]
T_{desired} [s]	[0.5, 2.0]	0.1 / 0.3	[0.0, 4.0]	1.25	[0.0, 4.0]
$s_{\min}$ [m]	see App. A.5	0.1 / 0.5	1.25	1.25	1.25
$\dot{v}_{\text{factor}}$ [m/s^2]	[1.5, 2.0]	0.1 / 0.3	1.75	1.75	1.75
$\dot{v}_{\text{comft}}$ [m/s^2]	[1.5, 2.0]	0.1 / 0.3	1.75	1.75	1.75

Table 6.1.: Boundaries of the simulated true behavior space $\mathcal{B}^*_{5D}$, the two 1-dimensional full behavior spaces $\mathcal{B}_{1D,\text{Head.}}$ and $\mathcal{B}_{1D,\text{Vel.}}$ defined over the single IDM parameter T_{desired} and v_{desired} respectively, and the 2-dimensional variant $\mathcal{B}_{2D}$ defined over both parameters.

predict other agents, yet, with the full behavior space $\mathcal{B}$ being defined only over a subset of the parameters of the simulated behavior space $\mathcal{B}^*_{5D}$.

Sec. 3.3.1 motivates the definition of behavior spaces over a subset of parameters of classical driving models, e.g., the IDM and focuses thereby on the desired time headway T_{desired}. Another interesting parameter in the IDM applicable for designing a behavior space is the desired velocity v_{desired}. The evaluation considers three types of full behavior spaces, two 1-dimensional variants $\mathcal{B}_{1D,\text{Head.}}$ and $\mathcal{B}_{1D,\text{Vel.}}$ defined over the single IDM parameter T_{desired} and v_{desired} respectively, and a 2-dimensional variant $\mathcal{B}_{2D}$ defined over both parameters. The definitions of the simulated behavior space and the 1D and 2D full behavior spaces are given in Tab. 6.1. Thereby, the remaining parameters of the IDM are set experimentally around the respective mean parameter range of the simulated behavior space. Preliminary experiments showed that the planning performance is not greatly affected when keeping these parameters within meaningful bounds.

6.1.4. Setup of RSBG, RC-RSBG and Baseline Planners

The evaluation compares the RSBG and RC-RSBG planner to several baseline interactive planners:

- **SBG** is similar to RSBG apart from applying *random* action selection among $A_j(\langle H_o \rangle, \theta'_j)$ in Alg. 3. It represents SM-MCTS approaches solving the Stochastic Bayesian Game (SBG) [184, 187] with hypotheses defined over behavior spaces.

- **MDP** does not incorporate belief information over hypotheses. Instead it uses a single hypothesis, $K = 1$, over the full behavior space, $\mathcal{B}^1 \equiv \mathcal{B}$, to predict other participants with random action selection in Alg. 3. Since it models stochastic transitions independent of prior states, this planner type is referred to as Markov Decision Process (MDP).

- **RMDP** uses a single hypothesis, $\mathcal{B}^1 \equiv \mathcal{B}$, similar to the MDP baseline, with *worst-case* action selection for other agents. Since it models stochastic transitions independent of prior states, yet, selects the transition function parameters, i.e., the behavior states, provoking worst-case outcome, it is referred to as Robust Markov Decision Process (RMDP) (cf. 3.4.3).

- **IntentSBG** uses two intent-based behavior hypothesis, $\theta^0 =$ "yield" and $\theta^1 =$ "no yield" ($K = 2$). These apply an IDM model. For "give way", the model is parameterized to take into account vehicles on other lanes as front vehicles. For "take way", it only considers

vehicles touching the own lane. Microscopic variations within the intents are defined using sampling from the full behavior space $\mathcal{B}_{1D,\text{Head.}}$ defined in Sec. 6.1.3. Belief tracking employs the product posterior.

- **Cooperative** selects actions for *ego and other agents* with UCT thereby applying a combined cooperative reward function. Details are given in the explanation of the reward parameterization below. This baseline shall model cooperative interactive planning using SM-MCTS as presented in [52, 53, 114].

- **FullInfo** variants, e.g., RC-RSBGFullInfo, are equal to SBG, RSBG, respectively RC-RSBG planners, but have access to the actual behavior policies π_j to apply these during prediction. These baselines estimate an upper bound on the performance in the ideal, unrealizable case of having full knowledge over the unknown behavior spaces $\mathcal{B}_j$ of other participants.

The following paragraphs give details on applied action spaces, reward settings, and other parameters employed in the evaluation.

Action Space

Planners RSBG, MDP, IntentSBG, RMDP, SBGFullInfo and RSBGFullInfo use UCT action selection for the ego agent (cf. Alg. 4), the cooperative approach additionally for the other agents. The planners RC-RSBG and RC-RSBGFullInfo apply risk-constrained planning with risk-constrained ego action selection (cf. Sec. 4.3).

All planners use an equal ego action space. For the freeway scenario, it consists of the macro actions lane changing, lane keeping at constant accelerations, $\dot{v}_i = \{-5, -2, 0, 2, 5\}\,[\text{m/s}^2]$ and gap keeping based on the IDM. For the left turn scenario, it consists of only the longitudinal accelerations $\dot{v}_i = \{-5, 1, 5\}\,[\text{m/s}^2]$. The cooperative approach employs the respective action space also to predict the other agents.

Envelope and Collision Indicators

The parameters of the envelope violation indicator (cf. Sec. 4.5) are set to $\text{acc}_{\text{max, ego}} = 5\,\text{m}\,\text{s}^{-2}$ and $\text{acc}_{\text{max, other}} = 5\,\text{m}\,\text{s}^{-2}$. To be able to draw parallels between the human violation risk (cf. Sec. 4.1.1) and the RC-RSBG planning results, the response times are set to $T_{\text{ego, react}} = 1\,\text{s}$ and $T_{\text{other, react}} = 1\,\text{s}$ which are common response times of human drivers.

The evaluations of the single-objective planners, e.g., the RSBG planner, in Sec. 6.2 apply an indicator $f_{\text{collision}}$, that indicates a collision when another vehicle overlaps with a static safety boundary of 0.5 m around the ego vehicle. By introducing a static safety boundary, the analysis follows approaches that introduce safety by negatively rewarding, not the collision, but the violation of a safety margin. This indicator is applied for all planners evaluated in Sec. 6.2. When $f_{\text{collision}}$ evaluates to true for a newly expanded tree state, i.e., the state violates the static safety margin, the state is assumed to be terminal.

In the evaluation of risk-constrained planning in Sec. 6.3, $f_{\text{collision}}$ indicates actual collisions. Switching from the detection of static safety margin violations to a detection of an actual collision

in the planning evaluations is important to analyze if the weak constraint on the collision risk (cf. Sec. 4.1.3) is sufficient. It shall serve to resolve ambiguities while constraining the envelope violation risk can target safety. This indicator is applied for all planners evaluated in Sec. 6.3 and 6.4.

Reward Setting

The RC-RSBG and RC-RSBGFullInfo planners apply the simplistic reward function

$$u_i(o,a) = 1.0 \cdot \textsc{GoalReached}(o') \tag{6.1}$$

which gives a positive reward when the ego agent transitions into a goal state o' when joint action a is executed by the agents in observation state o. The RC-RSBG planners also support more complex reward functions, e.g., based on deviations from the desired velocity. Though such reward functions implicitly influence the microscopic nature of the planned behavior, they can be helpful to improve exploration, which is impeded using a sparse goal-based reward setting. Nevertheless, the simplistic reward function is applied to avoid that results are influenced by the reward parameterization when evaluating risk-constrained planning.

The single-objective planners apply one of the following two reward functions:

- The reward function

$$u_i(o,a) = 0.1 \cdot \textsc{GoalReached}(o') - 1.0 \cdot f_{\text{collision}}(o') \tag{6.2}$$

  considers only collisions and goal reaching and is applied for the evaluations in Sec. 6.2.

- A reward function modeling risk-awareness is defined as

$$u_i(o,a) = 0.1 \cdot \textsc{GoalReached}(o') - 0.1 \cdot \frac{f_{\text{envelope}}(o') \cdot \tau_{\text{predict}}}{\beta \cdot \mathsf{T}_{\text{Plan}}} - 1.0 \cdot f_{\text{collision}}(o'). \tag{6.3}$$

  It is designed such that it fully erases the goal reward when the predicted envelope violation duration $\sum_{\forall t' \leq \mathsf{T}_{\text{Plan}}} f_{\text{envelope}}(\cdot)\tau_{\text{predict}}$ within the planning horizon T_{Plan}, exceeds the allowed violation $\beta \cdot \mathsf{T}_{\text{Plan}}$. It is applied in the evaluations of risk-constrained planning in Sec. 6.3 and Sec. 6.4.

The Cooperative planner applies a global reward function $u_{glob,j}(o,a)$ combining the agent's individual reward $u_j(o,a)$ with the other agents' rewards (including the ego agent), $m \neq j$, using cooperation factor c:

$$u_{glob,j}(o,a) = 1/N\big[(1-c) \cdot u_j(o,a) + c \cdot \sum_{\forall m \neq j} u_m(o,a)\big] \tag{6.4}$$

Thereby, $u_j(o,a)$ and $u_m(o,a)$ are calculated for the respective agent using one of the previous defined reward functions. The cooperation factor is set to $c = 0.1$.

Other Parameters

All planners use the fundamental prediction duration $\tau_a = 0.2\,\mathrm{s}$, a maximum search depth of $d_{\max} = 10$ which gives $\mathsf{T}_{\text{Plan}} = \sum_{d=1}^{d_{\max}-1} d \cdot \tau_a = 11\,\mathrm{s}$ due to the linearly increasing prediction duration $\tau_{\text{predict}} = d \cdot \tau_a$. The number of nearest agents considered during search, is set to $N_{-i} = 3$ for the freeway enter and $N_{-i} = 6$ for the left turn scenario. The parameters of the histogram approximation are $\Delta w_{\mathrm{b}} = 0.1\,\mathrm{m\,s^{-2}}$ and $N_{\text{samples}} = 10000$. The history length for sum posterior tracking is set to $L_{\mathrm{H}} = 20$. The IntentRSBG uses a shorter history of $L_{\mathrm{H}} = 5$ to reduce sensitivity of the product posterior to small likelihood values as discussed in Sec. 6.2.3. The exploration constant is $\kappa = 1.4$ in case of UCT action selection, and $\kappa = 10.0$ in case of the RC-RSBG planner. The filter factor is set to $\upsilon = 3.5$.

6.1.5. Benchmarking Metrics

The evaluation applies the following metrics to compare the performance of the planners.

- P_{suc} [%]: The percentage of scenario trials the ego vehicle reached the scenario goal within the allowed simulation time.
- P_{col} [%]: The percentage of scenario trials the ego vehicle collided, i.e, the shape of the vehicles overlap.
- $P_{\text{stat. safe}}$ [%]: The percentage of driven time the indicator $f_{\text{collision}}$ indicated a collision in a scenario averaged over all trial scenarios. Note that, in Sec. 6.2, $P_{\text{stat. safe}} \neq P_{\text{col}}$ due to the definition of $f_{\text{collision}}$ using a static safe distance boundary.
- $\overline{T}_{\text{suc}}$ [s]: The average time to reach the goal for successful trials.
- β^*, [1]: The observed envelope violation risk is the percentage of simulation time the envelope is violated, $\beta^* = \frac{1}{N_\int} \sum_{\forall \int} \frac{\sum_{o^t \in \int} f_{\text{envelope}}(o^t) \cdot \tau_a}{L(\int) \cdot \tau_a}$ with $o^t \in \int$ giving the simulated states, $L(\int)$ the executed length of the scenario $\int$, and $N_\int$ the total number of scenarios.
- t_{w} [s]: The expected waiting time, $t_{\mathrm{w}} = \sum_{k=0}^{\infty} (\overline{T}_{\max} k + \overline{T}_{\text{suc}}) \cdot (P_{\text{suc}} P_{\max}^k)$ defines the expected time to solve a scenario. The calculation assumes that the ego vehicle encounters solvable scenarios with probability P_{suc} and duration $\overline{T}_{\text{suc}}$ and unsolvable scenarios with probability $P_{\max} = 1 - P_{\text{suc}} - P_{\text{col}}$ with duration equal to the allowed simulation time $\overline{T}_{\max}$.

The metric for a specific evaluation setting is calculated over a fixed set of 200 sampled scenarios. The collection of scenarios is sampled once and remains equal for a specific evaluation. Though the benchmarking collects all of the above metrics, each evaluation depicts only the metrics that reveal the most distinct performance differences.

6.2. Evaluating Behavior-Space- and Robustness-Based Planning

This section evaluates the concepts proposed in Chapter 3. It starts with analyzing the performance of the RSBG planner for different parameters of the hypotheses design process in

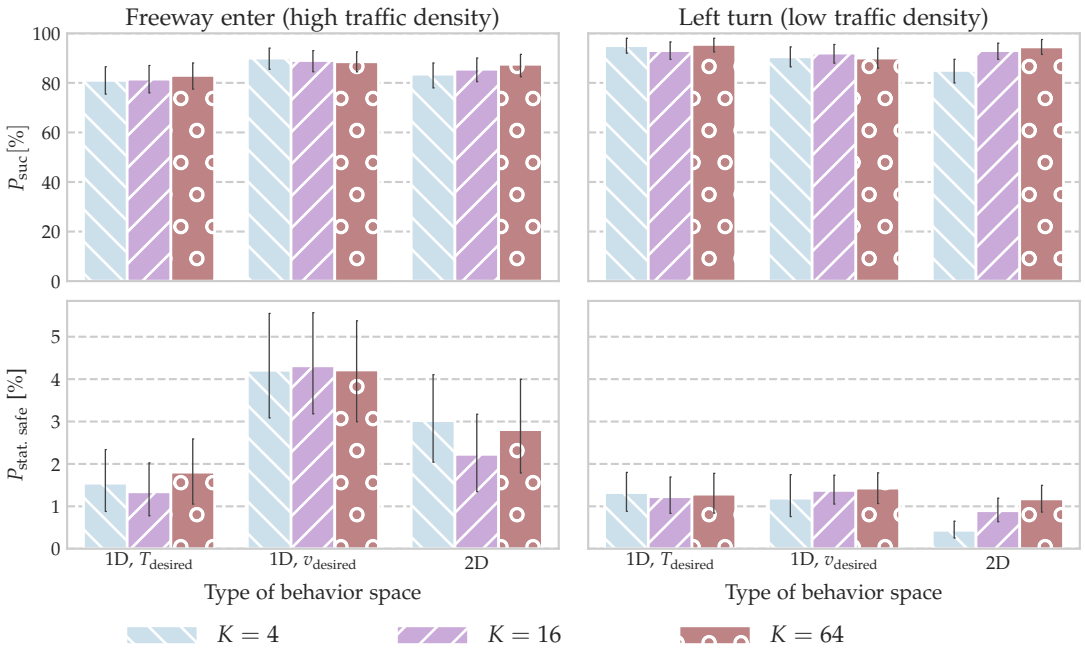
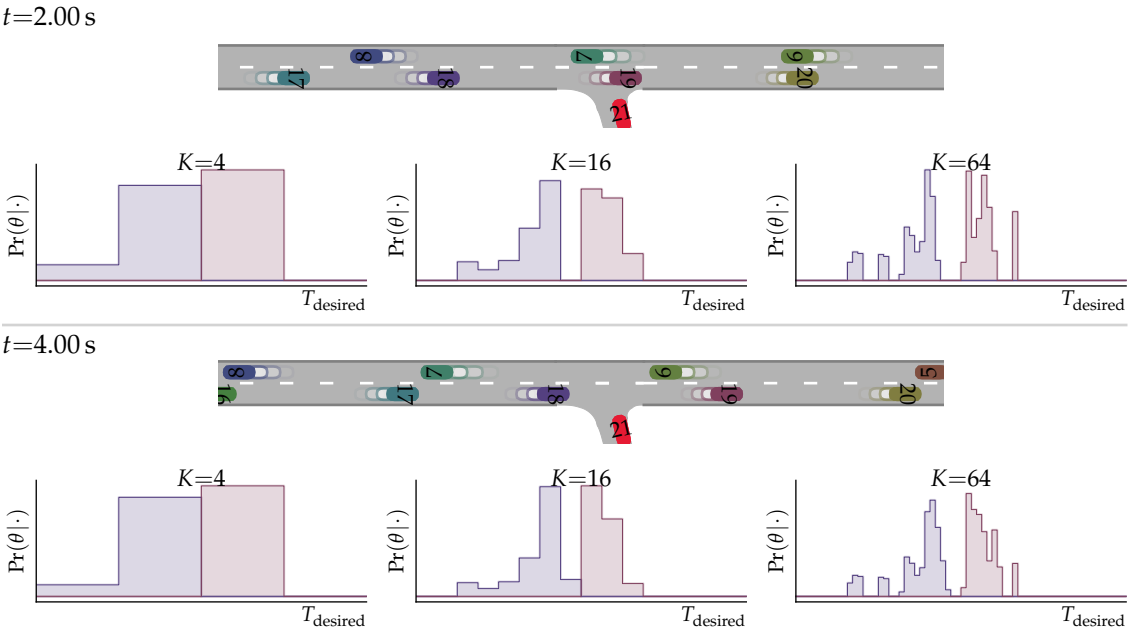


Figure 6.3.: Percentage of successful trials P_{suc} and static safe distance violations $P_{\text{stat. safe}}$ of the RSBG planner for different types of behavior spaces and number of hypotheses K. Safe distance violations are reduced when the behavior space is designed over the relevant behavior parameters of a scenario. The parameter T_{desired} dominates at higher traffic density (freeway enter) whereas at lower traffic density (left turn) v_{desired} and T_{desired} perform equally.

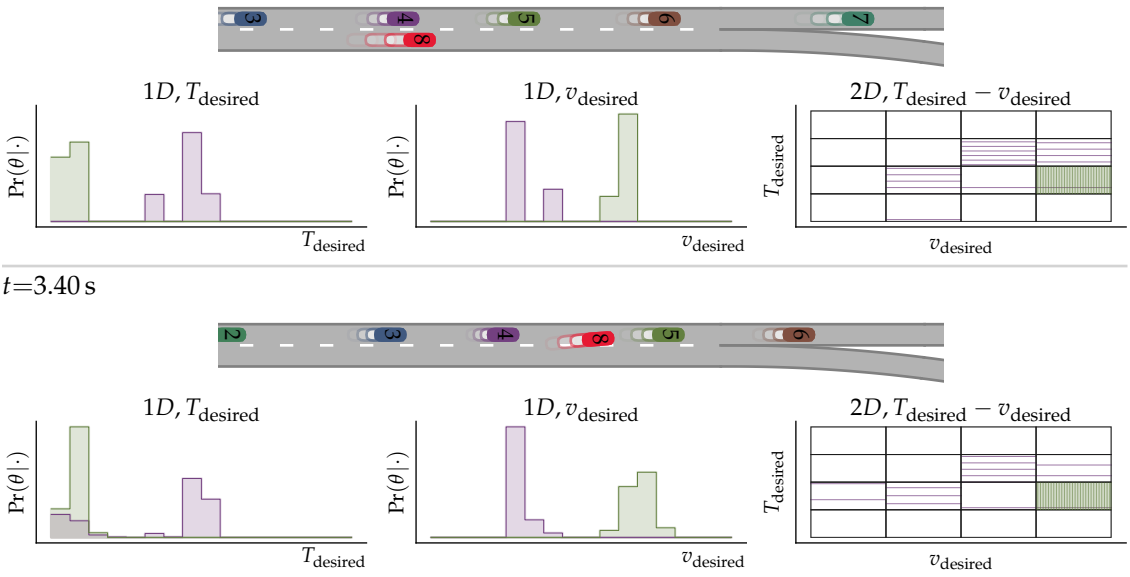
Sec. 6.2.1. Afterwards, Sec. 6.2.2 evaluates the effectiveness of robustness-based interactive planning. Finally, Sec. 6.2.3 compares the RSBG and the IntentRSBG planner.

6.2.1. Comparing Hypotheses Design Parameters

The quality of the behavior hypotheses designed according to Sec. 3.3.3 depends on the definition of the full behavior space $\mathcal{B}$ and the numbers of hypotheses K. Thus, the performance of the RSBG planner is compared for the two scenario types, the behavior spaces defined in Sec. 6.1.3 and varying $K \in \{4, 16, 64\}$ for a fixed number of 10000 search iterations.



(a) Variation of hypotheses number K.



(b) Variation of behavior space type.

Figure 6.4.: Tracked posterior beliefs over time for variations of behavior space dimension, type and number of hypothesis in the freeway enter and left turn scenario. A subset of tracked beliefs is shown with matching belief and vehicle colors. Increased hatch density depicts higher beliefs for the 2D behavior space.

For these different configurations, Fig. 6.3 depicts the percentage of successes and static safe distance violations. Concerning the success rates, there are no clear performance differences in both scenarios for different configurations of the hypotheses set, i.e., varying the behavior space type and the number of hypotheses. Yet, the percentage of static safe distance violations is nearly doubled for $\mathcal{B}_{1D,\text{Vel.}}$ compared to $\mathcal{B}_{1D,\text{Head.}}$ in freeway enter. It slightly increases for $\mathcal{B}_{2D}$

for $K = 4$ and $K = 64$. The left turn scenario does not show these significant differences in static safe distance violations.

Next, a qualitative analysis of tracked posterior beliefs over time in Fig. 6.4 shall give a deeper understanding of the cause of the quantitative results. Fig. 6.4a compares beliefs for different number of hypotheses in the left turn scenario. Regardless of the number of hypotheses K used to partition the behavior space, the highest tracked posterior beliefs cover equal parts of the behavior space.

The results support the motivation from Sec. 3.3.4 and show that the sum posterior is helpful to capture the unknown behavior spaces of other agents by using an *Or* combination of probabilities. The belief distribution is independent of the number of hypotheses, which may cause that the success and violation percentages of the RSBG planner only marginally depend on K.

The differences in beliefs for different behavior space types are analyzed for the freeway enter scenario in Fig. 6.4b. The freeway enter scenario has higher traffic density and smaller distance between vehicles compared to the left turn scenario. Higher traffic density lets the parameter T_{desired} dominate in the output of the IDM. At time $t = 0.8\,\text{s}$ shortly before the ego vehicle enters the left lane the posterior beliefs for $\mathcal{B}_{1D,\text{Head.}}$ and $\mathcal{B}_{1D,\text{Vel.}}$ of the green and violet vehicles are concentrated on small parts of the behavior space. At the time $t = 3.4\,\text{s}$ the ego vehicle has entered the left lane and started to interact with the other vehicles. The beliefs of the violet vehicle with $\mathcal{B}_{1D,\text{Head.}}$ are distributed towards lower values of T_{desired}. In contrast, the beliefs over v_{desired} remain concentrated at small parts of the behavior space. Using behavior space $\mathcal{B}_{1D,\text{Head.}}$, therefore, allows in this situation to better extract subtle information about the microscopic behavior variations than using $\mathcal{B}_{1D,\text{Vel.}}$. Since errors in microscopic prediction can cause higher safe distance violations, this potentially explains the higher static safe distance violations obtained with $\mathcal{B}_{1D,\text{Vel.}}$. The beliefs for $\mathcal{B}_{2D}$ seem to qualitatively reflect the 1D beliefs over $\mathcal{B}_{1D,\text{Vel.}}$ and $\mathcal{B}_{1D,\text{Head.}}$ along the specific dimensions at time $t = 0.8\,\text{s}$. A diversification of the beliefs at time $t = 3.4\,\text{s}$ as given with $\mathcal{B}_{1D,\text{Head.}}$ is not observed. Possibly, the additional uncertainty over v_{desired} impedes to capture meaningful beliefs over T_{desired}. At first glance, it seems that multidimensional behavior spaces better represent microscopic behavior variations due to modeling a larger set of unknown parameters. However, when keeping the number of hypotheses constant, a lower resolution results in each behavior space dimension for multidimensional behavior spaces. The reduction of resolution outweighs the advantages of multidimensional behavior spaces in addition to certain parameters negatively impacting information capturing.

Overall, the evaluation reveals the benefits of the sum posterior for belief tracking in behavior spaces and underlines the usefulness of the parameter T_{desired} to capture relevant behavior variations in dense interactive traffic situations supporting the motivation of behavior spaces in Sec. 3.3.1. The following evaluations choose the 1D behavior space $\mathcal{B}_{1D,\text{Head.}}$ with $K = 16$.

6.2.2. Comparing Robustness- and Non-Robustness-Based Exploration

The RSBG introduced in Sec. 3.4 integrates robustness-based optimality to reduce sample-complexity when predicting continuous behavior variations in interactive planning. This section evaluates how this criterion improves planning performance in dense interactive traffic.

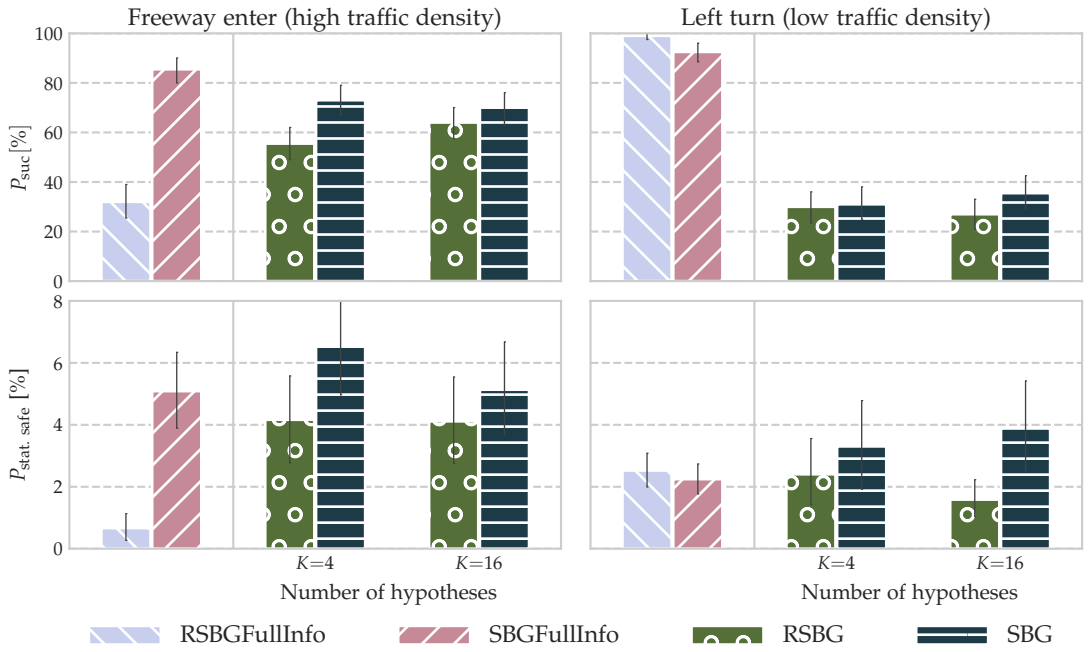
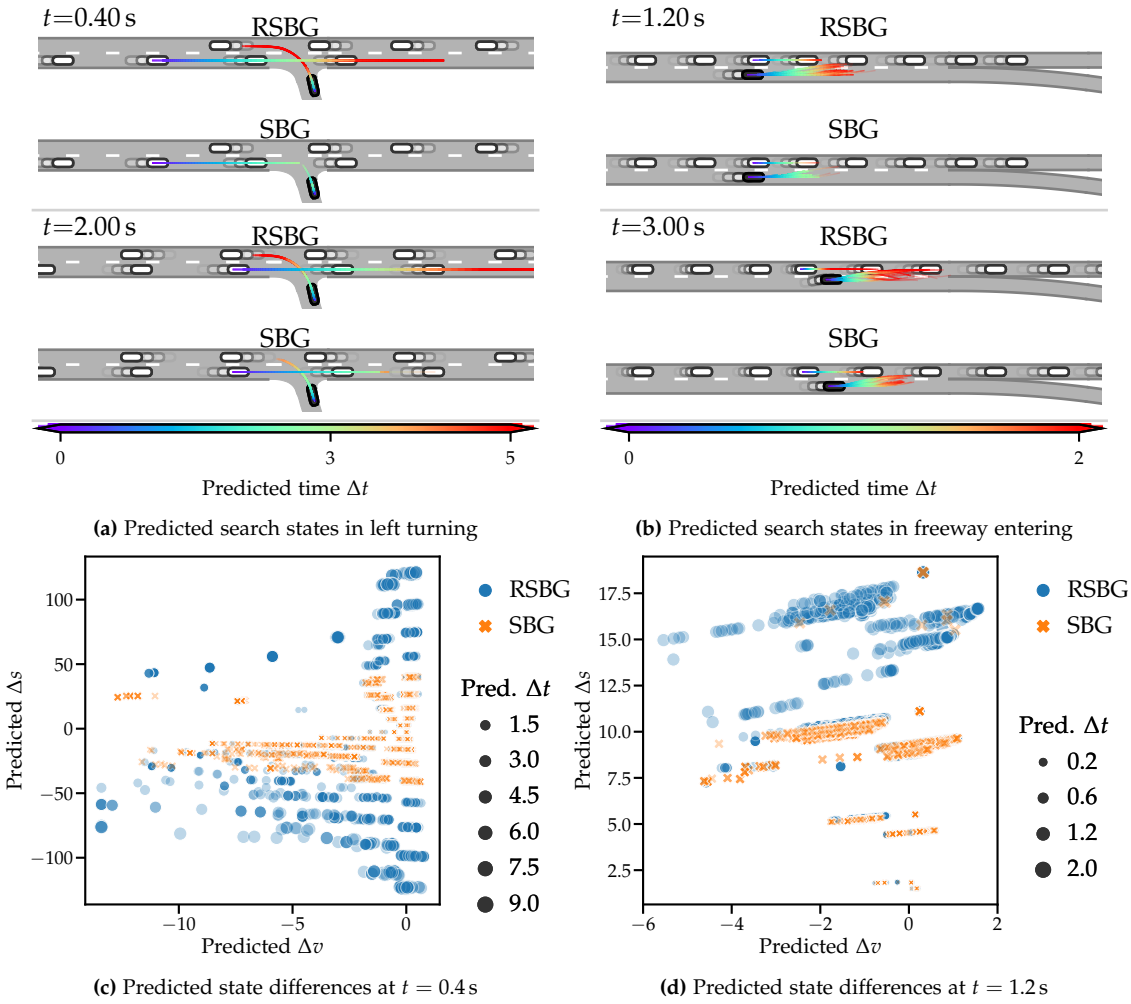


Figure 6.5.: Comparison of robustness- and non-robustness-based planning. Robustness-based planners (RSBG and RSBGFullInfo) sample-efficiently detect worst-case outcomes with respect to the ego reward function. Detection of worst-case outcomes is more relevant at higher traffic density reducing the percentage of static safe distance violations in the freeway enter scenario whereas at lower traffic density, differences to non-robustness-based planners (SBG and SBGFullInfo) are reduced.

First, a quantitative analysis compares the RSBG planner with its non-robustness-based variant, the SBG planner. The evalation also includes the baselines RSBGFullInfo and SBGFullInfo to evaluate the benefits of robustness-based planning when predictions are fully accurate. To investigate sample efficiency the number of search iterations is reduced to 1000. Since the sample complexity depends on the number of hypotheses K (cf. Sec. 3.4.1 and 3.4.4), the evaluation is performed for $K = 4$ and $K = 16$.

Fig. 6.5 depicts the percentage of successes and static safe distance violations. In freeway enter, the RSBGFullInfo planner achieves 5%, the RSBG planner around 2% for $K = 4$ and 1% for $K = 16$ less static safe distance violations than the non-robustness-based planners. In left turn, the FullInfo planners achieve near similar performance. The RSBG also provokes less static safe distance violations than the SBG planner. These differences indicate that the robustness criterion helps to detect worst-case outcomes sample-efficiently at a lower iteration number. In freeway enter for the SBG planner and in left turn for the RSBG planner the static safe distance violations reduce with increasing K which may be caused due to the dependence of the sample complexities onto K. Both $\mathcal{O}_{\text{RSBG}}$ and $\mathcal{O}_{\text{SBG}}$ exponentially decrease with K with a larger decrease of $\mathcal{O}_{\text{SBG}}$ for a fixed number of other agents (cf. Fig. 3.9). However, there is no direct connection between sample complexity and safe distance violations. The differences in the belief representation and the exploration characteristic for different hypotheses also affect the overall planning performance. The left turn scenario has a lower traffic density. Detecting microscopic variations that lead to a worst-case, unsafe outcome plays a minor role, and a different factor



(a) Predicted search states in left turning

(b) Predicted search states in freeway entering

(c) Predicted state differences at $t = 0.4\,\mathrm{s}$

(d) Predicted state differences at $t = 1.2\,\mathrm{s}$

Figure 6.6.: Comparison of exploration depths of robustness-based (RSBG) and non-robustness-based (SBG) planning. Top: Predicted search states of the ego vehicle and selected other agents for RSBG and SBG planners in specific simulation states. Bottom: Differences between predicted states and current state for other participants considered in the search at a specific time step for c) left turning and d) freeway entering.

contributes to the planning performance. Worst-case action selection concerning the ego-reward also negatively values states avoiding the scenario goal. This explorative behavior results in a more goal-directed exploration in the left turn scenario, which increases the success rate of the RSBGFullInfo planner. However, it provokes more static safe distance violations than in freeway enter.

Next, a qualitative analysis provides further insights into why robustness-based planning helps predict worst-case outcomes at lower iterations. Fig. 6.6 compares the predicted states within the search trees of RSBG and SBG planners. The RSBG planner achieves a larger search depth. It shows larger predicted times Δt (cf. Fig. 6.6a and 6.6b) and predicted longitudinal velocities Δv and distances Δs for other vehicles (cf. Fig. 6.6c and 6.6d). Robustness-based planning allows for deeper exploration since agents are more likely to repeatedly select the same action during subsequent search iterations when it is a worst-case action instead of randomly sampling from

previously expanded actions. These exploration differences qualitatively explain the benefits of robustness-based planning observed in the previous quantitative analysis.

The analysis reveals that the robustness criterion sample efficiently detects worst-case outcomes, especially in denser traffic. However, it also suffers from the fundamental problem of single-objective optimality criteria: It cannot differentiate between the worst-case outcome "collision" and "not reaching the goal". These results underline the need for a multi-objective optimality criterion separately dealing with efficiency and safety. Robustness-based planning can then help to sample-efficiently plan given such a criterion when considering worst-case outcomes only for the violation risk (cf. Sec. 4.3.3).

6.2.3. Comparing Behavior-Space and Intent-Based Prediction

Another motivation to apply behavior spaces for prediction is to circumvent the definition of intents (cf. Sec. 3.3.2). The following evaluation shall compare behavior-space-based and intent-based prediction in interactive planning. For this, the performance of the RSBG planner is compared to the IntentRSBG planner for a fixed number of 10000 search iterations. Two types of simulated reactions of other participants are evaluated which may have different effect on the planning performances: 1) Other participants do not show any yielding behavior, i.e., intents are not simulated for other participants, 2) Other participants show yielding intents with variations of intents over time. For the latter case, the IDM is extended to simulate the intents $i_j^t \in \{\text{"give way"}, \text{"take way"}\}$. For the intent "give way", vehicles on other lanes, for the intent "take way" only vehicles overlapping with the driving corridor are considered as front vehicles. To simulate variations of intents over time, intents remain fixed for a certain time duration $\Delta t_{\text{give way}}$ and $\Delta t_{\text{take way}}$ before switching to the respective other intent. The durations are sampled from two uniform distributions $\Delta t_{\text{give way}} \sim \mathcal{U}(2\,\text{s}, 5\,\text{s})$ for freeway enter and $\Delta t_{\text{give way}} \sim \mathcal{U}(5\,\text{s}, 6\,\text{s})$ for left turn and $\Delta t_{\text{take way}} \sim \mathcal{U}(1\,\text{s}, 2\,\text{s})$ for both scenarios.

Fig. 6.7 depicts the percentages of successes and collisions. The RSBG outperforms the IntentRSBG planner, both in the case "with intent simulation" and "w/o intent simulation" and in both scenario types. In the case "w/o intent simulation", it reaches a higher number of successful trials and provokes equal or fewer collisions. In the case "with intent simulation", IntentRSBG can benefit from better prediction and increases its success percentage while, however, provoking a larger amount of collisions. In contrast, the success percentage is reduced for the RSBG planner in left turn, while no collisions occur. The decrease of successful trials in the case "with intent simulation" is due to other vehicles partly blocking the intersection when "yielding" is active, which is a limitation of the IDM in intersection scenarios. Overall, these differences reveal that a purely intent-based prediction cannot reliably cope with situations where other participants do not clearly show a certain intent. Even if other drivers act according to a simulated intent model, accurately predicting microscopic variations, as given by behavior-space-based prediction, is of higher importance than correctly detecting and predicting intents.

Next, a qualitative analysis provides further insights. Fig. 6.8 compares how RSBG and IntentRSBG solve a freeway entering scenario with simulated intents. In Fig. 6.8a, the beliefs

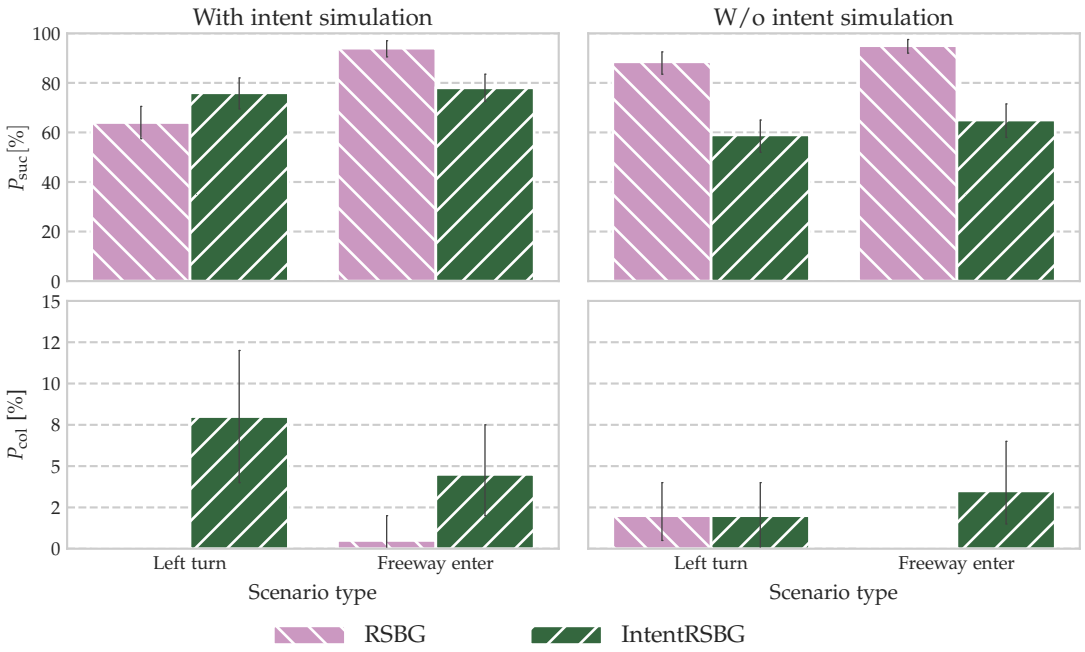
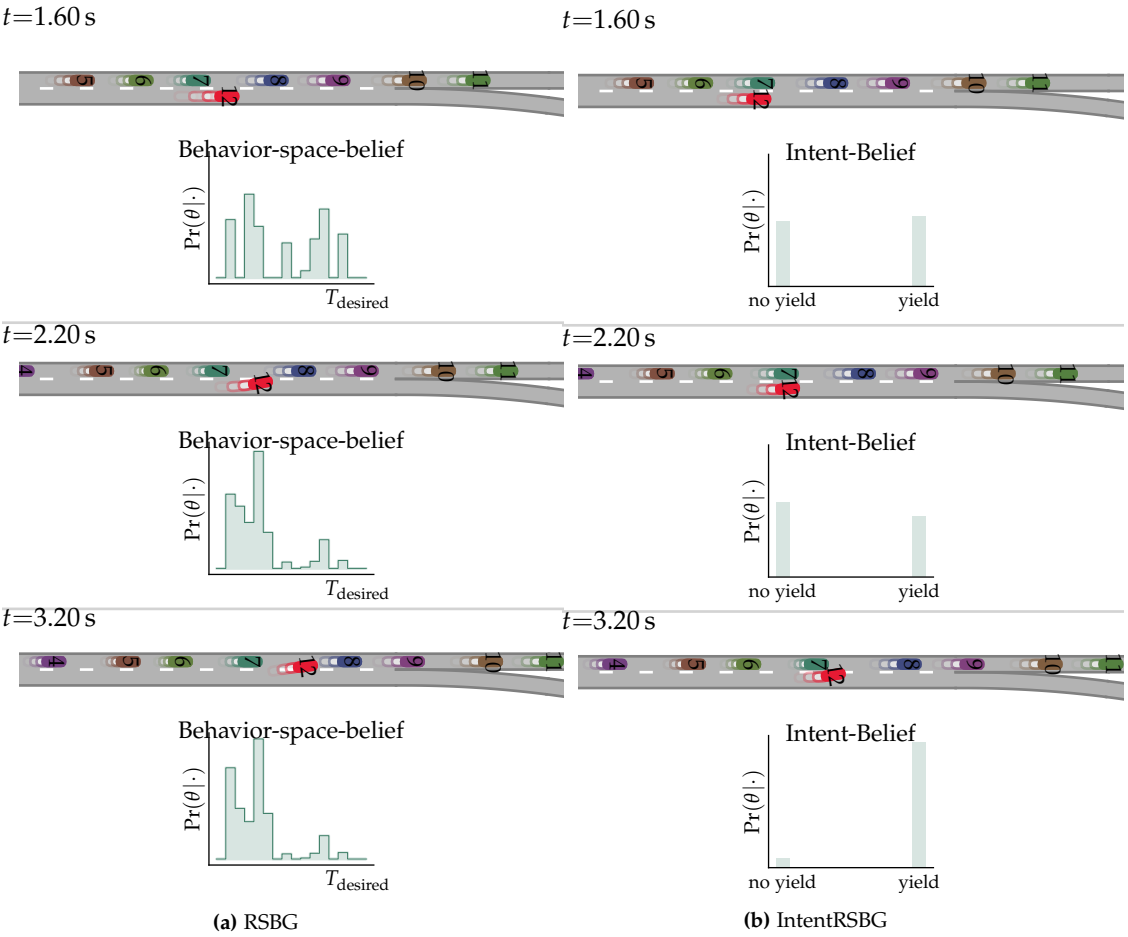


Figure 6.7.: Comparison of the performances of the RSBG and IntentRSBG planners evaluated in freeway enter and left turn scenarios with and w/o simulation intents. Behavior-space-based (RSBG) outperforms intent-based (IntentRSBG) planning since it enables prediction of microscopic behavior variations being crucial to solve dense interactive scenarios.

of the RSBG planner for vehicle seven start to shift towards lower values of T_{desired} when the ego vehicle becomes the front vehicle of vehicle seven between scenario times $t = 1.6\,\text{s}$ and $t = 2.2\,\text{s}$, reflecting how vehicle seven interacts on a microscopic level with the ego vehicle. This information facilitates completing the scenario at time $t = 3.2\,\text{s}$. In contrast, the IntentRSBG planner models the microscopic behavior of an intent by sampling from the full behavior space $\mathcal{B}_{1D,\text{Head.}}$ and is thus not able to adapt quickly to a change in the traffic situation. It waits until the ambiguity in intents is resolved at time $t = 3.2\,\text{s}$ to start merging. However, its deficiency in detecting microscopic variations almost provokes a collision.

Another drawback of intent-based prediction observed during preliminary experiments is its lack of robustness against inaccurate likelihood functions. The product posterior used to track beliefs over intents is sensitive to errors in the definition of the stochastic behavior hypotheses. A low probability of a specific intent within the histories of probabilities maintained for each intent hypothesis multiplicatively influences the belief at later time steps. Under certain circumstances, the beliefs of both intents become zero. Such a loss of belief information does not occur with the sum posterior.

Overall, the evaluation shows that hypotheses defined over behavior spaces in combination with the sum posterior better predict the microscopic behavior variations in dense interactive traffic than purely intent-based models. The concept avoids the definition of intent hypotheses and still succeeds in scenarios where intents change over time, which is an example of a shift of the unknown behavior spaces of other agents, $\mathcal{B}_j^t \neq \mathcal{B}_j^{t'}$ (cf. Sec. 3.3.2 and Sec. 3.3.4).



(a) RSBG **(b)** IntentRSBG

Figure 6.8.: Comparison of scenarios solved with RSBG and IntentRSBG planners. The beliefs over microscopic behavior variations given with behavior-space-based prediction allow the RSBG planner to more quickly change the lane. Having available only binary beliefs over intents impedes to detect fine-grained motions of other participants leading to a delayed and unsafe lane change with the IntentRSBG planner.

6.2.4. Comparing the RSBG Planner against Non-Belief-Based Baselines

This section compares the RSBG planner using belief-based prediction against the baseline planning algorithms presented in Sec. 6.1.4 which do not take into account belief information. The planners are evaluated in two variants, one using a limited planning time of 1 s and the other a fixed number of 20k search iterations.

The percentages of successes and collisions are depicted in Fig. 6.9. The RSBG planner outperforms the baseline planners in freeway enter. It achieves a higher success rate for both analyzed variants than the MDP and RMDP planner while provoking equal or fewer collisions. The Cooperative planner has a lower success percentage. In the left turn scenario, the MDP planner achieves the highest success percentage. However, for 1 s planning time, it collides frequently, whereas the RSBG planner avoids collisions at an only marginally decreased success percentage. When fixing the number of iterations, the planners can explore the problem space further, which increases success rates for RSBG, RMDP and MDP planners. The Cooperative planner has a drastic drop in performance compared to freeway enter and shows a large collision percent-

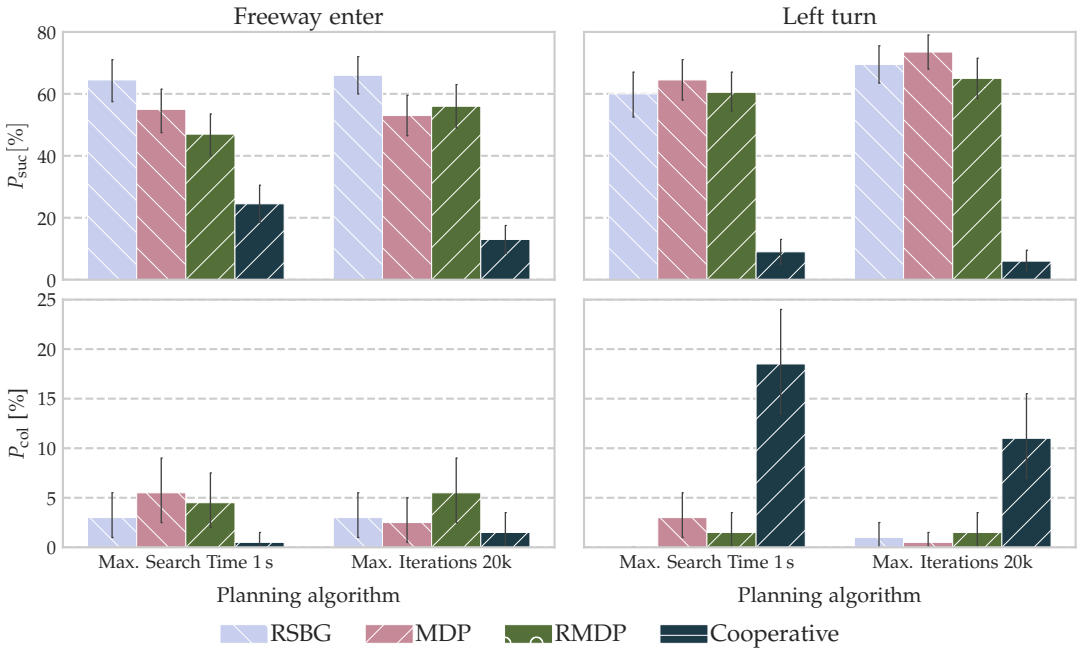


Figure 6.9.: Comparison of non-belief-based baseline and RSBG interactive planners. Incorporating belief information (RSBG) improves performance in dense interactive scenarios.

age. Examples of how scenarios evolve are given for all planners and the two scenario types in App. A.6.

The differences in performance can be explained as follows. Freeway entering, due to higher traffic density, requires a more fine-grained prediction of microscopic behavior variations, yet, compared to left turning, selecting the best homotopic variant is manageable. In contrast, in left turn, it is more relevant to accurately predict the potential turning gap, i.e., determine the best homotopy. The MDP planner achieves an average performance in both scenarios. Not adapting to the behavior of others and randomly predicting over the entire behavior space fosters passive behavior (freeway enter) or requires a large number of search iterations to prevent collisions (left turn, max. iterations). The worst-case prediction of RMDP over the entire behavior space leads to passive behavior and inaccurate microscopic predictions causing collisions (freeway enter). The Cooperative planner in this thesis uses the same action space for ego and other agents. Suppose the action space for other agents does not accurately cover the actual behavior, in this case, simulated with the IDM. In that case, the Cooperative assumption does not hold provoking collisions (left turn). Incorporating belief information increases the prediction accuracy and, by that, improves the detection of merging gaps (freeway enter). Further, it helps to reduce collisions at a lower number of available search iterations (left turn). The analysis shows that belief-based planning is an essential ingredient to plan interactively in dense traffic.

The analyses in this and previous sections show remaining collisions for all analyzed planners, even though the planners already consider violating a static safety margin as the collision event. The safety margin and reward specification do not map in an interpretable manner to the collision statistic. These findings underline the need for the proposed interpretable risk

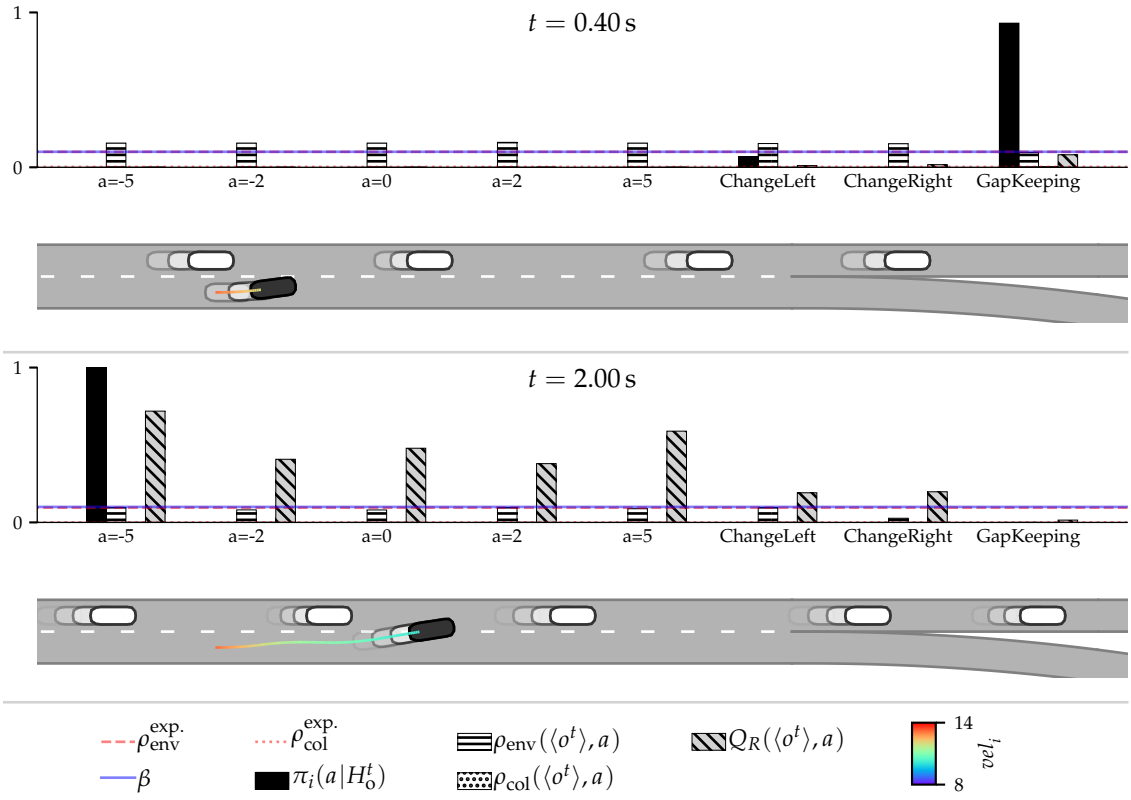


Figure 6.10.: Analysis of the RC-RSBG planner's risk-constrained stochastic policy at risk level $\beta = 0.1$. The planned stochastic policy π_i balances action-risk estimates, $\rho_{\text{env}}(\langle o^t \rangle, a_i)$ and $\rho_{\text{col}}(\langle o^t \rangle, a_i)$ yielding an expected envelope risk $\rho_{\text{env}}^{\text{exp.}}$ fulfilling the risk constraint $\beta = 0.1$ while the expected planned collision risk $\rho_{\text{col}}^{\text{exp.}}$ is close to zero and higher returns $Q_R(\langle H_o \rangle, a_i)$ are preferred. A lower risk level results in the ego vehicle slowly approaching the target lane.

formalism (cf. Chapter 4). It is evaluated in the following section.

6.3. Evaluating Risk-Constrained Planning

This section evaluates the concepts proposed in Chapter 4. It starts with analyzing qualitatively the policies generated by risk-constrained stochastic policy optimization in Sec. 6.3.1. Sec. 6.3.2 compares the RC-RSBG to the baseline planners. Finally, the interpretability of the proposed risk formalism and its capability to balance safety and efficiency are evaluated in Sec. 6.3.3.

6.3.1. Analyzing Risk-Constrained Stochastic Policies

First, scenarios driven with the RC-RSBG planner are qualitatively analyzed in the freeway enter scenario for differing envelope violation risk levels $\beta = 0.1$ in Fig. 6.10 and $\beta = 0.6$ in Fig. 6.11. The planned stochastic policy π_i (black bars) correctly balances envelope and collision action-risk estimates, $\rho_{\text{env}}(\langle H_o^t \rangle, a_i)$ (vertically-striped bars) and $\rho_{\text{col}}(\langle H_o^t \rangle, a_i)$ (dotted bars) such that the expected planned envelope risk $\rho_{\text{env}}^{\text{exp.}}$ (dashed red) fulfills the respective risk constraint β (blue) while the expected planned collision risk $\rho_{\text{col}}^{\text{exp.}}$ (dotted red) is close to zero. Given

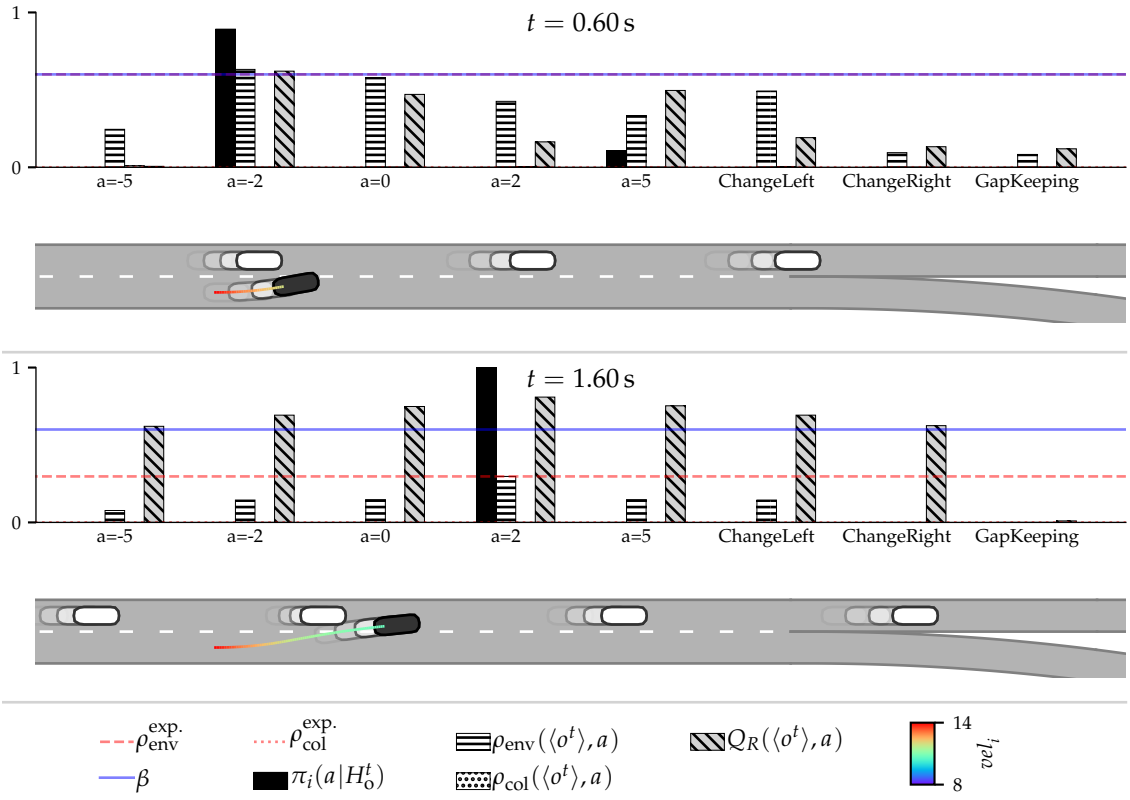


Figure 6.11.: Analysis of the RC-RSBG planner's risk-constrained stochastic policy at risk level $\beta = 0.6$. As presented in Fig. 6.10, yet, a higher risk level results in an abrupt cut-in of the ego vehicle directly in front of its rear vehicle.

these constraints, the planned policy prefers actions with higher expected action-return values $Q_R(\langle H_o \rangle, a_i)$ (diagonally-striped bars).

The RC-RSBG planner can generate two types of stochastic policies. On the one hand, if feasible, it plans a deterministic policy, i.e., a stochastic policy with a single action taking probability one, such that the expected risk of a single action matches the specified risk level (Fig. 6.10, $t = 2.0\,\text{s}$) or falls below the specified risk level (Fig. 6.11, $t = 1.6\,\text{s}$). On the other hand, if a deterministic policy cannot satisfy the constraints, a stochastic policy is planned to balance $\rho_{\text{env}}^{\text{exp.}}$ and $\rho_{\text{col}}^{\text{exp.}}$ such that the expected envelope and collision risk do not exceed the constraints (Fig. 6.10, $t = 0.4\,\text{s}$ & Fig. 6.11, $t = 0.6\,\text{s}$).

At a lower specified risk ($\beta = 0.1$) the ego vehicle conservatively approaches the target lane (cf. driven trajectory in Fig. 6.10, $t = 2.0\,\text{s}$) by going straight shortly before crossing the lane boundary. Such behavior is similar to how human drivers sometimes indicate to other drivers the desire to change lanes. Interestingly, the risk level $\beta = 0.1$ for which this behavior arises is around the percentage of envelope violations of human drivers during lane changes of 4 % to 8 % (cf. Sec. 4.1.1). In contrast, at a higher allowed risk level ($\beta = 0.6$, cf. Fig. 6.11), the ego vehicle performs a cut-in at high velocity shortly before the other rear vehicle. Similar differences in behavior are observed for the left turn scenario in App. A.7. Overall, a natural behavior arises solely by constraining the allowed envelope violations over time without tuning safe distance

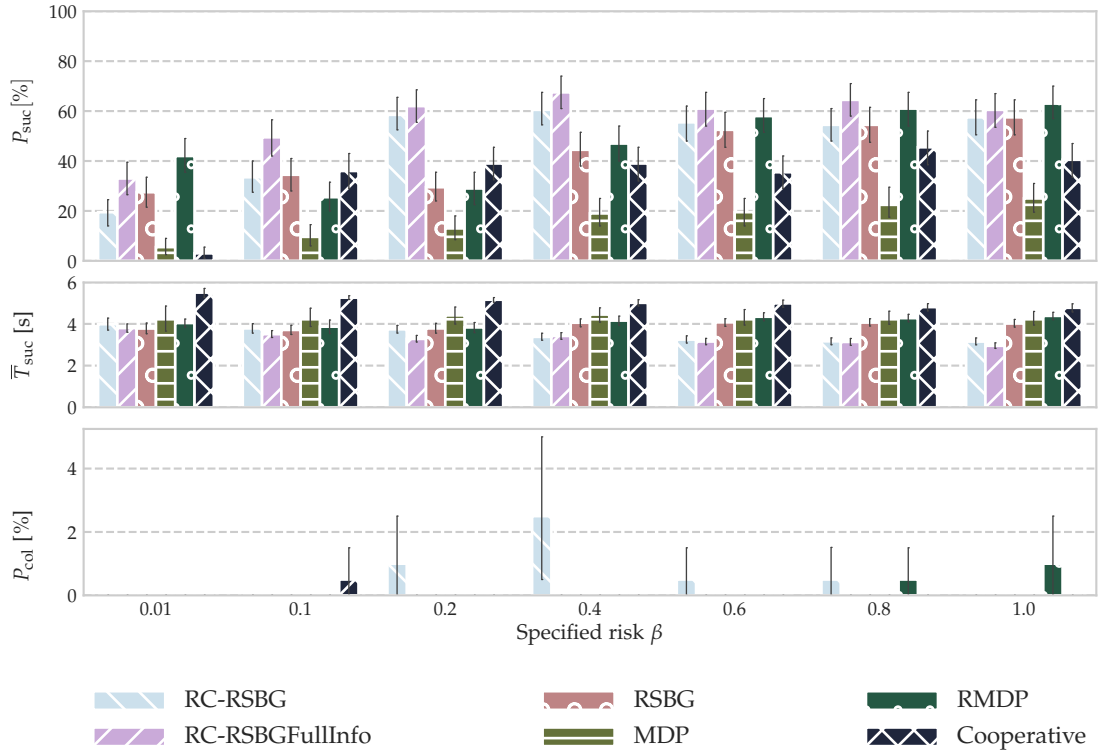


Figure 6.12.: Performance of the RC-RSBG and baseline planners in the freeway enter scenario.

margins or other cost terms.

The above findings reveal that the RC-RSBG planner correctly implements the risk-constrained optimality criteria from Sec. 4.1.2 using a stochastic policy. The qualitative differences in ego behavior for different risk levels show that the interpretable risk formalism balances the safety and efficiency of the planned behavior. Quantitative analyses are given in the following sections.

6.3.2. Evaluating the Performance of the RC-RSBG Planner

Next, the performance of the RC-RSBG planner is evaluated over increasing envelope risk constraint β. For the single-objective baseline approaches, the risk-aware reward function defined in Eq. (6.3) is applied. All planners apply a fixed number of 20k iterations. Results are given in Fig. 6.12 for the freeway enter and in Fig. 6.13 for the left turn scenario.

The success percentages of RC-RSBG and RCRSBGFullInfo planners increase steadily from $\beta = 0.01$ to $\beta = 0.4$. Risk levels $\beta > 0.4$ do not further increase the successes. Thereby, the average time to reach the goal steadily decreases. With higher β the RC-RSBG planner increasingly relies on the accuracy of the prediction model and lesser on the safety provided by the envelope restriction. In the case of prediction model inaccuracies, this provokes collisions for $\beta \geq 0.2$. The RCRSBGFullInfo planner does not show any collision in freeway enter due to having full access to the actual behavior of other participants. In left turn, however, it collides for $\beta \geq 0.2$, yet, achieves an overall high success rate.

For the baseline planners, the success rates decline partly when increasing β, e.g., from $\beta = 0.01$

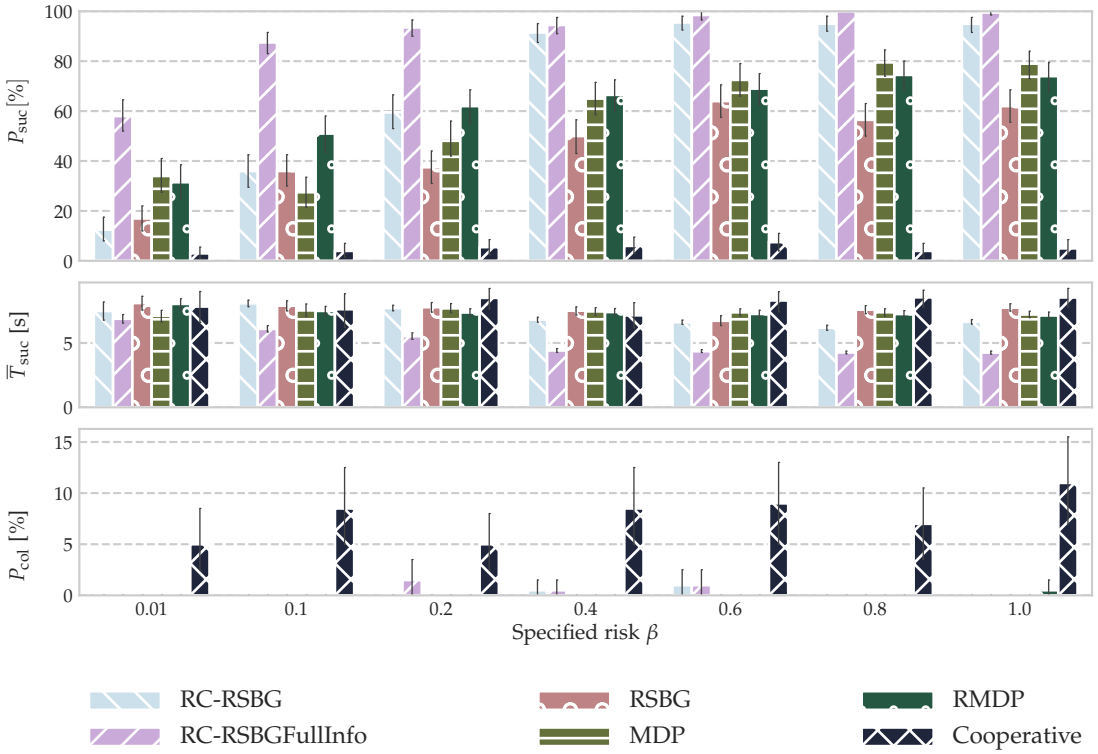


Figure 6.13.: Performance of the RC-RSBG and baseline planners in the left turn scenario.

to $\beta = 0.1$ for RMDP in freeway enter, or for MDP from $\beta = 0.01$ to $\beta = 0.1$ in left turn. A steady increase is thus not observable. The time to reach the goal remains near-constant. In freeway enter, the performance of the MDP planner drops drastically compared to using the simplistic reward function without risk (cf. Sec. 6.2.4). The Cooperative planner can not benefit from the risk-aware reward and still collides frequently in left turn. In contrast, by benefiting from belief information, the RSBG planner manages to achieve near equal success rates in freeway enter than the RC-RSBG planner and provokes fewer collisions than the RC-RSBG planner for $\beta \geq 0.2$.

In summary, the risk-aware reward function used by the single-objective baseline planners does not reveal a clear relation between β and the safety (P_{col}) and efficiency (P_{suc} and $\overline{T}_{\text{suc}}$) statistics. In contrast, such a relation is observable with the RC-RSBG and RCRSBGFullInfo planners. With the risk-aware single-objective reward function, the action return is always reduced when an action violates the safety envelope, independent of the chosen β, and the planner always tries to reduce envelope violations independent of the actual constraint. In contrast, with the RC-RSBG and RCRSBGFullInfo planners, such a violating action can still be preferred as long as the constraint is satisfied. The difference in optimality assumptions explains the more passive behavior generated by the single-objective planners with lower success and collision percentage for higher risk levels. This finding indicates the usefulness of multi-objective optimality to integrate risk constraints and supports the motivation and definition of the interpretable risk formalism (cf. Sec. 4.1).

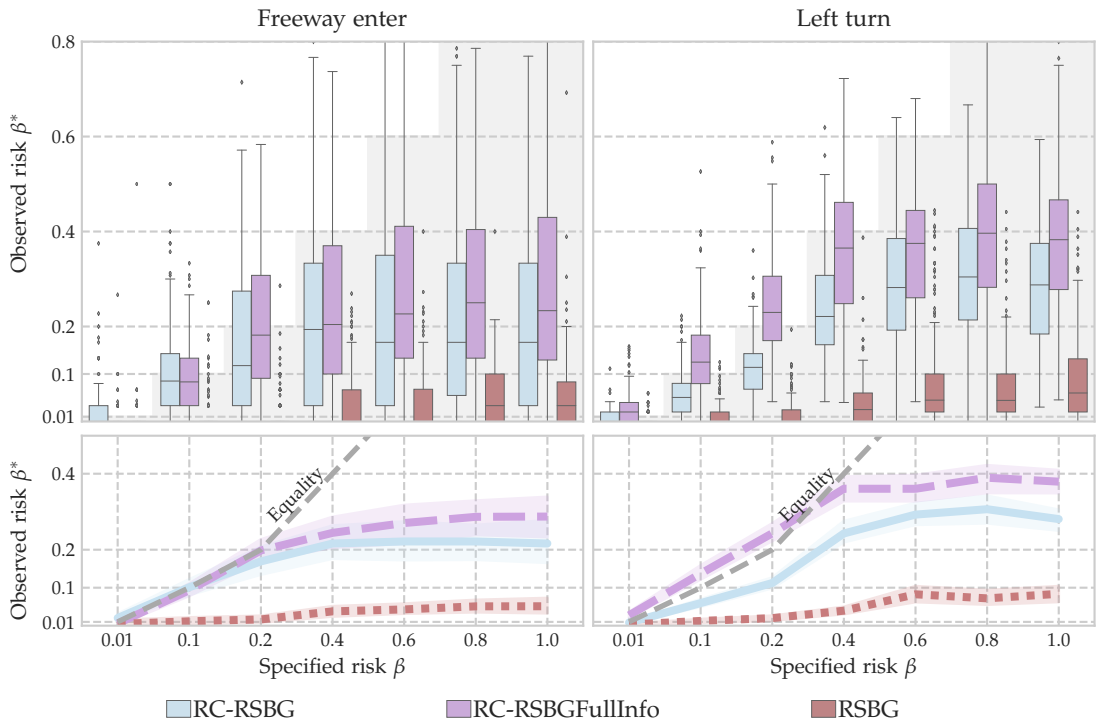


Figure 6.14.: Comparison of observed envelope violation risks. When satisfying the interpretable risk formalism (RC-RSBG and RCRSBGFullInfo), the specified risk level reflects the observed risk. Single-objective specifications of risk (RSBG) cannot achieve such an interpretable relationship.

6.3.3. Studying the Practicality of Interpretable Risk to Balance Safety and Efficiency

This section further analyzes how β balances safety and efficiency by looking at the observed envelope violation risk β^* and the expected scenario waiting time t_w for the results of the previous section.

The mean envelope violation risk (cf. Fig. 6.14, bottom) is equal to the allowed risk ($\beta^* \approx \beta$) for $\beta \leq 0.2$ in freeway enter for RC-RSBG and RC-RSBGFullInfo and $\beta \leq 0.4$ in rural left turn for RC-RSBGFullInfo. The correspondence of risk levels indicates that the *interpretable risk formalism and planning approach reflects the observed risk*. In contrast, the RSBG planner does not show any interpretable correlation between β^* and β.

Starting from $\beta \geq 0.2$ in freeway enter, and $\beta \geq 0.4$ in left turn, the observed risk β^* saturates and does not increase further. On the one hand, this is reasonable in a traffic environment since above a certain β efficiency can not be improved further by more frequently violating the envelope. On the other hand, this shows that the RC-RSBG planner interprets the risk level as an actual constraint since the observed risk is fully exploited for lower allowed risk.

Analyzing the distribution of the observed envelope violation risk (cf. Fig. 6.14, top) reveals that for $\beta < 0.4$, in some scenarios, the allowed envelope violation risk is exceeded. The occurrence of such outliers is in line with the interpretable risk formalism, defined using an expectation over uncertain future observations, and underlines the stochastic nature of the gen-

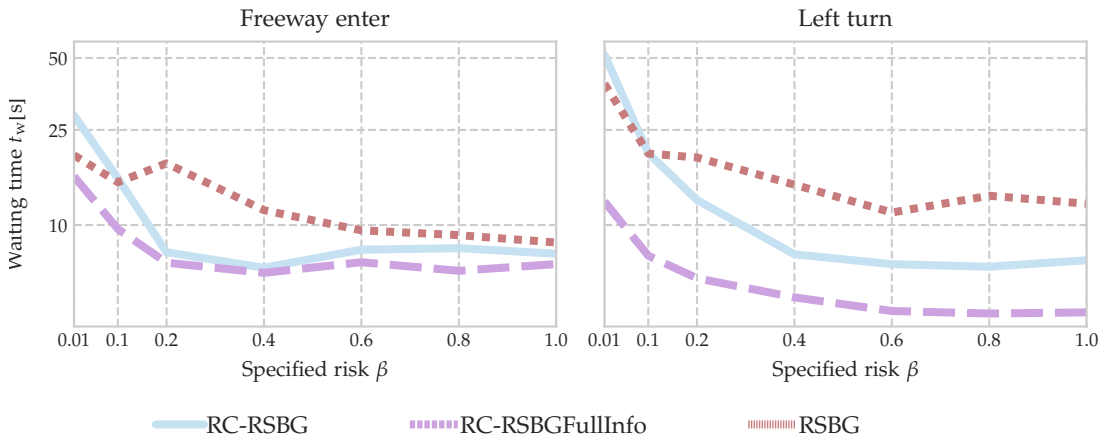


Figure 6.15.: Comparison of scenario waiting times. The waiting time is based on the results of Fig. 6.12 and Fig. 6.13. The expected time to solve a scenario smoothly declines when satisfying the interpretable risk formalism (RC-RSBG and RCRSBGFullInfo). Such a continuous decline is not observed with single-objective specifications of risk (RSBG).

erated ego behavior. This stochastic interpretation of safety is in contrast to safety formulations which altogether forbid envelope violations and assume strict determinism of the environment and policy (cf. Sec. 1.1.1).

The waiting time (cf. Fig. 6.15) subsumes the metrics P_{suc} and $\overline{T}_{\text{suc}}$ to provide a comprehensive measure indicating efficiency, while P_{col} (cf. Fig. 6.12 and Fig. 6.13) represents the safety measure. Interestingly, the collision results suggest to choose $\beta \leq 0.1$ to avoid collisions in freeway enter which resembles the time-based safety envelope violation risk of humans during lane changing, $\beta_{\text{human}} \leq 10\%$ [6]. Similarly also the left turn scenario requires $\beta \leq 0.1$ to prevent collisions. The waiting time declines smoothly with increasing β for $\beta \leq 0.4$ in freeway enter and over the full range of β in left turn for RC-RSBG and RC-RSBGFullInfo planners. In contrast, the single-objective planning with the RSBG planner does not show such a continuous decrease of the waiting time.

Overall, the proposed risk formalism and RC-RSBG planner provide an interpretable way to balance safety and efficiency given

- the relation of the risk level β to the human safety statistics (cf. results in Sec. 6.3.1 and Sec. 6.3.2),
- the option to compromise collisions (cf. results in Sec. 6.3.2) and efficiency (cf. Sec. 6.3.3) using β, and
- the correspondence between specified and observed envelope violation risk (cf. results in Sec. 6.3.3).

Given the quantitative results for $\beta = 0.1$ (no collisions, medium success, and waiting time) and the qualitative analysis (natural lane changing and turning behavior, similar safety statistics as humans to prevent collisions) in previous sections, the risk level is fixed to $\beta = 0.1$ for the following evaluations.

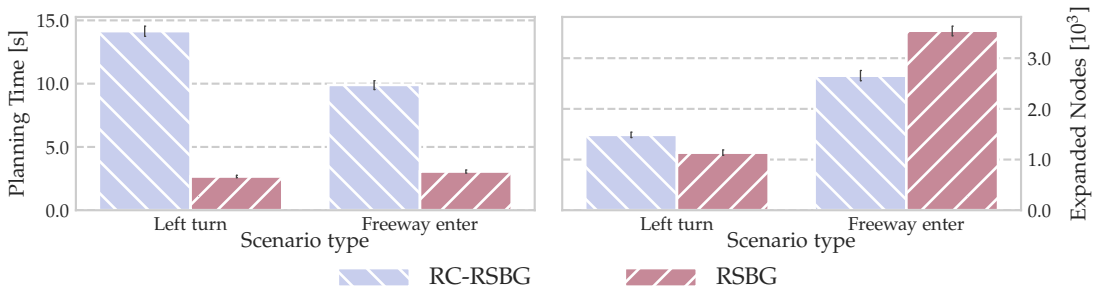


Figure 6.16.: Comparison of planning times and expanded nodes. The planning time of the RC-RSBG planner is significantly increased compared to the RSBG planner due to the multi-objective nature of risk-constrained interactive planning.

6.4. Evaluating Experience-Based and Parallelized Risk-Constrained Planning

This section evaluates the concepts proposed in Chapter 5. Sec. 6.4.1 analyzes computational demands of the RC-RSBG planner. The performance benefits of root-parallelization are evaluated in Sec. 6.4.2. Sec. 6.4.3 analyzes how risk-constrained planning can be accelerated with prior experiences through value initialization.

6.4.1. Evaluating the Computational Demands of the RC-RSBG Planner

Multi-objective planning comes with additional computational demands since it requires solving a constrained optimization problem repeatedly during planning (cf. 4.3.1). Solving the linear program in the stochastic action selection procedure of the RC-RSBG planner requires more than 200 µs[§].

This section analyzes how the additional computational demands of multi-objective planning affect the RC-RSBG planner. Fig. 6.16 depicts the planning times and number of expanded nodes of the RC-RSBG and RSBG planners when performing a fixed number of 20k search iterations.

The number of expanded nodes, i.e., tree states, is comparable for both planner types yet, significantly lower than the number of iterations, especially in the left turn scenario. This difference is due to the number of expanded nodes not increasing further when all valid, i.e., non-terminal collision or success states, have been explored. Subsequent iterations only perform action selection steps to refine the node statistics. The lower number of valid states in the left turn scenario arises due to the lower action space and the driving area of the ego vehicle being more restricted than in the freeway enter scenario.

The planning time of the RC-RSBG planner increases, as expected, significantly compared to the RSBG planner for a fixed number of iterations due to the overhead of solving the linear program in each selection step. Interestingly, a significant difference in planning times between the left turn and freeway enter scenario is observable with the RC-RSBG planner. This difference exists potentially due to a differing number of performed action selection steps. In the left turn

[§]Using a C++ implementation based on `https://developers.google.com/optimization` on an Intel 3.2 Ghz CPU.

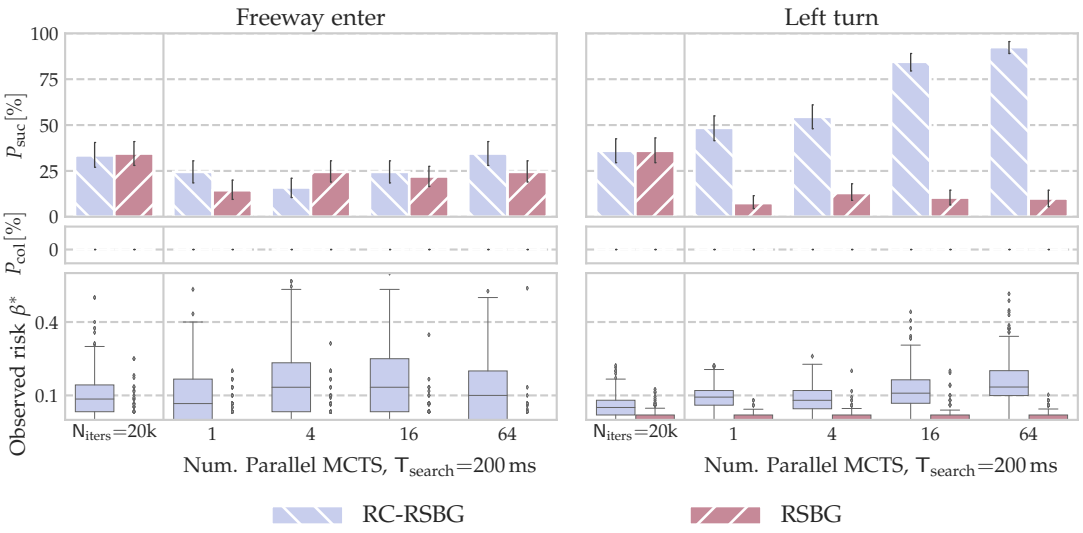


Figure 6.17.: Comparison of parallel multi- and single-objective planners. The success percentage of the RC-RSBG planner (multi-objective) steadily increases with the number of parallel MCTS while the observed risk satisfies the risk constraint $\beta = 0.1$. Such benefits from parallelization are not observed with the RSBG planner (single-objective).

scenario, the search tree contains fewer expanded nodes which increases the number of executed action selection steps due to the constant iteration number. In contrast, in freeway enter, the search tree contains an increased number of expanded nodes. The planning time reduces since the linear program is not solved when new states are created in the expansion step.

Overall, the planning time of the RC-RSBG planner increases significantly due to the multi-objective nature of the planning problem. This drawback limits the applicability of the RC-RSBG planner in an actual AV. For this, the planning component should achieve an update frequency below 200 ms. The following sections evaluate how to reduce the planning time of the RC-RSBG planner without sacrificing performance.

6.4.2. Comparing Parallelization of Single- and Multi-Objective Planning

This section compares the benefits of root-parallelization in multi-objective and single-objective planning. For this, the performances of root-parallel implementations of the RC-RSBG and RSBG planner are evaluated over an increasing number of parallel MCTS while restricting the total allowed planning time to the desired update frequency of 200 ms. The exploration parameters are adjusted as in the previous evaluations. Fig. 6.17 depicts the success rates and the observed risk. Collisions did not occur in all evaluated settings. The results for 20k iterations from Sec. 6.3 and 6.4.1 for $\beta = 0.1$ are given as a reference for planning without time constraints. Details on the experiment setup with limited planning time are given in App. A.8.

A drop of the success rate compared to the reference results occurs for both planners in freeway enter when limiting planning time to 200 ms (number of parallel MCTS is one). Nevertheless, only the RC-RSBG planner reaches the reference performance for 64 MCTS. In left turn, the RC-RSBG planner manages to outperform the reference result significantly. The success percentage

increases steadily with the number of parallel MCTS. It seems that parallelization has larger benefits in scenarios that require a more accurate prediction of the correct homotopic variants, e.g., in left turn, but do not demand a detailed microscopic prediction, as, e.g., required in freeway enter. Increasing the number of parallel MCTS with single-objective planning, i.e., the RSBG planner, achieves a marginal gain in performance in freeway enter but completely loses performance in the left turn scenario.

Comparing the observed risk to the specified risk level $\beta = 0.1$ shows that the risk constraint is still satisfied with root-parallelization. It seems that the reference planner and the single MCTS with limited planning time fulfill the constraint more conservatively, i.e., with lower deviations around the mean of the observed risk. However, parallelization exactly meets the constraint for 64 MCTS in freeway enter. In left turn, the constraint is satisfied for 4 and 16 and slightly violated for 64 MCTS.

The above results underline the advantages of parallelizing multi-objective planning, i.e., the RC-RSBG planner, motivated in Sec. 5.5. Given the additional risk information, a final plan can be better combined from parallel search runs and fulfill both the goal-directed and the risk-constrained optimality specifications.

6.4.3. Comparing Experience- and Rollout-Based Exploration

This section evaluates experience-based risk-constrained planning. For each scenario type, experience data is generated according to the data generation process described in Sec. 5.4 and a neural network g_v is trained to predict value experiences according to Sec. 5.3. Details on the training process, parameters, and results are given in App. A.9.

The warming starting of search nodes with learned value experiences replaces rollout-based exploration. Further, preliminary experiments revealed that warm starting allows omitting count-based exploration, i.e., setting the exploration parameter $\kappa = 0$, and only exploring via the stochastic action policy. Thereby, the filter factor controlling the support of the stochastic ego policy is reduced to $v = 0.2$ given the higher initial accuracy of the risk and return estimates with warm starting. In rare cases, when inputting learned risk estimates, the linear program solver requires unlimited computation time without categorizing the problem as unsolvable. Thus, the available solver time is limited to 2 ms for the experience-based planner.

The evaluation compares the performances of two experience-based (EB) variants of the RC-RSBG planner, 1) a single and 2) a parallelized variant with 64 MCTS, each with restricted planning time of $\mathsf{T}_{\text{search}} = 200\,\text{ms}$. Additionally, two rollout-based results are given as a reference, 1) the variant with maximum iterations $\mathsf{N}_{\text{iters}} = 20\,\text{k}$ evaluated in Sec. 6.3 and 2) the planning-time-restricted single MCTS variant evaluated in the previous Sec. 6.4.2 and 6.4.1. The success and collision percentage, and the observed risk and number of performed iterations for the above EB and rollout variants are depicted in Fig. 6.18. Details on the experiment setup with restricted planning time are given in App. A.8.

EB planning can nearly double the success percentage compared to non-EB planning in freeway enter. The parallelized variant avoids collisions. In left turn, no benefits of experience-based planning regarding success and collision percentage arise compared to the parallelized

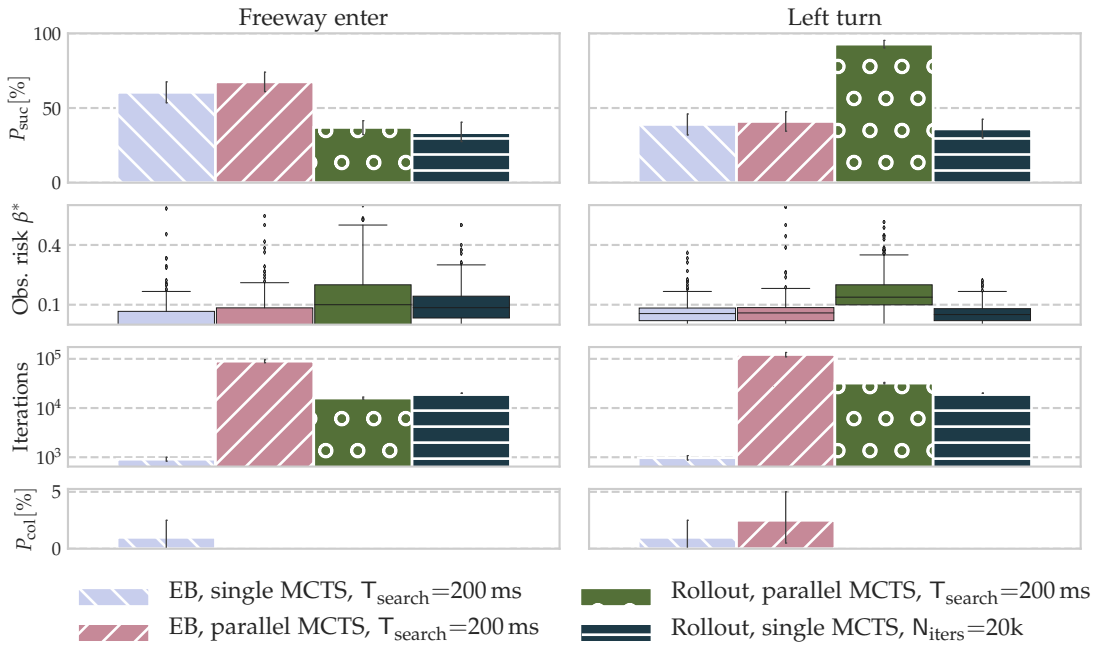


Figure 6.18.: Comparison of the experience- and rollout-based RC-RSBG planner. Experience-based planning significantly increases success rate in scenarios with a larger action space (freeway enter). In scenarios with a smaller action space (left turn) no benefits are observable compared to solely parallelizing the search.

rollout-based planner. In both scenarios, EB planning more accurately fulfills the risk constraint $\beta = 0.1$ than the rollout-based variant 2).

The results show that warm starting in combination with the presented data generation and learning of prior experiences is a meaningful technique to guide the search of the RC-RSBG planner. Significant benefits of EB planning arise in scenarios requiring larger action spaces as given in freeway enter. Here, warm starting helps reduce the support of the stochastic policy, which narrows the width of the search tree. These benefits cannot arise in scenarios with lower action space as in left turn. Additionally, the worse performance of EB in left turn can be explained due to a worse training result in left turn compared to freeway enter. Details are given in App A.9. Nevertheless, solely parallelizing the search is already sufficient in left turn to achieve a satisfying online planning performance.

Prior experiences enable online planning. Though EB planning spends computation time on the inferences of the value network, it avoids the computational demands of the rollout step. The number of performed search iterations of the rollout-based parallel variant is by order of magnitude lower than for the experience-based parallel variant indicating a reduction of computation time when performing neural network inference instead of a rollout step in each iteration.

Overall, the evaluations in Sec. 6.4.2 and Sec. 6.4.3 show that parallelized and experience-based variants of the RC-RSBG planner each have advantages in specific scenario domains for accelerating risk-constrained planning. The results suggest a potential applicability of the RC-RSBG planner to enable real-time risk-constrained interactive planning in AVs.

6.5. Summary of the Evaluation

In summary, the evaluation contributes three significant findings.

The evaluation shows that the RSBG planner outperforms existing non-belief-based planners to navigate through congested traffic successfully. It thereby benefits from sample-efficient planning due to robustness-based optimality and prediction using behavior hypotheses in behavior spaces superior to intent-based prediction approaches. A static safety margin to account for planning and prediction inaccuracies and the risk-based reward setting of the single-objective RSBG and baseline planners cannot provide an interpretable way to balance safety and efficiency.

The evaluation of the RC-RSBG planner reveals that the risk level β balances safety and efficiency in an interpretable manner. The qualitative analysis shows that the AV performs lane changing and left turning in a natural, human-like way at a risk level of $\beta = 0.1$. The quantitative analysis shows the correspondence between specified and observed risk levels and a continuous increase of efficiency and decrease of safety when raising the risk level. For risk level $\beta = 0.1$, no collisions are provoked in the statistical analyses for the freeway enter and left turn scenario. Since the risk level $\beta = 0.1$ is in the range of human time-normalized safety envelope violations (cf. 4.1.1), this resembling indicates that the proposed risk formalism may indeed be connected to the human understanding of risk in the evaluated scenarios.

The evaluation of the experience-based and parallelized RC-RSBG planner reveals that risk-constrained interactive planning works under limited planning time. Experience-based and parallelized planning increases efficiency while ensuring the satisfaction of the risk constraint. Thereby, parallelization was shown to be beneficial when the scenario type requires deeper exploration of the problem space along the time dimension. In contrast, experience-based planning was found to improve efficiency in scenarios requiring a larger action space.

Overall, the evaluation demonstrates that the RC-RSBG planner enables an AV to balance safety and efficiency in an interpretable manner when navigating through simulated dense traffic scenarios with uncertainty about other participants behavior. These properties of the RC-RSBG planner are preserved in an online planning situation, restricting the available planning time.

7 Future Work

This chapter outlines research directions carrying forward the work presented in this thesis. Sec. 7.1 discusses how to improve the design of behavior spaces and hypotheses. The integration of perception and execution uncertainty is discussed in Sec. 7.2. Sec. 7.3 proposes to model errors in the optimality of a plan as epistemic uncertainty within the risk formalism and planner. Necessary research to transfer the Risk-Constrained Robust Stochastic Bayesian Game (RC-RSBG) planner to real AVs are presented in Sec. 7.4. Sec. 7.5 proposes research tasks utilizing the interpretable risk formalism to argue about the probability of collision from a safety engineering perspective.

7.1. Improving Behavior Spaces and Hypotheses

The results in Sec. 6.2.1 show a link between planning performance and behavior parameters, e.g., T_{desired}, used to design the behavior space. Future research should further investigate how the hypothetical policy and the behavior space connect to the RC-RSBG planning performance and how prediction using behavior spaces compares to the prediction and planning capability of other interactive planners.

Analytical Hypothetical Policies A possible research direction in this regard is investigating different designs of analytical models for the hypothetical policy. The comparison should include classical driver models, i.e., variants of the Intelligent Driver Model (IDM) [267, 268] and other lane following models [269, 270], but also models outputting lateral actions such as the MOBIL model [271] and assess the usefulness of each driver model for behavior-space-based prediction. Classical driver models often require tuning multiple parameters leaving unclear what parameters mainly dominate during interactions. Artificial analytical driver models could provide better understandable models by employing only a single hidden behavior parameter for each considered direction of movement to achieve optimal separation of hypotheses (cf. Sec. 3.3.3). Such a single parameter could express the desire to move into a certain longitudinal or lateral direction and be inspired by physical concepts such as friction or inertia. Ideally, such a model

would transfer to different types of participants and predict other vehicles' lane changes and behavior variations of pedestrians and bicyclists.

Learning Hypothetical Policies The hypothetical policy can also be learned from human driving data. For this, the learned hypothetical policy, predicting human driving actions, must be based on a compact latent space with only a few continuous parameters. As a starting point, latent architectures, e.g., using autoencoders [272], can be combined with neural-network-based prediction [91]. After learning, the learned latent space defines the behavior space and hypotheses set. Other input features may remain. Preliminary research tasks are to develop a meaningful neural network architecture and latent feature training process and establish a connection between the size of the latent space and prediction accuracy for different learned models. Fig. 7.1 depicts a possible architecture and training process.

Evaluation of Prediction Capabilities Future evaluations should compare the planning performances for different hypothetical policies and designs of behavior spaces and integrate recorded human driving data to judge the quality of behavior-space-based prediction in real-world situations. The INTERACTION dataset [273] is already partly integrated into BARK. It consists of a variety of dense interactive traffic situations. The accuracy of the prediction can be measured by evaluating the prediction probability given by the mixture distribution over hypotheses (cf. (3.1)) at the ground truth human driving action from the next simulation step.

Detailed Baseline Planner Evaluation Performance differences between MDP, RMDP, cooperative and intent-based planners are given and analyzed in the evaluation. Nevertheless, future work should improve the understanding of the prediction capabilities and limitations obtained with each model. A comprehensive parameter study should analyze the benefits and drawbacks of each prediction model when, e.g., varying traffic densities, cooperativeness level of the cooperative planner, behavior-spaces of MDP and RMDP planner. The analysis should also evaluate the benefits of belief-state planning compared to the QMDP approximation used in this thesis when applying intent-based prediction.

7.2. Integration of Other Uncertainty Types

This thesis focuses on the definition of and planning under an interpretable risk given the uncertainty of other participants' behaviors. Nevertheless, in real-world driving (cf. Fig. 2.1), the risk of violating safety envelopes is also influenced by uncertainties about sensed ego and other participants' states and the presence of objects, e.g., invisible due to occlusions. Further, controller uncertainties, e.g., inaccurate state tracking and delays in execution, influence the observed risk. Future work should investigate the influence of perception and execution uncertainty on the interpretable risk and how to integrate these uncertainties into the RC-RSBG planner.

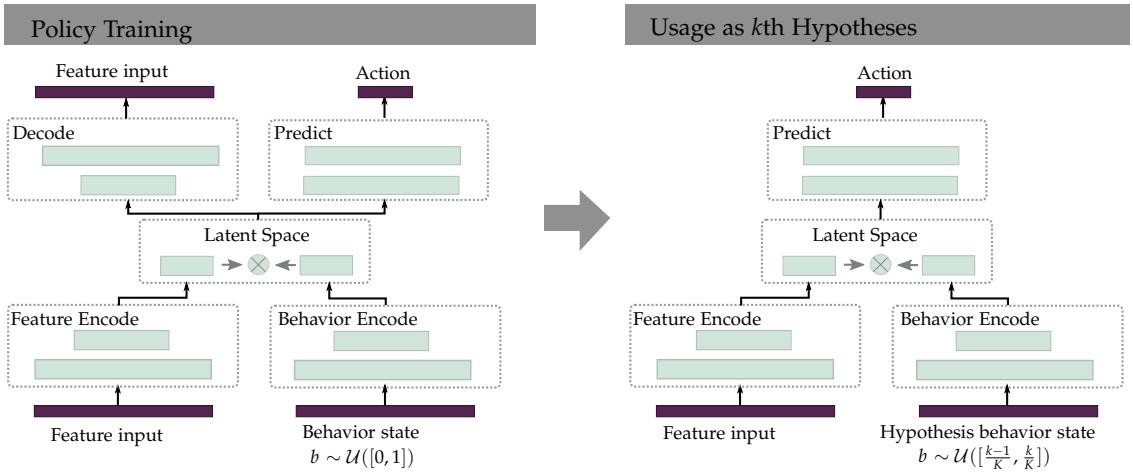


Figure 7.1.: Representation of hypothetical policies using neural networks. During training, the latent space combines state features with encoded behavior features and learns to represent action variations in the data given the randomness over the full behavior space $\mathcal{B} = [0, 1]$. The decoding stage helps to find a compact latent space. The trained network becomes the kth hypothesis by sampling from only the kth partition of the behavior space.

Consideration during Planning A first option is modeling that the perception and execution uncertainty change the observation sequence distribution used in the definition of the violation risk (cf. Def. 4.1). Currently, the observation sequence distribution depends only on the participants' policies. Therefore, an open research task is to integrate perception uncertainty available in the form of belief information over the observed states of ego, other and occluded vehicles into the risk formalism. In principle, the observable initial states at the beginning of all observation sequences must be replaced with a belief distribution over initial states. The execution uncertainty adds variations to the states in the observation sequences. Planning under such an adapted risk formalism could integrate state beliefs with root sampling and execution uncertainty by applying a stochastic environment transition function EnvironmentMove. When realizing this option for integrating perception and execution uncertainty, the primary research task is to make the integration computationally feasible, e.g., by extending and improving the worst-case action selection.

Probabilistic Safety Envelopes A second option is to limit the available action space of the RC-RSBG planner based on probabilistic safety envelopes defined for the perception and execution uncertainties. The envelope definitions from Sec. 4.5 guarantee safety only in the case of accurate state estimates. Recent work by the author of this thesis in [274] proposes a safety envelope definition that provides a maximum risk of violating the actual safety envelope given uncertainty in the state estimates of other participants (cf. Fig. 7.2). Future research should investigate how such probabilistic envelopes, also extendable to execution uncertainties, can restrict the available set of actions of the RC-RSBG planner. The overall risk of violating a safety envelope could be combined probabilistically.

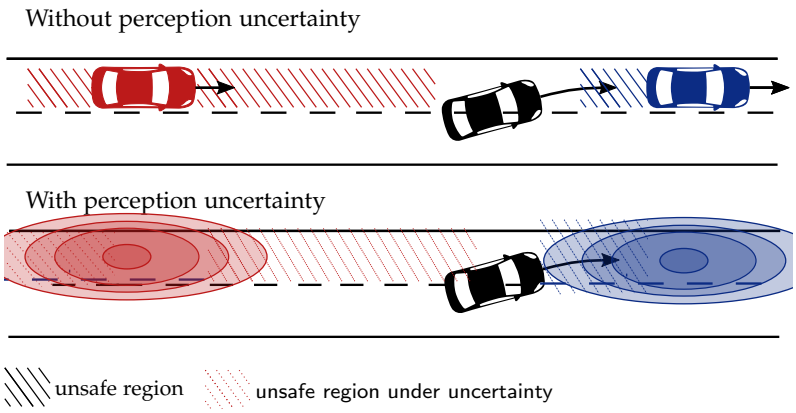


Figure 7.2.: This thesis neglects perception uncertainty and defines safety envelopes using physical limits of vehicle dynamics (top). Probabilistic safety envelopes, recently proposed by Bernhard et al. [274], deal with perception uncertainty. By allowing only actions fulfilling these envelopes, the RC-RSBG can additionally incorporate perception, and similarly also execution uncertainties. Picture taken from [274].

Evaluating Inaccuracies in Perception and Execution The approaches from the proposed research directions should then systematically be benchmarked. For this, as a starting point, BARK has already been extended to simulate perception inaccuracies in [274]. Similarly, a simulative analysis of execution uncertainties can be implemented in BARK.

7.3. Modeling the Influence of Solution Inaccuracy onto Risk

This thesis' risk definition and risk-constrained planner focus on how aleatory uncertainty, i.e., the randomness inherent to the environment due to prediction uncertainty, influences the risk level. The thesis analyzes the efficiency of robustness-based planning to detect worst-case outcomes and shows that prior experiences and parallelization enable online risk-constrained planning. However, it remains open how these concepts quantitatively affect the optimality of a plan and influence the observed envelope violation risk. An important future research direction is thus modeling the optimality of the solution as a form of epistemic uncertainty and integrating it into the risk formalism and RC-RSBG planner. Risk-constrained planning integrating epistemic uncertainty would, for instance, make the Autonomous Vehicle (AV) behave more safely to avoid envelope violations caused by insufficient exploration of the search space. Three research tasks are proposed as a starting point aiming at the integration of epistemic uncertainty into the RC-RSBG planner.

Representation This task aims to find an interpretable representation of epistemic uncertainty for action-return and action-risk estimates. For such a definition to be meaningful, it must integrate into the aleatory risk definition. Thereby, the interpretability of the risk definition and validity of the planning approach must be preserved. As a starting point, this task compares the effectiveness of probabilistic approaches applying, e.g., histograms or Gaussian models [275] with interval concepts often used to characterize epistemic uncertainty [276].

Backpropagation This task investigates how to backpropagate the epistemic uncertainty during MCTS. It may be inspired from existing work backpropagating epistemic uncertainty in MCTS using Bayesian approaches [275] or variance information [277], and from the field of risk analysis, which combines aleatory and epistemic uncertainties in fault tree [278] and sensitivity analysis [276].

Prior Experiences and Parallelization This task, on the one hand, analyzes how parallelization affects epistemic uncertainty. It should establish a quantitative connection between the number of parallel searches and the accuracy of the value estimates, which could be used to adjust the appropriate number of parallel searches automatically. On the other hand, this task investigates how to integrate epistemic uncertainty from the prior learned experiences, e.g., inferred with Bayesian neural networks [279] into experience-based planning. Such methods could deepen the understanding of how experience training results connect to the performance of the RC-RSBG planner.

7.4. Real-World Navigation with Adaptation of Risk and Behavior Spaces

This thesis focuses on and evaluates scenario-based planning. However, in the real world, the AV must consecutively navigate in the environment requiring the planning module to handle transitions between different scenario types, e.g., switching from a left turning to a lane following scenario type. Though one could set up the RC-RSBG planner with a continuous reward function, e.g., based on a velocity potential function, to avoid handling transitions, it makes sense to adjust risk and behavior spaces to the current situation.

Scenario Management The first research task aims to develop a higher-level scenario management module. It should detect the scenario type, e.g., based on the road layout, and select a goal definition and meaningful risk level. The selection can either be taken from a database and be adapted online to the observed traffic parameters such as traffic density and velocities or match more complex dynamic definitions of maneuver templates [280].

Behavior Space Management A second research task should develop a management module that adapts the hypothetical policy and behavior space to the encountered scenario type. For instance, one could design different hypothetical policies and behavior spaces given the approaches in Sec. 7.1. An idea to select the optimal model during online planning may be inspired by the concept of behavioral hypotheses testing presented for the Stochastic Bayesian Game (SBG) in [179]. Beliefs over hypotheses provide a relative measure of likelihood. In contrast, hypothesis testing analyzes if a hypothesis set's absolute measure of truth is correct. The approach in [179], in principle, performs a statistical hypotheses test using a score function with parameters learned during the interaction process. A hypothetical policy and behavior space are rejected if the p-value of the test is below a specified significance level.

Application on a Real Vehicle In a third major research task, extensions are developed for the RC-RSBG planner to run it as interactive planner within a complete AV architecture (cf. Fig. 2.1). This requires developing a trajectory smoother generating a trajectory out of the discrete action plan with more comfortable, continuous acceleration changes. However, smoothing can affect the optimality of the original discrete plan and the achieved risk level. Research should thus go into modeling the smoothing process either beforehand as epistemic uncertainty during planning (cf. Sec. 7.3) or developing an action space consisting of time-space corridors such that an action sequence defines the allowed region in which to plan a drivable trajectory subsequently. Another task is to analyze and improve the consistency of plans generated by the RC-RSBG planner under real-world conditions, characterized by sensing noise and deficiencies in trajectory tracking, and to analyze if the capability of online risk-constrained planning shown in simulation transfers to real-world situations.

7.5. Assuring Safety using the Interpretable Risk Formalism

This thesis proposes an interpretable risk formalism that constrains the time-normalized risk of violating a safety envelope. A risk level $\beta = 0.1$ avoids collisions in the evaluation. However, an open research question is how to assure from a safety engineering perspective that the amount of hazardous events, i.e., severe collisions in dense traffic, is acceptable for risk level $\beta = 0.1$. In the future, the dependence between the risk level as well as the severity and probability of collisions should be further analyzed to establish a safety argumentation based on the interpretable risk formalism. The following ideas are related to the safety concept outlined for risk-based safety envelopes by Bernhard et al. [274]. The developed approach should focus on arguing safety for traffic situations with low accident severity, as it is given, e.g., in slow, dense traffic. By that, it complements safety argumentations for high severity traffic given by fail-safe trajectory planning [32] or Responsibility-Sensitive Safety (RSS) [24]. Three major research tasks can contribute to such argumentation.

Concept of an Argumentation A first research task details the following idea for a safety argumentation. The envelope definitions in Sec. 4.5 guarantee an absence of collisions when the AV satisfies the safety envelope. However, humans frequently violate envelopes without provoking collisions [6]. The evaluation in Sec. 6.3 supports this observation when setting the interpretable risk level to $\beta = 0.1$. Given the probability $P_{\text{violate}\to\text{col}}(\text{m})$ that a safety envelope violation results in a collision during maneuver m, one can define the collision probability $P_{\text{col}}(m)$ of the ego agent during maneuver m when other vehicles *do not violate* their safety envelopes. For instance, when, for $m =$ "lane changing" and $\beta = 0.1$ in dense traffic, $P_{\text{violate}\to\text{col}}(\text{m}) = 10^{-3}$, then $P_{\text{col}}(m) = \beta \cdot P_{\text{violate}\to\text{col}}(\text{m}) = 10^{-4}$.

Measuring $P_{\textbf{violate}\to\textbf{col}}(\textbf{m})$ Determining and arguing that $P_{\text{violate}\to\text{col}}(\text{m})$ reliably holds in different encountered instances of a maneuver is the goal of the second research task. For this, for large amounts of scenarios recorded with digital infrastructures such as Providentia [281] and

generated in simulation, it must be analyzed under which circumstances envelope violations lead to a collision. Specifically, these analyses should include different parameterizations of the safety envelope definition, e.g., varying response times and maximum accelerations, and evaluate different traffic densities and velocities. For a safety argumentation to be accepted by legal authorities, different values of $P_{\text{violate}\rightarrow\text{col}}(\text{m})$ must then eventually be standardized similarly to exposure rates of different maneuvers given in the safety standard ISO26262 [282].

Severity of Safety Envelope Violations The risk formalism and RC-RSBG planner developed in this thesis consider *if* an envelope is violated or not. Metrics measuring *how* an envelope is violated, e.g., the severity of envelope violations [283], should be integrated into the risk definition and the RC-RSBG planner in a third research task. This integration can be achieved by changing the envelope indicator function to return continuous values, e.g., the severity of an envelope violation. The risk-constrained action selection of the RC-RSBG planner is based on solving a Constrained POMDP (C-POMDP), and thus supports straightforwardly the satisfaction of constraints for continuous cost criteria. Better estimates of probabilities $P_{\text{violate}\rightarrow\text{col}}(\text{m})$ can potentially be achieved when considering also the severity of envelope violations.

Overall, assuring the safety of AVs in dense traffic is a major challenge. Striving for a statistical relationship between envelope violations and accidents provides a step forward in solving this challenge.

8 Conclusion

This thesis develops an interpretable risk formalism and integrates it into an interactive planning algorithm for AVs. The approach balances safety and efficiency when navigating dense traffic, given uncertainty about other traffic participants' behaviors. The developed risk formalism includes both inter- and intra-driver behavior variations and provides a quantitative mapping between specified risk and observed safety statistics of the Autonomous Vehicle (AV). The presented risk-constrained interactive planner is capable of online planning.

First, this thesis deals with the problem of interactive planning given uncertainty about other participants' microscopic inter- and inter-driver behavior variations. It presents a concept to probabilistically predict microscopic variations using a combination of behavior hypotheses partitioning a behavior space and the sum posterior for belief tracking. The chapter then develops the Robust Stochastic Bayesian Game (RSBG), a game-theoretic model, which integrates robustness-based optimality to plan sample-efficiently using Simultaneous-Move MCTS (SM-MCTS) under continuous behavior variations of other participants.

Secondly, inspired by the statistics of human safety envelope violations, an *interpretable risk* is defined as the allowed maximum percentage of safety envelope violations over time. Planning under this risk formalism is modeled as Risk-Constrained Robust Stochastic Bayesian Game (RC-RSBG). A risk-constrained interactive planner is presented that integrates backpropagation of time-normalized risk estimates and risk-constrained stochastic ego action selection implementing a Constrained POMDP (C-POMDP) solver. The chapter concludes with safety envelope definitions for lane changing and intersection scenarios.

Thirdly, the thesis presents two concepts to reduce computational demands of the RC-RSBG planner. On the one hand, it develops warm starting with prior learned return and risk estimates to guide the node expansions of the RC-RSBG planner, and an accompanying data generation process and the learning of prior experiences. On the other hand, it discusses the benefits of parallelized planning in multi-objective problem domains as given with the RC-RSBG planner.

A statistical evaluation against various baseline interactive planners in the contributed simulation framework BARK then underlines the benefits of the proposed concepts. Prediction using hypotheses defined in behavior spaces increases success rate compared to planning with non-belief- and intent-based prediction. Robustness-based optimality better explores worst-case

outcomes by increasing the search depth. The interpretable risk measure serves to balance safety and efficiency. The specified allowed percentage of envelope violations correspond to the observed averaged envelope violations in simulation. Parallelized implementations of the RC-RSBG planner and integration of prior experiences result in improved efficiency while avoiding collisions, even when restricting the available planning time to real-time demands.

This thesis shows that certain aspects of the behavioral safety of an AV in dense traffic, i.e., its statistics of safety envelope violations, can be specified and statistically be interpreted with the proposed risk-constrained interactive planner. In domains like autonomous driving, statistical interpretability of the safety and other performance criteria will play an increasingly important role in enabling legal authorities to judge the quality of a system's behavior. Recent research [284] suggests that reward may be enough to define general artificial intelligence comprehensively. In contrast, this thesis shows that multi-objective optimality provides major benefits to achieve the requirement of statistical interpretability. An overall outcome of the future work presented in the previous chapter should, therefore, be to understand further the differences between single- and multi-objective planning to balance safety and efficiency in domains inside and outside of AVs.

A Appendices

A.1. Intelligent Driver Model

The Intelligent Driver Model (IDM) [93, 186] is a microscopic driver model that calculates a longitudinal acceleration acc_{IDM} for a rear vehicle V_R with

$$\text{acc}_{\text{IDM}} = \dot{v}_{\text{max}} \left[1 - \left(\frac{vel_{\text{R}}}{v_{\text{desired}}} \right)^4 - \left(\frac{\Delta s^*(vel_{\text{R}}, \Delta vel^t)}{\Delta s} \right)^2 \right]. \tag{A.1}$$

It integrates a free-road term which relates a parameter for the desired velocity v_{desired} to the current velocity of the IDM vehicle, vel_{R}. An interaction term relates the desired gap

$$\Delta s^*(vel_{\text{R}}, \Delta vel^t) = s_{\text{min}} + vel_{\text{R}} \cdot T_{\text{desired}} + \frac{vel_{\text{R}} \cdot \Delta vel}{2\sqrt{\dot{v}_{\text{factor}} \cdot \dot{v}_{\text{comft}}}} \tag{A.2}$$

depending on rear velocity vel_{R} and relative velocity Δvel between front and rear vehicle, to the current longitudinal gap Δs between front and rear vehicle. Further parameters of the model are the desired time headway T_{desired}, the minimum spacing s_{min}, the acceleration factor $\dot{v}_{\text{factor}}$, the comfortable braking $\dot{v}_{\text{comft}}$ and the maximum allowed acceleration $\dot{v}_{\text{max}}$. Extreme values of the accelerations can result when Δs becomes small. The maximum possible deceleration and acceleration are thus limited to physical feasible vehicle accelerations with limit parameters $\dot{v}_{\text{lim,+}}$and $\dot{v}_{\text{lim,-}}$.

A.2. Creation of Intelligent Driver Model Joint Distribution Data

To create the joint distribution over the IDM output given in Fig. 3.3, a rear vehicle V_{R} is located around $s_0 = 0$, an AV V_{A} at $s_0 = 15\,\text{m}$ and a front vehicle at V_{F} at $s_0 = 30\,\text{m}$. The vehicle lengths are $L_{\text{A}} = 4$. All IDM parameters, except T_{desired}, are kept constant with $v_{\text{desired}} = 50\,\text{km}\,\text{h}^{-1}$, $\dot{v}_{\text{factor}} = 1.7\,\text{m}\,\text{s}^{-2}$, $s_{\text{min}} = 1.0\,\text{m}$, $\dot{v}_{\text{comft}} = 1.7\,\text{m}\,\text{s}^{-2}$ and $\dot{v}_{\text{max}} = 5.0\,\text{m}\,\text{s}^{-2}$.

The velocities of the vehicles are sampled uniformly as $vel_{\mathrm{R}} \sim \mathcal{U}(30\,\mathrm{km\,h^{-1}}, 30.1\,\mathrm{km\,h^{-1}})$, $vel_{\mathrm{A}} \sim \mathcal{U}(32\,\mathrm{km\,h^{-1}}, 32.1\,\mathrm{km\,h^{-1}})$ and $vel_{\mathrm{F}} \sim \mathcal{U}(35\,\mathrm{km\,h^{-1}}, 35.1\,\mathrm{km\,h^{-1}})$. The position of the rear vehicle is also slightly varied uniformly with $s_0 \sim \mathcal{U}(0\,\mathrm{m}, 0.1\,\mathrm{m})$.

To generate the joint distributions $f(\mathrm{acc}_{\mathrm{IDM}}, T_{\mathrm{desired}})$, the desired time headway T_{desired} is varied in steps of 0.01 s between 0.0 s and 4.0 s. For each value of T_{desired}, for each relative distance to the rear vehicle $s = s_0 + \Delta s, \Delta s = \{-5, 0, 5\}[\mathrm{m/s}]$, and for each of the two cases that either V_{A} or V_{F} are set as leading vehicle, the output of the IDM model (cf. App. A.1) is collected for 1000 velocity samples.

A.3. Derivation of Sample Complexity of SBGs

Given that $|\Theta_{-i}| = K^{N_{-i}}$ and $|A_{-i}| \approx (|\mathcal{B}|/K)^{N_{-i}}$, one obtains

$$\begin{aligned} |\Theta_{-i}| \cdot |A_{-i}|^{T_{\mathrm{P}}} &\approx K^{N_{-i}} \cdot \left[\left(\frac{|\mathcal{B}|}{K} \right)^{N_{-i}} \right]^{T_{\mathrm{P}}} = \\ &= K^{N_{-i}} \cdot |\mathcal{B}|^{N_{-i} T_{\mathrm{P}}} \cdot K^{-N_{-i} T_{\mathrm{P}}} = \\ &= |\mathcal{B}|^{N_{-i} T_{\mathrm{P}}} \cdot K^{N_{-i} - N_{-i} T_{\mathrm{P}}} \end{aligned} \tag{A.3}$$

A.4. Derivation of Sample Complexity of RSBGs

Given that $|\Theta_{-i}| = K^{N_{-i}}$ and $|A_k| \approx |\mathcal{B}|/K$, one obtains

$$\begin{aligned} |\Theta_{-i}| \cdot |A_k|^{T_{\mathrm{P}}} &\approx K^{N_{-i}} \cdot \left[\frac{|\mathcal{B}|}{K} \right]^{T_{\mathrm{P}}} = \\ &= K^{N_{-i}} \cdot |\mathcal{B}|^{T_{\mathrm{P}}} \cdot K^{-T_{\mathrm{P}}} = \\ &= |\mathcal{B}|^{T_{\mathrm{P}}} K^{N_{-i} - T_{\mathrm{P}}} \end{aligned} \tag{A.4}$$

A.5. Traffic Parameters of the Evaluation

		Sampling range		Behavior space boundary	
Section	Scenario	Δs	vel_j [m/s]	s_{min} [m]	v_{desired} [m/s]
6.2.1, 6.3, 6.4	Freeway enter	[15, 25]	[8, 14]	[2.0, 2.5]	[8, 14]
	Left turn	[30, 35] (top lane) [15, 30] (bottom lane)	[8, 14]	[4.0, 4.5]	[8, 14]
6.2.2	Freeway enter	[5, 10]	[13, 14]	[0.5, 1.0]	[13, 14]
	Left turn	[15, 20] (top lane) [15, 20] (bottom lane)	[13, 14]	[2.0, 2.5]	[13, 14]
6.2.3	Freeway enter	[8, 12]	[8, 14]	[0.5, 1.0]	[8, 14]
	Left turn	[30, 35] (top lane) [15, 30] (bottom lane)	[8, 14]	[2.0, 2.5]	[8, 14]
6.2.4	Freeway enter	[5, 10]	[8, 14]	[0.5, 1.0]	[8, 14]
	Left turn	[15, 20] (top lane) [15, 20] (bottom lane)	[8, 14]	[2.0, 2.5]	[8, 14]

Table A.1.: Traffic parameters used in the evaluation.

The performance characteristics the RSBG and RC-RSBG planners become especially evident under certain traffic conditions. Therefore, the scenario parameters affecting traffic density and driver aggressiveness are adapted depending on the evaluation. The changes in parameters between evaluations are given in Tab. A.1. The initial scenario state is given by sampling longitudinal distances between vehicles Δs and velocities of other vehicles vel_j. The behavior space boundaries are partly adapted. The acceleration limits of the other drivers are $\dot{v}_{\text{lim},+}$=$-5\,\text{m}\,\text{s}^{-2}$ and $\dot{v}_{\text{lim},+}$=$5\,\text{m}\,\text{s}^{-2}$ in freeway enter. In the left turn scenario, larger braking of other drivers is allowed with $\dot{v}_{\text{lim},+}$=$-8\,\text{m}\,\text{s}^{-2}$. The larger s_{min} and deceleration in left turn compared to freeway enter enabled oncoming vehicles to brake such that enough space is left for the ego vehicle to cross the intersection avoiding the vehicles blocking each other. This parameterization served the purpose of the evaluation in this thesis and allowed to employ the IDM, being a car-following model also in an intersection scenario. The initial ego velocity is zero in the left turn scenario and sampled uniformly from [8, 14] [m/s] in freeway enter.

A.6. Scenario Examples for RSBG and Baseline Planners

Sec. 6.2.4 quantitatively compares the RSBG planner to non-belief-based planning approaches. Fig. A.1 and Fig. A.2 depict, for the two scenario types, how the evaluated planner variants executed a scenario when starting from the exact same initial scenario conditions.

The RSBG planner is the fastest in completing the freeway enter scenario. It completes the left turn scenario directly after the MDP planner due to more cautiously approaching the vehicle on the turning lane at time $t = 5.0\,\text{s}$. The RMDP shows similar behavior in left turn and more conservative behavior in freeway enter than the RSBG planner. The Cooperative planner acts cautiously in freeway enter and provokes a collision in left turn at $t = 4.0\,\text{s}$.

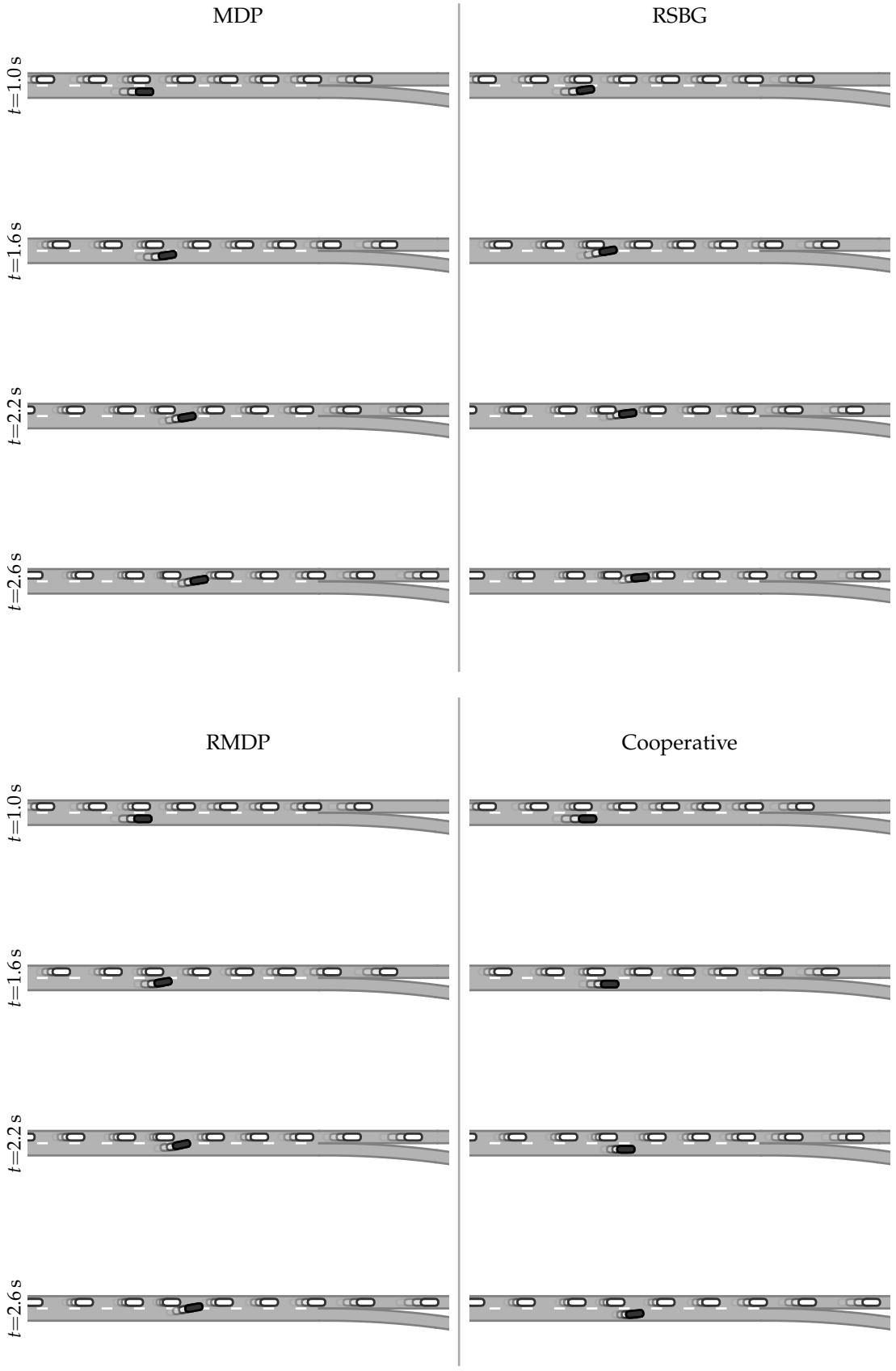


Figure A.1.: Comparison of RSBG to baseline planners in freeway entering for $\mathsf{N}_{\text{iters}} = 1k$.

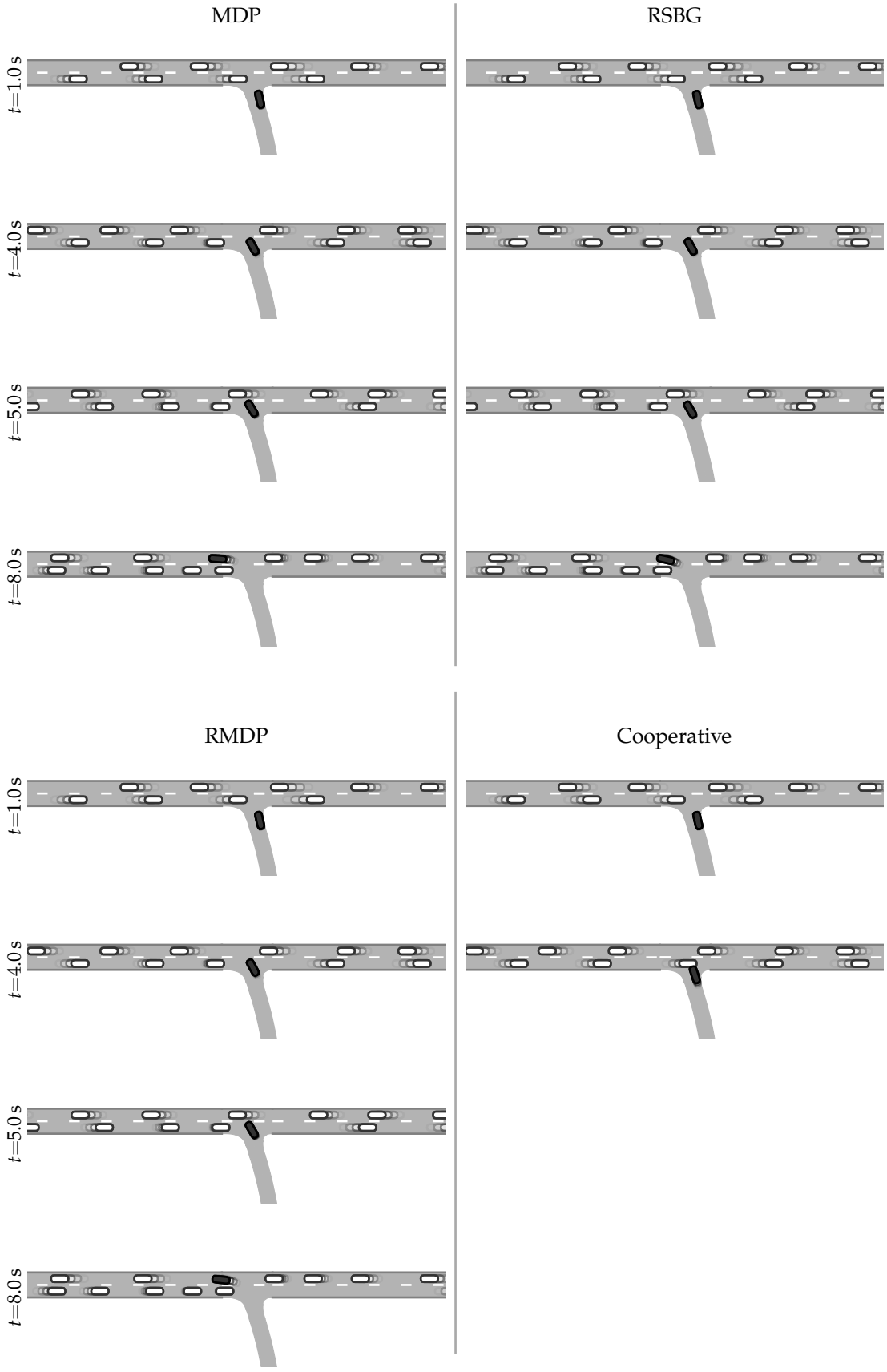


Figure A.2.: Comparison of RSBG to baseline planners in left turn for $N_{\text{iters}} = 20k$.

A.7. Examples of Left Turning with the RC-RSBG Planner

This section extends the qualitative analysis of Sec. 6.3.1. Fig. A.3 and A.4 depict executed left turn scenarios for risk level $\beta = 0.1$ and $\beta = 0.6$. At lower risk, the ego vehicle cautiously drives into the intersection and lets an oncoming car pass at the top lane before turning left. At a higher risk level, the ego vehicle more aggressively drives into the intersection and, thereby, takes the right of way of the oncoming car on the top lane. The behavior observed with risk level $\beta = 0.1$ is sometimes taken by humans to cross a crowded intersection. Thereby, the human driver does not cross the intersection in a single pass. Instead, it slowly enters to make the oncoming vehicles at the bottom lane brake. Then, while occupying the bottom lane, it waits until a gap opens up on the top lane to finish the turning maneuver. This two-step passing arises only by adjusting the risk level to $\beta = 0.1$. In contrast, for $\beta = 0.6$ an aggressive, unsafe behavior occurs, in which the ego vehicle performs a single maneuver without including an intermediate braking step.

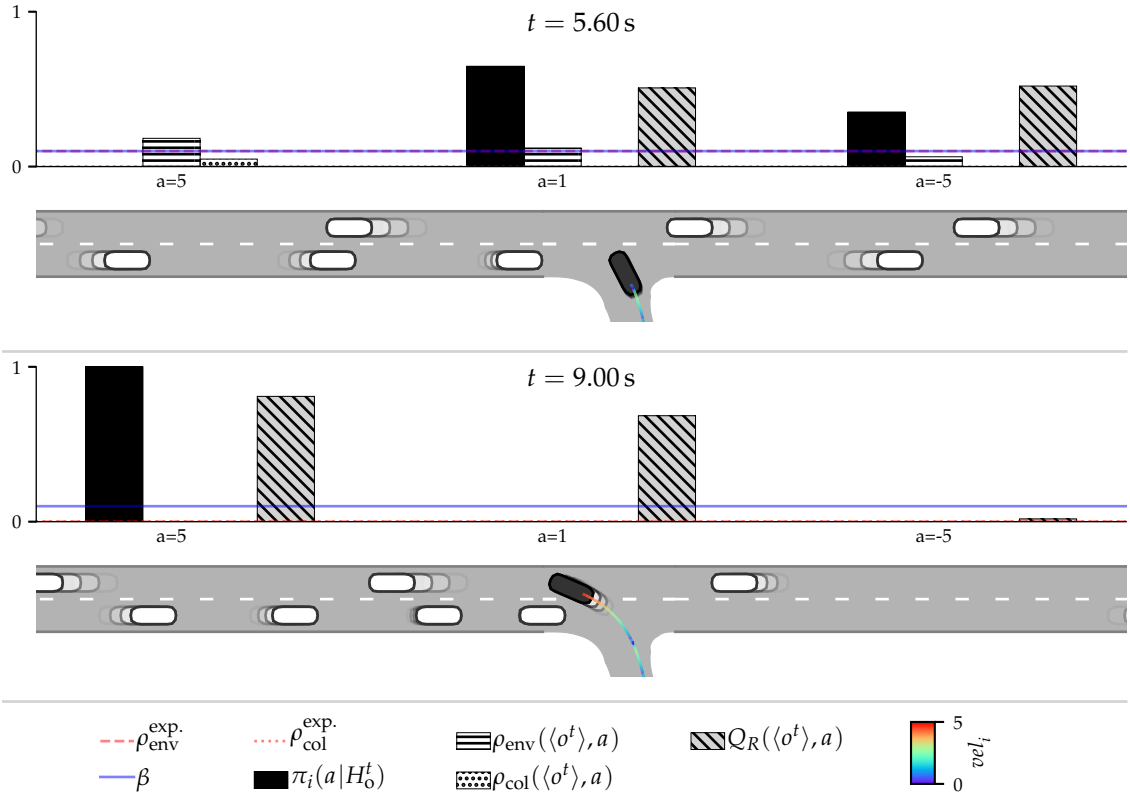


Figure A.3.: Analysis of the RC-RSBG planner's risk-constrained stochastic policy at risk level $\beta = 0.1$. The planned stochastic policy π_i balances action-risk estimates, $\rho_{\mathrm{env}}(\langle o^t \rangle, a_i)$ and $\rho_{\mathrm{col}}(\langle o^t \rangle, a_i)$ yielding an expected envelope risk $\rho_{\mathrm{env}}^{\mathrm{exp.}}$ fulfilling the risk constraint $\beta = 0.1$ while the expected planned collision risk $\rho_{\mathrm{col}}^{\mathrm{exp.}}$ is close to zero and higher returns $Q_R(\langle H_{\mathrm{o}} \rangle, a_i)$ are preferred. A lower risk level results in the ego vehicle passing the intersection in two steps, by first slowing entering the intersection and then giving right of way to the oncoming vehicle on the top lane.

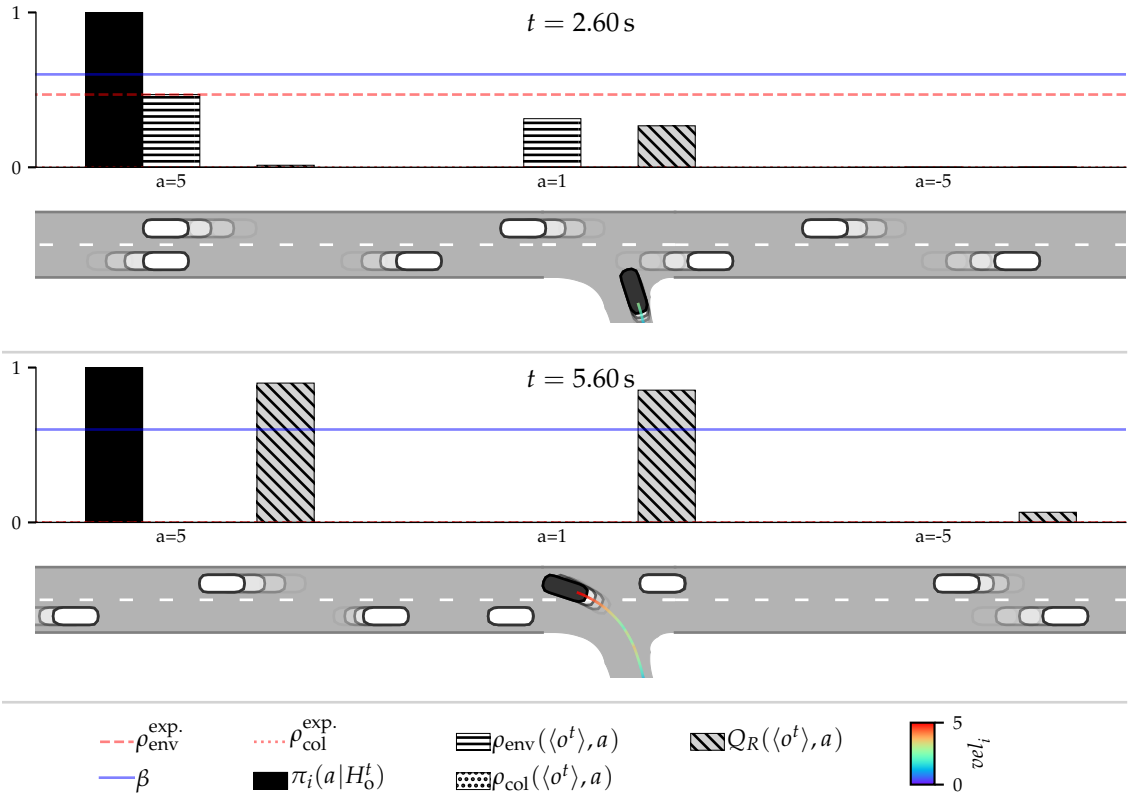


Figure A.4.: Analysis of the RC-RSBG planner's risk-constrained stochastic policy at risk level $\beta = 0.6$. As presented in Fig. A.3, yet, a higher risk level results in the ego vehicle performing an aggressive, unsafe maneuver taking the right of way from the oncoming vehicle on the top lane.

A.8. Experiment Setup for Restricting the Planning Time

The experiments run in distributed manner on an Intel® Xeon® CPU server with 50 cores @ 2.40GHz and 264Gb of memory. Slightly different experiment outcomes can arise when limiting the available planning time to $\mathsf{T}_{\text{search}} = 200\,\text{ms}$ and repeating experiments with the same planner configurations, scenario database, and equal random seeds. This nondeterminism is due to variations in the actual processing available to the planning algorithms in the different runs. Each experiment is repeated three times with the same conditions. Then, the analysis employs the experiment run with the *highest collision percentage*. This process shall mitigate that nondeterminism leads to a misinterpretation of the results.

Further, the parallelized implementation does not run multiple searches synchronously, e.g., using multiple parallel threads. Instead, all searches are run sequentially for the maximum allowed processing time to prevent other factors, e.g., the memory bandwidth available to each thread, influencing the results in a non-deterministic manner.

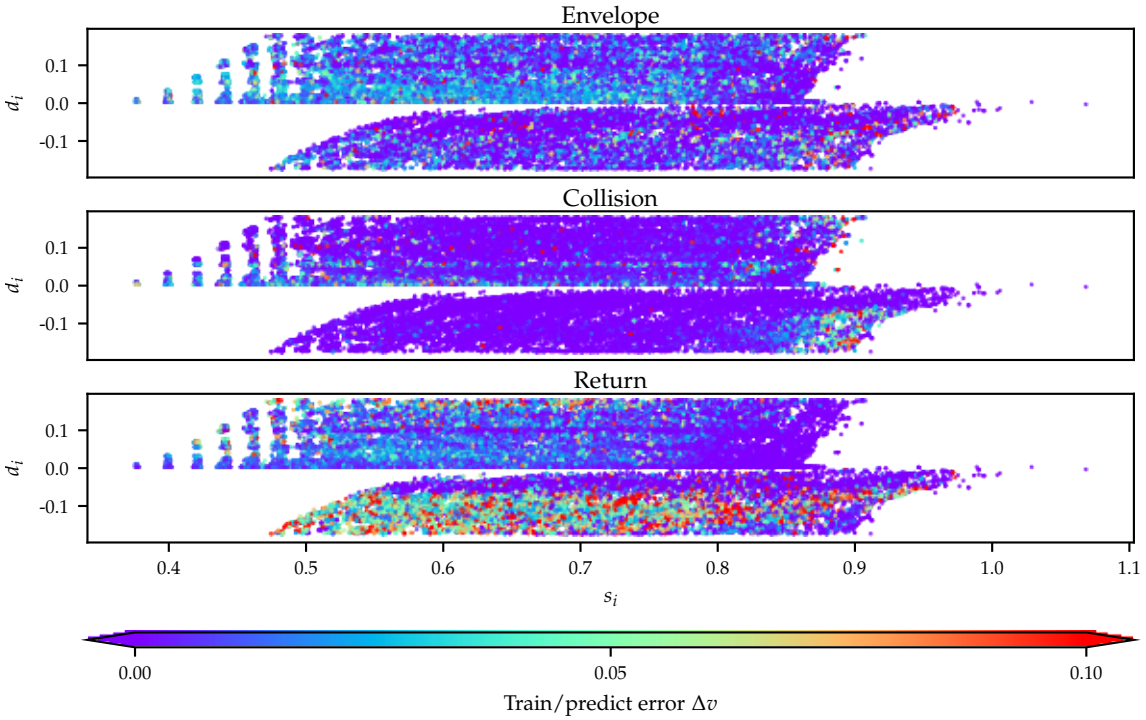


Figure A.5.: Absolute prediction errors in freeway enter. The errors are shown over the longitudinal ego coordinate s_i and lateral ego coordinate d_i for the combined test and train dataset and averaged over all other state features and actions.

A.9. Parameters and Results of Experience Generation and Training

For the data generation process, the number of initial scenario states is set to $M = 100$ for freeway enter and, due to the higher allowed scenario duration, reduced to $M = 30$ for left turn. The parameter maximum simulated scenario steps N_S is set to the respective maximum allowed scenario duration (cf. Sec. 6.1.2). The number of search iterations for state collection is set to $\mathrm{N}_{\mathrm{iters}}^{\mathrm{col}} = 100$ while the number of iterations to calculate the experience values is set to $\mathrm{N}_{\mathrm{iters}}^{\mathrm{est}} = 1000$. The number of other agents considered in the feature representation is similar to the number of agents considered by the RC-RSBG planner (cf. Sec. 6.1.4). The resulting data sets of both scenarios are limited to the size 10^5.

The supervised learning with batch stochastic gradient descent applies an Adam optimizer without weight decay with a learning rate of 0.001, a batch size of 128, and a train to test ratio of 0.9. The parameters were found using grid-search hyper-parameter optimization. Training runs for 10^6 steps with checkpointing every 5k steps. The checkpoint with the lowest test loss

Scenario	Loss/Test	Loss/Train	Abs. Diff/Collision	Abs. Diff/Envelope	Abs. Diff/Return
Freeway enter	0.00187	0.00181	0.01165 ± 0.02739	0.01835 ± 0.03213	0.01687 ± 0.02964
Left turn	0.08576	0.08673	0.14713 ± 0.26042	0.14935 ± 0.22531	0.17799 ± 0.13003

Table A.2.: Quantitative experience training results.

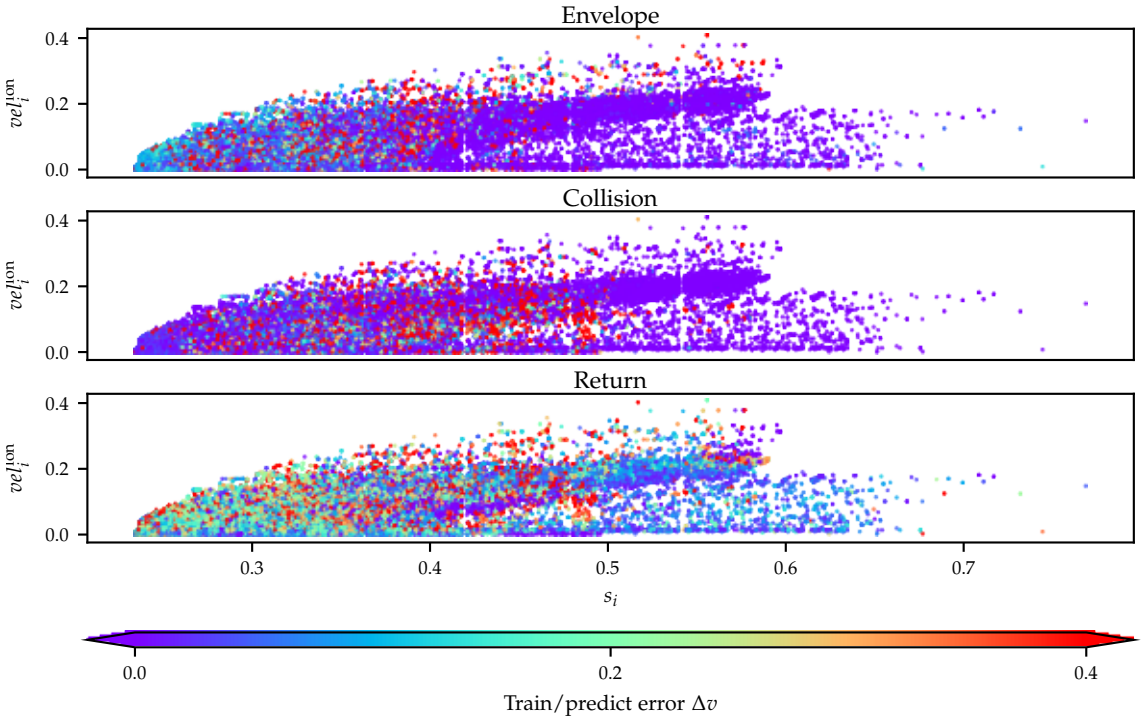


Figure A.6.: Absolute prediction errors in left turn. The errors are shown over the longitudinal ego coordinate s_i and velocity vel_i^{lon} for the combined test and train dataset and averaged over all other state features and actions.

is selected as the final network of each scenario type respectively. Tab. A.2 gives the resulting train and test losses and the mean absolute errors calculated individually for the different value types over the test data. Additionally, the mean absolute value differences are given for freeway enter over longitudinal and lateral ego position in Fig. A.5, and for left turn over longitudinal ego position and velocity in Fig. A.6 for the whole data set.

Analyzing the training results reveals that the prediction errors are significantly more prominent for the left turn scenario. On the one hand, generalizing is impeded in the left turn scenario since the larger number of considered traffic participants increases the dimension of the input representation. Further, the feature representation based on Frenet coordinates may be suboptimal to represent traffic at an intersection. Lastly, a more complex structure of the value functions may arise in the left turn scenario since long-term predictions play a more critical role. Learning such value functions is impeded and yields more significant prediction errors.

Abbreviations

SBG	Stochastic Bayesian Game. 6, 8, 25–31, 33, 40–44, 46, 89, 96, 97, 117, , 136–138
MDP	Markov Decision Process. 17, 18, 21, 42, 46, 58, 62, 72, 89, 90, 100, 101, 105, 114, 125, 133
RMDP	Robust Markov Decision Process. 7, 42, 89, 90, 100, 101, 105, 114, 125
RSBG	Robust Stochastic Bayesian Game. 8, 20, 25, 30, 40, 42–44, 46–51, 55–61, 63–65, 83, 87, 89, 90, 92, 93, 95–101, 105–110, 112, 121, 125–127, 136–139, 141
POMDP	Partially Observable Markov Decision Process. 4, 7, 17–21, 23, 31, 45, 46, 51, 58, 59, 62, 63, 119, 121, 133
CC-POMDP	Chance-Constrained POMDP. 58,
I-POMDP	Interactive POMDP. 31,
RC-RSBG	Risk-Constrained Robust Stochastic Bayesian Game. 6–9, 21, 51, 56–66, 69, 71, 73–76, 78, 80–83, 87–92, 102–119, 121, 122, 125, 128–130, 136, 138, 139, 141
QMDP	QMDP. 45, 47, 48, 63,
C-MDP	Constrained MDP. 58,
C-POMDP	Constrained POMDP. 7, 51, 59, 60, 62–64, 119, 121
CC-MDP	Chance-Constrained MDP. 58,
NN	Neural Network. 74, 75, 80,
RRT	Rapidly Exploring Random Tree. 72,
HBA	Harsanyi Bellman Ad-Hoc. 26, 30, 40,
BAMDP	Bayes-Adaptive MDP. 18, 21, 46, 48, 72,
MCTS	Monte Carlo Tree Search. iii, 2, 7, 17, 19, 25, 30, 31, 40, 45, 46, 49, 50, 62, 63, 72, 73, 80, 81, 86, 109, 110, 117, 121, 134, 139

MPC	Model Predictive Control. 4, 17, 19,
DL	Deep Learning. 17,
RL	Reinforcement Learning. 17, 20–22,
IRL	Inverse Reinforcement Learning. 20,
DRL	Deep Reinforcement Learning. 17, 22,
SM-MCTS	Simultaneous-Move MCTS. 7–9, 19, 20, 23, 25, 30, 40, 44–47, 49, 59, 86, 89, 90, 121, , 139
MIP	Mixed Integer Programming. 19, 20,
MIQP	Mixed Integer Quadratic Programming. 19,
LP	Linear Programming. 19, 59, 62, 134
MILP	Mixed Integer LP. 19, 59,
iLQG	iterative Linear Quadratic Gaussian control. 19,
LTL	Linear Temporal Logic. 17, 21,
POMCP	Partially Observable Monte Carlo Planning. 45, 46, 58, 59, 62,
ABT	Adaptive Belief Tree. 45, 58,
DESPOT	Determinized Sparse Partially Observable Tree. 45,
BAMCP	Bayes-Adaptive Monte Carlo Planning. 46, 48, 72,
BAFA	Bayes-Adaptive planning with Function Approximation. 46,
RAO*	Risk-bounded AO*. 58,
RSS	Responsibility-Sensitive Safety. 21, 66, 67, 118,
TTC	Time-To-Collision. 20,
CSP	Collision State Probability. 4,
CEP	Collision Event Probability. 4,
AV	Autonomous Vehicle. 1–3, 5, 7, 8, 11–13, 15–20, 22, 25, 26, 28–33, 40, 41, 46, 47, 50, 63, 81, 84, 109, 111–113, 116–119, 121–123, 135, 137
IDM	Intelligent Driver Model. 29, 32, 33, 35, 36, 39, 84, 88, 89, 95, 98, 101, 113, 123–125, 137
UCT	Upper Confidence bound applied to Trees. 45, 49, 50, 64, 72, 73, 90, 92, , 136
DUCB	Decoupled Upper Confidence Bound. 45, 50,
EXP3	Exponential-Weight Algorithm for Exploration and Exploitation. 45, 50,
CFR	Counterfactual Regret Minimization. 46,

Symbols

i	Index denoting ego agent, i.e., the AV.
j	Index denoting other agents, i.e., traffic participants.
N, N_{-i}	Total number and number of other agents $N_{-i} = N - 1$.
o^t, o_i^t, o_j^t	Observable states of the environment, of the ego and another agents at time t.
$\mathcal{O}$	Space of observable environment states.
b_j^t	Behavior state of agent j at time t.
$\mathcal{B}_j^t$	Unknown behavior space of agent j at time t.
$\mathcal{B}$	Full behavior space for hypotheses design.
$\mathcal{B}^k$	Behavior space assigned to hypothesis k.
a^t, a_j^t, A	Joint action, action of agent j and action space.
x, y, α, vel	Vehicle state in global coordinate system.
d, s, α^F, vel^{lon}, vel^{lat}	Vehicle state in Frenet coordinate system.
H_{o}^t	Action-observation history up to time t.
π_i	Policy of the ego agent.
π_j	Unknown policy of other agents.
k, K	Hypotheses index and number of hypotheses.
θ^k, Θ	Hypothetical agent type and type space.
θ_{-i}, Θ_{-i}	Specific combination of agent types for all other agents in the space of possible type combinations Θ_{-i}.
π^*	Deterministic hypothetical policy to define behavior hypotheses over full behavior space.
π_{θ^k}	Stochastic policy representing behavior hypotheses k.
$\widehat{\pi}_j$	Stochastic policy defined as mixture distribution over behavior hypotheses to predict agent j.
u_i, u_j	Utility function of ego and other agents.

Q_R	Expected discounted cumulative utility of the ego agent.
$\Pr(\theta^k \mid H_o^t, j)$, $\Pr(\theta_{-i} \mid H_o^t)$	Posterior probability for hypothesis k for a single other agent j, and for a combination of types for all other agents.
τ_a, τ_{predict}	Fundamental prediction duration and prediction duration at current search depth.
$\langle H'_o \rangle$	Search tree node for action-observation history H'_o.
$\mathcal{O}$, $\mathcal{O}_{\text{RSBG}}$, $\mathcal{O}_{\text{SBG}}$	Asymptotic worst-case sample complexities of RSBG and SBG.
ρ, ρ_{env}, ρ_{col}	Violation risk, envelope violation and collision violation risk.
β	Parameter specifying the maximum allowed safety envelope violation risk of the RC-RSBG planner.
κ	Exploration parameter used in UCT and risk-constrained stochastic action selection.
υ	Tolerance parameter to increase support of stochastic ego policy based on action counts.
f, f_{envelope}, $f_{\text{collision}}$	Indicator functions returning violation of a general safety measure, and specifically the violation of a safety envelope and the occurrence of a collision.
$\mathsf{N}_{\text{iters}}$	Maximum number of allowed search iterations.
$\mathsf{T}_{\text{search}}$	Maximum number of allowed search time.

List of Figures

List of Tables

List of Algorithms

Bibliography

[1] K. Bengler, K. Dietmayer, B. Farber, M. Maurer, C. Stiller, and H. Winner. "Three Decades of Driver Assistance Systems: Review and Future Perspectives." In: *IEEE Intelligent Transportation Systems Magazine* 6.4 (2014), pp. 6–22.

[2] A. Nikitas, E.T. Njoya, and S. Dani. "Examining the Myths of Connected and Autonomous Vehicles: Analysing the Pathway to a Driverless Mobility Paradigm." In: *International Journal of Automotive Technology and Management* 19.1-2 (2019), pp. 10–30.

[3] L. Lim and A.M. Tawfik. "Estimating Future Travel Costs for Autonomous Vehicles (AVs) and Shared Autonomous Vehicles (SAVs)." In: vol. 2018-November. IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC. 2018, pp. 1702–1707.

[4] Left turn. *Google Earth 9.148.0.0. (3/23/20). Munich, Germany. 48°06'50"N 11°29'07"E. Eye Alt 544 m. GeoBasis-DE/BKG (©2009)*. http://www.earth.google.com. Oct. 2021.

[5] Merging. *Google Earth 9.148.0.0. (3/23/20). Munich, Germany. 48°10'33"N 11°35'31"E. Eye Alt 504 m. GeoBasis-DE/BKG (©2009)*. http://www.earth.google.com. Oct. 2021.

[6] K. Esterle, L. Gressenbuch, and A. Knoll. "Formalizing Traffic Rules for Machine Interpretability." In: *2020 IEEE 3rd Connected and Automated Vehicles Symposium (CAVS)*. 2020, pp. 1–7.

[7] C. Pek, P. Zahn, and M. Althoff. "Verifying the Safety of Lane Change Maneuvers of Self-Driving Vehicles Based on Formalized Traffic Rules." In: *2017 IEEE Intelligent Vehicles Symposium (IV)*. 2017, pp. 1477–1483.

[8] Y. Nishimura, A. Fujita, A. Hiromori, H. Yamaguchi, T. Higashino, A. Suwa, H. Urayama, S. Takeshima, and M. Takai. "A Study on Behavior of Autonomous Vehicles Cooperating with Manually-Driven Vehicles." In: *2019 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. 2019, pp. 212–219.

[9] I. Nastjuk, B. Herrenkind, M. Marrone, A.B. Brendel, and L.M. Kolbe. "What Drives the Acceptance of Autonomous Driving? An Investigation of Acceptance Factors from an End-User's Perspective." In: *Technological Forecasting and Social Change* 161 (2020).

[10] Francesca M. Favarò, Nazanin Nader, Sky O. Eurich, Michelle Tripp, and Naresh Varadaraju. "Examining Accident Reports Involving Autonomous Vehicles in California." In: *PLOS ONE* 12.9 (Sept. 2017), pp. 1–20.

[11] Eric R. Teoh and David G. Kidd. "Rage against the Machine? Google's Self-Driving Cars versus Human Drivers." In: *Journal of Safety Research* 63 (2017), pp. 57–60.

[12] W. Biever, L. Angell, and S. Seaman. "Automated Driving System Collisions: Early Lessons." In: *Human Factors* 62.2 (2020), pp. 249–259.

[13] Peng Liu and Zhigang Xu. "Self-Driving Vehicles: Do Their Risks Outweigh Their Benefits?" In: *HCI in Mobility, Transport, and Automotive Systems*. Ed. by Heidi Krömker. Cham: Springer International Publishing, 2019, pp. 26–34.

[14] M. Chikaraishi, D. Khan, B. Yasuda, and A. Fujiwara. "Risk Perception and Social Acceptability of Autonomous Vehicles: A Case Study in Hiroshima, Japan." In: *Transport Policy* 98 (2020), pp. 105–115.

[15] Wilko Schwarting, Javier Alonso-Mora, and Daniela Rus. "Planning and Decision-Making for Autonomous Vehicles." In: *Annual Review of Control, Robotics, and Autonomous Systems* 1.1 (May 2018), pp. 187–210.

[16] S. Lefèvre, D. Vasquez, and C. Laugier. "A Survey on Motion Prediction and Risk Assessment for Intelligent Vehicles." In: *ROBOMECH Journal* 1.1 (2014).

[17] J. Wishart, S. Como, M. Elli, B. Russo, J. Weast, N. Altekar, and E. James. "Driving Safety Performance Assessment Metrics for ADS-equipped Vehicles." In: *SAE International Journal of Advances and Current Practices in Mobility* 2.5 (2020), pp. 2881–2899.

[18] A. Pierson, W. Schwarting, S. Karaman, and D. Rus. "Learning Risk Level Set Parameters from Data Sets for Safer Driving." In: *2019 IEEE Intelligent Vehicles Symposium (IV)*. 2019, pp. 273–280.

[19] Y. Akagi and P. Raksincharoensak. "Stochastic Driver Speed Control Behavior Modeling in Urban Intersections Using Risk Potential-Based Motion Planning Framework." In: *2015 IEEE Intelligent Vehicles Symposium (IV)*. June 2015, pp. 368–373.

[20] L. Li, J. Gan, K. Zhou, X. Qu, and B. Ran. "A Novel Lane-Changing Model of Connected and Automated Vehicles: Using the Safety Potential Field Theory." In: *Physica A: Statistical Mechanics and its Applications* 559 (2020), p. 125039.

[21] N. Raju, P. Kumar, S. Arkatkar, and G. Joshi. "Determining Risk-Based Safety Thresholds through Naturalistic Driving Patterns Using Trajectory Data on Expressways." In: *Safety Science* 119 (2019), pp. 117–125.

[22] D. Iberraken, L. Adouane, and D. Denis. "Safe Autonomous Overtaking Maneuver Based on Inter-Vehicular Distance Prediction and Multi-Level Bayesian Decision-Making." In: *2018 IEEE 21st International Conference on Intelligent Transportation Systems (ITSC)*. 2018, pp. 3259–3265.

[23] Chongfeng Wei, Richard Romano, Natasha Merat, Yafei Wang, Chuan Hu, Hamid Taghavifar, Foroogh Hajiseyedjavadi, and Erwin R. Boer. "Risk-Based Autonomous Vehicle Motion Control with Considering Human Driver's Behaviour." In: *Transportation Research Part C: Emerging Technologies* 107 (2019), pp. 1–14.

[24] Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. "On a Formal Model of Safe and Scalable Self-Driving Cars." In: *CoRR* abs/1708.06374 (2017).

[25] J. Nilsson, M. Brännström, J. Fredriksson, and E. Coelingh. "Longitudinal and Lateral Control for Automated Yielding Maneuvers." In: *IEEE Transactions on Intelligent Transportation Systems* 17.5 (May 2016), pp. 1404–1414.

[26] Christian Pek, Stefanie Manzinger, Markus Koschi, and Matthias Althoff. "Using Online Verification to Prevent Autonomous Vehicles from Causing Accidents." In: *Nature Machine Intelligence* 2 (Sept. 2020), pp. 518–528.

[27] K. Leung, E. Schmerling, M. Zhang, M. Chen, J. Talbot, J.C. Gerdes, and M. Pavone. "On Infusing Reachability-Based Safety Assurance within Planning Frameworks for Human–Robot Vehicle Interactions." In: *International Journal of Robotics Research* 39.10-11 (2020), pp. 1326–1345.

[28] Dorsa Sadigh, S. Shankar Sastry, Sanjit A. Seshia, and Anca D. Dragan. "Information Gathering Actions over Human Internal State." In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2016, pp. 66–73.

[29] Dorsa Sadigh, Shankar Sastry, Sanjit A. Seshia, and Anca D. Dragan. "Planning for Autonomous Cars That Leverages Effects on Human Actions." In: *Proceedings of the Robotics: Science and Systems Conference (RSS)*. June 2016.

[30] M. Bahram, C. Hubmann, A. Lawitzky, M. Aeberhard, and D. Wollherr. "A Combined Model- and Learning-Based Framework for Interaction-Aware Maneuver Prediction." In: *IEEE Transactions on Intelligent Transportation Systems* 17.6 (2016), pp. 1538–1550.

[31] C. Martínez and F. Jiménez. "Implementation of a Potential Field-Based Decision-Making Algorithm on Autonomous Vehicles for Driving in Complex Environments." In: *Sensors (Switzerland)* 19.15 (2019), p. 3318.

[32] Christian Friedrich Pek. "Provably Safe Motion Planning for Autonomous Vehicles Through Online Verification." PhD thesis. München: Technische Universität München, 2020.

[33] X. Xu, X. Wang, X. Wu, O. Hassanin, and C. Chai. "Calibration and Evaluation of the Responsibility-Sensitive Safety Model of Autonomous Car-Following Maneuvers Using Naturalistic Driving Study Data." In: *Transportation Research Part C: Emerging Technologies* 123 (2021), p. 102988.

[34] A. Rodionova, I. Alvarez, M.S. Elli, F. Oboril, J. Quast, and R. Mangharam. "How Safe Is Safe Enough? Automatic Safety Constraints Boundary Estimation for Decision-Making in Automated Vehicles." In: *2020 IEEE Intelligent Vehicles Symposium (IV)*. 2020, pp. 1457–1464.

[35] Matthew Brown, Joseph Funke, Stephen Erlien, and J. Christian Gerdes. "Safe Driving Envelopes for Path Tracking in Autonomous Vehicles." In: *Control Engineering Practice* 61 (Apr. 2017), pp. 307–316.

[36] P. Trautman and A. Krause. "Unfreezing the Robot: Navigation in Dense, Interacting Crowds." In: *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2010, pp. 797–803.

[37] Julian Bernhard and Alois Knoll. "Risk-Constrained Interactive Safety under Behavior Uncertainty for Autonomous Driving." In: *2021 IEEE Intelligent Vehicles Symposium (IV)*. 2021, pp. 63–70.

[38] C. M. Hruschka, M. Schmidt, D. Töpfer, and S. Zug. "Uncertainty-Adaptive, Risk Based Motion Planning in Automated Driving." In: *2019 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*. 2019, pp. 1–7.

[39] X. Huang, A. Jasour, M. Deyo, A. Hofmann, and B. C. Williams. "Hybrid Risk-Aware Conditional Planning with Applications in Autonomous Vehicles." In: *2018 IEEE Conference on Decision and Control (CDC)*. Dec. 2018, pp. 3608–3614.

[40] M. -Y. Yu, R. Vasudevan, and M. Johnson-Roberson. "Risk Assessment and Planning with Bidirectional Reachability for Autonomous Driving." In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. 2020, pp. 5363–5369.

[41] M. Yu, R. Vasudevan, and M. Johnson-Roberson. "Occlusion-Aware Risk Assessment for Autonomous Driving in Urban Environments." In: *IEEE Robotics and Automation Letters* 4.2 (2019), pp. 2235–2241.

[42] J. I. Ge, B. Schürmann, R. M. Murray, and M. Althoff. "Risk-Aware Motion Planning for Automated Vehicle among Human-Driven Cars." In: *2019 American Control Conference (ACC)*. 2019, pp. 3987–3993.

[43] Anirudha Majumdar and Marco Pavone. "How Should a Robot Assess Risk? Towards an Axiomatic Theory of Risk in Robotics." In: *CoRR* abs/1710.11040 (2017).

[44] J. Müller and M. Buchholz. "A Risk and Comfort Optimizing Motion Planning Scheme for Merging Scenarios*." In: *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. 2019, pp. 3155–3161.

[45] L. Claussmann, M. O'Brien, S. Glaser, H. Najjaran, and D. Gruyer. "Multi-Criteria Decision Making for Autonomous Vehicles Using Fuzzy Dempster-Shafer Reasoning." In: *2018 IEEE Intelligent Vehicles Symposium (IV)*. June 2018, pp. 2195–2202.

[46] A. Philipp and D. Goehring. "Analytic Collision Risk Calculation for Autonomous Vehicle Navigation." In: *2019 International Conference on Robotics and Automation (ICRA)*. 2019, pp. 1744–1750.

[47] T. Puphal, M. Probst, and J. Eggert. "Probabilistic Uncertainty-Aware Risk Spot Detector for Naturalistic Driving." In: *IEEE Transactions on Intelligent Vehicles* 4.3 (2019), pp. 406–415.

[48] F. Damerow and J. Eggert. "Balancing Risk against Utility: Behavior Planning Using Predictive Risk Maps." In: *2015 IEEE Intelligent Vehicles Symposium (IV)*. Vol. 2015-August. 2015, pp. 857–864.

[49] Jennifer S. Mindell, Deborah Leslie, and Malcolm Wardlaw. "Exposure-Based, 'Like-for-Like' Assessment of Road Safety by Travel Mode Using Routine Health Data." In: *PLOS ONE* 7.12 (Dec. 2012), pp. 1–10.

[50] Constantin Hubmann. "Belief State Planning for Autonomous Driving: Planning with Interaction, Uncertain Prediction and Uncertain Perception." PhD thesis. Karlsruher Institut für Technologie (KIT), 2020.

[51] Y. Wang, Y. Ren, S. Elliott, and W. Zhang. "Enabling Courteous Vehicle Interactions through Game-Based and Dynamics-Aware Intent Inference." In: *IEEE Transactions on Intelligent Vehicles* 5.2 (2020), pp. 217–228.

[52] David Lenz, Tobias Kessler, and Alois Knoll. "Tactical Cooperative Planning for Autonomous Highway Driving Using Monte-Carlo Tree Search." In: *2016 IEEE Intelligent Vehicles Symposium (IV)*. 2016, pp. 447–453.

[53] Karl Kurzer, Chenyang Zhou, and Johann Marius Zöllner. "Decentralized Cooperative Planning for Automated Vehicles with Hierarchical Monte Carlo Tree Search." In: *2018 IEEE Intelligent Vehicles Symposium (IV)*. June 2018, pp. 529–536.

[54] Y. Lu and M. Kamgarpour. "Safe Mission Planning under Dynamical Uncertainties." In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2020, pp. 2209–2215.

[55] Constantin Hubmann, Jens Schulz, Julian Löchner, and Darius Burschka. "A Belief State Planner for Interactive Merge Maneuvers in Congested Traffic." In: *2018 IEEE 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 1617–1624.

[56] Haoyu Bai, Shaojun Cai, Nan Ye, David Hsu, and Wee Sun Lee. "Intention-Aware Online POMDP Planning for Autonomous Driving in a Crowd." In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 454–460.

[57] Maxime Bouton, Jesper Karlsson, Alireza Nakhaei, Kikuo Fujimura, Mykel J. Kochenderfer, and Jana Tumova. "Reinforcement Learning with Probabilistic Guarantees for Autonomous Driving." In: *Workshop on Safety, Risk and Uncertainty in Reinforcement Learning, Conference on Uncertainty in Artifical Intelligence (UAI)*. 2018.

[58] Stefano V. Albrecht, Jacob W. Crandall, and Subramanian Ramamoorthy. "Belief and Truth in Hypothesised Behaviours." In: *Artificial Intelligence* 235 (June 2016), pp. 63–94.

[59] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2006.

[60] T. Kessler, J. Bernhard, M. Buechel, K. Esterle, P. Hart, D. Malovetz, M. Truong Le, F. Diehl, T. Brunner, and A. Knoll. "Bridging the Gap between Open Source Software and Vehicle Hardware for Autonomous Driving." In: *2019 IEEE Intelligent Vehicles Symposium (IV)*. 2019, pp. 1612–1619.

[61] G. Velasco-Hernandez, D.J. Yeong, J. Barry, and J. Walsh. "Autonomous Driving Architectures, Perception and Data Fusion: A Review." In: *2020 IEEE 16th International Conference on Intelligent Computer Communication and Processing (ICCP)*. 2020, pp. 315–321.

[62] Sagar Behere and Martin Törngren. "A Functional Reference Architecture for Autonomous Driving." In: *Information and Software Technology* 73 (2016), pp. 136–150.

[63] Michael Montemerlo, Jan Becker, Suhrid Bhat, Hendrik Dahlkamp, Dmitri Dolgov, Scott Ettinger, Dirk Haehnel, Tim Hilden, Gabe Hoffmann, Burkhard Huhnke, et al. "Junior: The Stanford Entry in the Urban Challenge." In: *Journal of field Robotics* 25.9 (2008), pp. 569–597.

[64] Chris Urmson, J. Andrew Bagnell, Christopher R. Baker, Martial Hebert, Alonzo Kelly, Raj Rajkumar, Paul E. Rybski, Sebastian Scherer, Reid Simmons, Sanjiv Singh, et al. *Tartan Racing: A Multi-Modal Approach to the Darpa Urban Challenge*. Tech. rep. Robotics Institute, Carnegie Mellon University, DARPA Grand Challenge Tech Report, Apr. 2007.

[65] Andreas Geiger, Martin Lauer, Frank Moosmann, Benjamin Ranft, Holger Rapp, Christoph Stiller, and Julius Ziegler. "Team AnnieWAY's Entry to the 2011 Grand Cooperative Driving Challenge." In: *IEEE Transactions on Intelligent Transportation Systems* 13.3 (Sept. 2012), pp. 1008–1017.

[66] Chris Urmson, Joshua Anhalt, Drew Bagnell, Christopher Baker, Robert Bittner, et al. "Autonomous Driving in Urban Environments: Boss and the Urban Challenge." In: *Journal of Field Robotics* 25.8 (Aug. 2008), pp. 425–466.

[67] J. M. Wille, F. Saust, and M. Maurer. "Stadtpilot: Driving Autonomously on Braunschweig's Inner Ring Road." In: *2010 IEEE Intelligent Vehicles Symposium (IV)*. 2010, pp. 506–511.

[68] Hermann Winner, Stephan Hakuli, Felix Lotz, and Christina Singer. *Handbook of Driver Assistance Systems*. Cham: Springer International Publishing, 2016.

[69] Klaus Dietmayer. "Predicting of Machine Perception for Automated Driving." In: *Autonomous Driving: Technical, Legal and Social Aspects*. Ed. by Markus Maurer, J. Christian Gerdes, Barbara Lenz, and Hermann Winner. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 407–424.

[70] Frank Dierkes, Karl-Heinz Siedersberger, and Markus Maurer. "Corridor Selection under Semantic Uncertainty for Autonomous Road Vehicles." In: *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. 2018, pp. 505–512.

[71] Julius Ziegler, Philipp Bender, Thao Dang, and Christoph Stiller. "Trajectory Planning for Bertha—A Local, Continuous Method." In: *2014 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2014, pp. 450–457.

[72] Jean-Paul Laumond. *Robot Motion Planning and Control*. Vol. 229. Lecture Notes in Control and Information Sciences. London Berlin: Springer, 1998.

[73] Moritz Werling, Lutz Groell, and Georg Bretthauer. "Invariant Trajectory Tracking with a Full-Size Autonomous Road Vehicle." In: *IEEE Transactions on Robotics* 26 (Sept. 2010), pp. 758–765.

[74] Jesse Levinson, Jake Askeland, Jan Becker, Jennifer Dolson, David Held, et al. "Towards Fully Autonomous Driving: Systems and Algorithms." In: *2011 IEEE Intelligent Vehicles Symposium (IV)*. 2011, pp. 163–168.

[75] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, et al. "End to End Learning for Self-Driving Cars." In: *CoRR* abs/1604.07316 (2016).

[76] Ahmad Sallab, Mohammed Abdou, Etienne Perot, and Senthil Yogamani. "Deep Reinforcement Learning Framework for Autonomous Driving." In: *Electronic Imaging* 2017 (Jan. 2017), pp. 70–76.

[77] Shai Shalev-Shwartz and Amnon Shashua. "On the Sample Complexity of End-to-End Training vs. Semantic Abstraction Training." In: *CoRR* abs/1604.06915 (2016).

[78] Christos Katrakazas, Mohammed Quddus, Wen-Hua Chen, and Lipika Deka. "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.

[79] Brian Paden, Michal Cáp, Sze Zheng Yong, Dmitry S. Yershov, and Emilio Frazzoli. "A Survey of Motion Planning and Control Techniques for Self-driving Urban Vehicles." In: *CoRR* abs/1604.07446 (2016).

[80] P. Bender, Ö. Ş. Taş, J. Ziegler, and C. Stiller. "The Combinatorial Aspect of Motion Planning: Maneuver Variants in Structured Environments." In: *2015 IEEE Intelligent Vehicles Symposium (IV)*. 2015, pp. 1386–1392.

[81] Ömer Sahin Tas, Felix Hauser, and Christoph Stiller. "Decision-Time Postponing Motion Planning for Combinatorial Uncertain Maneuvering." In: *2018 IEEE 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 2419–2425.

[82] Roland Philippsen, Sascha Kolski, Kristijan Macek, and Roland Siegwart. "Path Planning, Replanning and Execution for Autonomous Driving in Urban and Offroad Environments." In: *Workshop on Planning, Perception and Navigation for Intelligent Vehicles, Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (Jan. 2007).

[83] Martin Friedl, Adrian Hupka, and Goerg Tanzmeister. "Vollautomatisiertes Valet Parking: Funktions- Und Planungsarchitektur." In: *Uni-DAS e.V. Workshop Fahrerassistenz*. Walting, June 2015.

[84] Xingxing Du, Xiaohui Li, Daxue Liu, and Bin Dai. "Path Planning for Autonomous Vehicles in Complicated Environments." In: *2016 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*. IEEE, 2016, pp. 1–7.

[85] Constantin Hubmann, Michael Aeberhard, and Christoph Stiller. "A Generic Driving Strategy for Urban Environments." In: *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2016, pp. 1010–1016.

[86] Wenda Xu, Jia Pan, Junqing Wei, and John M. Dolan. "Motion Planning under Uncertainty for On-Road Autonomous Driving." In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 2507–2512.

[87] Christian Pek and Matthias Althoff. "Computationally Efficient Fail-safe Trajectory Planning for Self-driving Vehicles Using Convex Optimization." In: *2018 IEEE 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 1447–1454.

[88] Dennis Fassbender, Benjamin C. Heinrich, and Hans-Joachim Wuensche. "Motion Planning for Autonomous Vehicles in Highly Constrained Urban Environments." In: *Intelligent Robots and Systems*. IEEE, 2016, pp. 4708–4713.

[89] Moritz Werling. *Ein neues Konzept für die Trajektoriengenerierung und -stabilisierung in zeitkritischen Verkehrsszenarien*. Schriftenreihe des Instituts für Angewandte Informatik / Automatisierungstechnik an der Universität Karlsruhe (TH) 34. Karlsruhe: KIT Scientific Publishing, 2011.

[90] W. Zhan, A.L. De Fortelle, Y.-T. Chen, C.-Y. Chan, and M. Tomizuka. "Probabilistic Prediction from Planning Perspective: Problem Formulation, Representation Simplification and Evaluation Metric." In: *2018 IEEE Intelligent Vehicles Symposium (IV)*. 2018, pp. 1150–1156.

[91] J. Schulz, C. Hubmann, N. Morin, J. Lochner, and D. Burschka. "Learning Interaction-Aware Probabilistic Driver Behavior Models from Urban Scenarios." In: *2019 IEEE Intelligent Vehicles Symposium (IV)*. 2019, pp. 1326–1333.

[92] Adelinde M. Uhrmacher and Danny Weyns. *Multi-Agent Systems: Simulation and Applications*. 1st. USA: CRC Press, Inc., 2009.

[93] Arne Kesting and Martin Treiber. "Calibrating Car-Following Models by Using Trajectory Data: Methodological Study." In: *Transportation Research Record* 2088.1 (2008), pp. 148–156.

[94] Christiaan N. Koppel, Sebastiaan M. Petermeijer, Jelle van Doornik, and David A. Abbink. "Lane Change Manoeuvre Analysis: Inter- and Intra-Driver Variability in Lane Change Behaviour." In: *Proceedings of the Driving Simulation Conference 2019 Europe VR*. Ed. by Andras Kemeny, Florent Colombet, Frédéric Merienne, and Stéphane Espié. Apr. 2019, pp. 127–134.

[95] Yanlei Gu, Yoriyoshi Hashimoto, Li-Ta Hsu, and Shunsuke Kamijo. "Motion Planning Based on Learning Models of Pedestrian and Driver Behaviors." In: *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2016, pp. 808–813.

[96] M. Schreier, V. Willert, and J. Adamy. "Bayesian, Maneuver-Based, Long-Term Trajectory Prediction and Criticality Assessment for Driver Assistance Systems." In: *2014 IEEE 17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. Oct. 2014, pp. 334–341.

[97] P. Pandey and J.V. Aghav. "Pedestrian–Autonomous Vehicles Interaction Challenges: A Survey and a Solution to Pedestrian Intent Identification." In: *Lecture Notes in Networks and Systems* 94 (2020), pp. 283–292.

[98] Sebastian Brechtel, Tobias Gindele, and Rüdiger Dillmann. "Learning Context Sensitive Behavior Models from Observations for Predicting Traffic Situations." In: *IEEE 2013 16th International Conference on Intelligent Transportation Systems (ITSC)*. Oct. 2013, pp. 1764–1771.

[99] Sebastian Brechtel, Tobias Gindele, and Rüdiger Dillmann. "Probabilistic Decision-Making under Uncertainty for Autonomous Driving Using Continuous POMDPs." In: *17th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2014, pp. 392–399.

[100] C. Hubmann, M. Becker, D. Althoff, D. Lenz, and C. Stiller. "Decision Making for Autonomous Driving Considering Interaction and Uncertain Prediction of Surrounding Vehicles." In: *2017 IEEE Intelligent Vehicles Symposium (IV)*. June 2017, pp. 1671–1678.

[101] C. Menéndez-Romero, M. Sezer, F. Winkler, C. Dornhege, and W. Burgard. "Courtesy Behavior for Highly Automated Vehicles on Highway Interchanges." In: *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, June 2018, pp. 943–948.

[102] J. Schulz, C. Hubmann, J. Lochner, and D. Burschka. "Interaction-Aware Probabilistic Behavior Prediction in Urban Environments." In: *2018 IEEE International Conference on Intelligent Robots and Systems (IROS)*. 2018, pp. 3999–4006.

[103] Enric Galceran, Alexander G. Cunningham, Ryan M. Eustice, and Edwin Olson. "Multipolicy Decision-Making for Autonomous Driving via Changepoint-Based Behavior Prediction: Theory and Experiment." In: *Autonomous Robots* 41.6 (Aug. 2017), pp. 1367–1382.

[104] N. C. Volpi, Y. Wu, and D. Ognibene. "Towards Event-Based MCTS for Autonomous Cars." In: *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. Dec. 2017, pp. 420–427.

[105] S.P. Chinchali, S.C. Livingston, M. Chen, and M. Pavone. "Multi-Objective Optimal Control for Proactive Decision Making with Temporal Logic Models." In: *International Journal of Robotics Research* 38.12-13 (2019), pp. 1490–1512.

[106] Vase Jordanoska, Igor Gjurkov, and Darko Danev. "COMPARATIVE ANALYSIS OF CAR FOLLOWING MODELS BASED ON DRIVING STRATEGIES USING SIMULATION APPROACH." In: *Mobility and Vehicle Mechanics* 44 (Dec. 2018), pp. 1–11.

[107] K. Driggs-Campbell and R. Bajcsy. "Identifying Modes of Intent from Driver Behaviors in Dynamic Environments." In: *2015 18th IEEE International Conference on Intelligent Transportation Systems (ITSC)*. 2015, pp. 739–744.

[108] K. Driggs-Campbell and R. Bajcsy. "Communicating Intent on the Road through Human-Inspired Control Schemes." In: *2016 IEEE International Conference on Intelligent Robots and Systems (IROS)*. Vol. 2016-November. 2016, pp. 3042–3047.

[109] Christoph Burger and Martin Lauer. "Cooperative Multiple Vehicle Trajectory Planning Using MIQP." In: *21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 602–607.

[110] S. Bansal, A. Cosgun, A. Nakhaei, and K. Fujimura. "Collaborative Planning for Mixed-Autonomy Lane Merging." In: *2018 IEEE International Conference on Intelligent Robots and Systems (IROS)*. 2018, pp. 4449–4455.

[111] L. Sun, Wei Zhan, Ching-Yao Chan, and M. Tomizuka. "Behavior Planning of Autonomous Cars with Social Perception." In: *2019 IEEE Intelligent Vehicles Symposium (IV)*. 2019, pp. 207–213.

[112] T. Kessler and A. Knoll. "Cooperative Multi-Vehicle Behavior Coordination for Autonomous Driving." In: *2019 IEEE Intelligent Vehicles Symposium (IV)*. 2019, pp. 1953–1960.

[113] T. Kessler, K. Esterle, and A. Knoll. "Linear Differential Games for Cooperative Behavior Planning of Autonomous Vehicles Using Mixed-Integer Programming." In: *2020 IEEE Conference on Decision and Control (CDC)*. 2020, pp. 4060–4066.

[114] Karl Kurzer, Florian Engelhorn, and J. Marius Zöllner. "Decentralized Cooperative Planning for Automated Vehicles with Continuous Monte Carlo Tree Search." In: *CoRR* abs/1809.03200 (2018).

[115] Maximilian Naumann, Martin Lauer, and Christoph Stiller. "Generating Comfortable, Safe and Comprehensible Trajectories for Automated Vehicles in Mixed Traffic." In: *2018 IEEE 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 575–582.

[116] M. Bouton, A. Nakhaei, K. Fujimura, and M.J. Kochenderfer. "Cooperation-Aware Reinforcement Learning for Merging in Dense Traffic." In: *2019 IEEE International Conference on Intelligent Transportation Systems (ITSC)*. 2019, pp. 3441–3447.

[117] David Lenz. "Motion Planning for Highly-Automated Vehicles under Uncertainties and Interactions with Human Drivers." PhD thesis. München: Technische Universität München, 2018.

[118] K. Kurzer, M. Fechner, and J. M. Zöllner. "Accelerating Cooperative Planning for Automated Vehicles with Learned Heuristics and Monte Carlo Tree Search." In: *2020 IEEE Intelligent Vehicles Symposium (IV)*. 2020, pp. 1726–1733.

[119] W. Schwarting, A. Pierson, J. Alonso-Mora, S. Karaman, and D. Rus. "Social Behavior for Autonomous Vehicles." In: *Proceedings of the National Academy of Sciences of the United States of America* 116.50 (2019), pp. 2492–24978.

[120] A. Zyner, S. Worrall, and E. Nebot. "Naturalistic Driver Intention and Path Prediction Using Recurrent Neural Networks." In: *IEEE Transactions on Intelligent Transportation Systems* 21.4 (2020), pp. 1584–1594.

[121] S. Ahmed, M.N. Huda, S. Rajbhandari, C. Saha, M. Elshaw, and S. Kanarachos. "Pedestrian and Cyclist Detection and Intent Estimation for Autonomous Vehicles: A Survey." In: *Applied Sciences (Switzerland)* 9.11 (2019).

[122] H. Cheng, W. Liao, M.Y. Yang, M. Sester, and B. Rosenhahn. "MCENET: Multi-context Encoder Network for Homogeneous Agent Trajectory Prediction in Mixed Traffic." In: *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. 2020, pp. 1–8.

[123] L. Sun, W. Zhan, and M. Tomizuka. "Probabilistic Prediction of Interactive Driving Behavior via Hierarchical Inverse Reinforcement Learning." In: *2018 21st IEEE Conference on Intelligent Transportation Systems (ITSC)*. 2018, pp. 2111–2117.

[124] Bhargav Naidu Matcha, Satesh Narayana Namasivayam, Mohammad Hosseini Fouladi, K. C. Ng, Sivakumar Sivanesan, and Se Yong Eh Noum. "Simulation Strategies for Mixed Traffic Conditions: A Review of Car-Following Models and Simulation Frameworks." In: *Journal of Engineering* 2020 (2020).

[125] Mohamed Hussein and Tarek Sayed. "A Methodology for the Microscopic Calibration of Agent - Based Pedestrian Simulation Models." In: *2018 IEEE 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 3773–3778.

[126] Tirthankar Bandyopadhyay, Chong Zhuang Jie, David Hsu, Marcelo H. Ang Jr, Daniela Rus, and Emilio Frazzoli. "Intention-Aware Pedestrian Avoidance." In: *Experimental Robotics*. Springer, 2013, pp. 963–977.

[127] Yi Wang, Kok Sung Won, David Hsu, and Wee Sun Lee. "Monte Carlo Bayesian Reinforcement Learning." In: *2012 Proceedings of the 29th International Conference on Machine Learning (ICML)*. 2012, pp. 1135–1142.

[128] E. Ward, N. Evestedt, D. Axehill, and J. Folkesson. "Probabilistic Model for Interaction Aware Planning in Merge Scenarios." In: *IEEE Transactions on Intelligent Vehicles* 2.2 (2017), pp. 133–146.

[129] J. Buyer, D. Waldenmayer, N. Susmann, R. Zollner, and J.M. Zollner. "Interaction-Aware Approach for Online Parameter Estimation of a Multi-Lane Intelligent Driver Model." In: *2019 22nd IEEE International Conference on Intelligent Transportation Systems (ITSC)*. 2019, pp. 3967–3973.

[130] Z.N. Sunberg, C.J. Ho, and M.J. Kochenderfer. "The Value of Inferring the Internal State of Traffic Participants for Autonomous Freeway Driving." In: *2017 Proceedings of the American Control Conference (ACC)*. 2017, pp. 3004–3010.

[131] Carl-Johan Hoel, Krister Wolff, and Leo Laine. "Automated Speed and Lane Change Decision Making Using Deep Reinforcement Learning." In: *2018 IEEE 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 2148–2155.

[132] M. Shen, H. Hu, B. Sun, and W. Deng. "Heuristics Based Cooperative Planning for Highway On-Ramp Merge." In: *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. 2018, pp. 1266–1272.

[133] Peter Stone, Gal A. Kaminka, Sarit Kraus, and Jeffrey S. Rosenschein. "Ad Hoc Autonomous Agent Teams: Collaboration without Pre-Coordination." In: *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*. AAAI'10. AAAI Press, 2010, pp. 1504–1509.

[134] C. Paxton, V. Raman, G. D. Hager, and M. Kobilarov. "Combining Neural Networks and Tree Search for Task and Motion Planning in Challenging Environments." In: *2017 IEEE International Conference on Intelligent Robots and Systems (IROS)*. IEEE, Sept. 2017, pp. 6059–6066.

[135] W. Zhan, C. Liu, C. Chan, and M. Tomizuka. "A Non-Conservatively Defensive Strategy for Urban Autonomous Driving." In: *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. 2016, pp. 459–464.

[136] C. Burger, T. Schneider, and M. Lauer. "Interaction Aware Cooperative Trajectory Planning for Lane Change Maneuvers in Dense Traffic." In: *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. 2020, pp. 1–8.

[137] Mykel J Kochenderfer. *Decision Making under Uncertainty: Theory and Application*. MIT press, 2015.

[138] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. Second Edition. Cambridge, MA: The MIT Press, 2018.

[139] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, et al. "Human-Level Control through Deep Reinforcement Learning." In: *Nature* 518.7540 (Feb. 2015), pp. 529–533.

[140] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. "Asynchronous Methods for Deep Reinforcement Learning." In: *2016 Proceedings of the International Conference on Machine Learning (ICML)*. 2016, pp. 1928–1937.

[141] Peter Wolf, Karl Kurzer, Tobias Wingert, Florian Kuhnt, and Johann Marius Zöllner. "Adaptive Behavior Generation for Autonomous Driving Using Deep Reinforcement Learning with Compact Semantic States." In: *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 993–1000.

[142] David Isele, Akansel Cosgun, Kaushik Subramanian, and Kikuo Fujimura. "Navigating Intersections with Autonomous Vehicles Using Deep Reinforcement Learning." In: *CoRR* abs/ 1705.01196 (2017).

[143] David Isele, Reza Rahimi, Akansel Cosgun, Kaushik Subramanian, and Kikuo Fujimura. "Navigating Occluded Intersections with Autonomous Vehicles Using Deep Reinforcement Learning." In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 2034–2039.

[144] Branka Mirchevska, Christian Pek, Moritz Werling, Matthias Althoff, and Joschka Boedecker. "High-Level Decision Making for Safe and Reasonable Autonomous Lane Changing Using Reinforcement Learning." In: *2018 IEEE 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 2156–2162.

[145] Mustafa Mukadam, Akansel Cosgun, Nakhaei Alireza, and Fujimura Kikuo. "Tactical Decision Making for Lane Changing with Deep Reinforcement Learning." In: *Conference on Neural Information Processing (NIPS)*. 2017.

[146] X. Li, X. Xu, and L. Zuo. "Reinforcement Learning Based Overtaking Decision-Making for Highway Autonomous Driving." In: *2015 Sixth International Conference on Intelligent Control and Information Processing*. Nov. 2015, pp. 336–342.

[147] Branka Mirchevska, Manuel Blum, Lawrence Louis, Joschka Boedecker, and Moritz Werling. "Reinforcement Learning for Autonomous Maneuvering in Highway Scenarios." In: *11. Uni-DAS e.V. Workshop Fahrerassistenz Und Automatisiertes Fahren*. 2017.

[148] Shai Shalev-Shwartz, Nir Ben-Zrihem, Aviad Cohen, and Amnon Shashua. "Long-Term Planning by Short-Term Prediction." In: *CoRR* abs/1602.01580 (2016). arXiv: 1602.01580.

[149] Mykel J. Kochenderfer, Tim A. Wheeler, and Kyle H. Wray. *Algorithms for Decision Making*. MIT Press, 2022.

[150] Simon Ulbrich and Markus Maurer. "Probabilistic Online POMDP Decision Making for Lane Changes in Fully Automated Driving." In: *2013 IEEE 16th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2013, pp. 2063–2067.

[151] Zachary Sunberg and Mykel J. Kochenderfer. "Online Algorithms for POMDPs with Continuous State, Action, and Observation Spaces." In: *Twenty-Eighth International Conference on Automated Planning and Scheduling (ICAPS)*. 2018, pp.259–263.

[152] Zachary Sunberg and Mykel J. Kochenderfer. "Improving Automated Driving through Planning with Human Internal States." In: *CoRR* abs/2005.14549 (2020). arXiv: 2005.14549.

[153] Trong Nghia Hoang and Kian Hsiang Low. "Interactive POMDP Lite: Towards Practical Planning to Predict and Exploit Intentions for Interacting with Self-interested Agents." In: *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*. IJCAI '13. AAAI Press, 2013, pp. 2298–2305.

[154] Stefan Albrecht and Subramanian Ramamoorthy. "A Game-theoretic Model and Best-response Learning Method for Ad Hoc Coordination in Multiagent Systems." In: *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-agent Systems*. AAMAS '13. St. Paul, MN, USA: International Foundation for Autonomous Agents and Multiagent Systems, 2013, pp. 1155–1156.

[155] P. Hang, C. Lv, C. Huang, J. Cai, Z. Hu, and Y. Xing. "An Integrated Framework of Decision Making and Motion Planning for Autonomous Vehicles Considering Social Behaviors." In: *IEEE Transactions on Vehicular Technology* 69.12 (2020), pp. 14458–14469.

[156] John Mern, Anil Yildiz, Zachary Sunberg, Tapan Mukerji, and Mykel J. Kochenderfer. "Bayesian Optimized Monte Carlo Planning." In: *AAAI Conference on Artificial Intelligence (AAAI)*. Vol. 35(13). 2021, pp. 11880–11887.

[157] Klemens Esterle, Tobias Kessler, and Alois Knoll. "Optimal Behavior Planning for Autonomous Driving: A Generic Mixed-Integer Formulation." In: *2020 IEEE Intelligent Vehicles Symposium (IV)*. Oct. 2020, pp. 1914–1921.

[158] Wilko Schwarting, Alyssa Pierson, Sertac Karaman, and Daniela Rus. "Stochastic Dynamic Games in Belief Space." In: *CoRR* abs/1909.06963 (2019).

[159] Cameron B. Browne, Edward Powley, Daniel Whitehouse, Simon M. Lucas, Peter I. Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. "A Survey of Monte Carlo Tree Search Methods." In: *IEEE Transactions on Computational Intelligence and AI in Games* 4.1 (Mar. 2012), pp. 1–43.

[160] Viliam Lisy, Vojta Kovarik, Marc Lanctot, and Branislav Bosansky. "Convergence of Monte Carlo Tree Search in Simultaneous Move Games." In: *Advances in Neural Information Processing Systems 26*. Ed. by C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger. Curran Associates, Inc., 2013, pp. 2112–2120.

[161] F. Fabiani and S. Grammatico. "Multi-Vehicle Automated Driving as a Generalized Mixed-Integer Potential Game." In: *IEEE Transactions on Intelligent Transportation Systems* 21.3 (2020), pp. 1064–1073.

[162] Olivier Pietquin and Fabio Tango. "A Reinforcement Learning Approach to Optimize the Longitudinal Behavior of a Partial Autonomous Driving Assistance System." In: *Proceedings of the 20th European Conference on Artificial Intelligence*. IOS Press, 2012, pp. 987–992.

[163] Edouard Leurent, Yann Blanco, Denis V. Efimov, and Odalric-Ambrym Maillard. "Approximate Robust Control of Uncertain Dynamical Systems." In: *CoRR* abs/1903.00220 (2019).

[164] X. Huang, Sungkweon Hong, A. Hofmann, and B. Williams. "Online Risk-Bounded Motion Planning for Autonomous Vehicles in Dynamic Environments." In: *2019 International Conference on Automated Planning and Scheduling (ICAPS)*. Vol. 29(1). 2019, pp. 214–222.

[165] D. Li, Y. Wu, B. Bai, and Q. Hao. "Behavior and Interaction-Aware Motion Planning for Autonomous Driving Vehicles Based on Hierarchical Intention and Motion Prediction." In: *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. 2020.

[166] Patrick Hart and Alois Knoll. "Graph Neural Networks and Reinforcement Learning for Behavior Generation in Semantic Environments." In: *2020 IEEE Intelligent Vehicles Symposium (IV)*. 2020, pp. 1589–1594.

[167] Patrick Hart and Alois C. Knoll. "Using Counterfactual Reasoning and Reinforcement Learning for Decision-Making in Autonomous Driving." In: *CoRR* abs/2003.11919 (2020).

[168] Karl Kurzer, Christoph Hörtnagl, and J. Marius Zöllner. "Parallelization of Monte Carlo Tree Search in Continuous Domains." In: *CoRR* abs/2003.13741 (2020).

[169] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, et al. "Mastering the Game of Go with Deep Neural Networks and Tree Search." In: *Nature* 529.7587 (Jan. 2016), pp. 484–489.

[170] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, et al. "Mastering the Game of Go without Human Knowledge." In: *Nature* 550.7676 (Oct. 2017), pp. 354–359.

[171] C.-J. Hoel, K. Driggs-Campbell, K. Wolff, L. Laine, and M.J. Kochenderfer. "Combining Planning and Deep Reinforcement Learning in Tactical Decision Making for Autonomous Driving." In: *IEEE Transactions on Intelligent Vehicles* 5.2 (2020), pp. 294–305.

[172] J. Chen, C. Zhang, J. Luo, J. Xie, and Y. Wan. "Driving Maneuvers Prediction Based Autonomous Driving Control by Deep Monte Carlo Tree Search." In: *IEEE Transactions on Vehicular Technology* 69.7 (2020), pp. 7146–7158.

[173] Julian Bernhard and Alois Knoll. "Robust Stochastic Bayesian Games for Behavior Space Coverage." In: *Robotics: Science and Systems (RSS), Workshop on Interaction and Decision-Making in Autonomous-Driving*. 2020.

[174] Samuel Barrett and Peter Stone. "Cooperating with Unknown Teammates in Complex Domains: A Robot Soccer Case Study of Ad Hoc Teamwork." In: *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*. AAAI'15. AAAI Press, 2015, pp. 2010–2016.

[175] Peter Stone, Gal A. Kaminka, and Jeffrey S. Rosenschein. "Leading a Best-Response Teammate in an Ad Hoc Team." In: *Agent-Mediated Electronic Commerce. Designing Trading Strategies and Mechanisms for Electronic Markets*. Ed. by Will van der Aalst, John Mylopoulos, Norman M. Sadeh, Michael J. Shaw, Clemens Szyperski, Esther David, Enrico Gerding, David Sarne, and Onn Shehory. Vol. 59. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 132–146.

[176] Stefano V. Albrecht and Peter Stone. "Reasoning about Hypothetical Agent Behaviours and Their Parameters." In: *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*. AAMAS '17. São Paulo, Brazil: International Foundation for Autonomous Agents and Multiagent Systems, 2017, pp. 547–555.

[177] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. 3rd. Upper Saddle River, NJ, USA: Prentice Hall Press, 2009.

[178] Lucian Busoniu, Robert Babuška, and Bart De Schutter. "Multi-Agent Reinforcement Learning: An Overview." In: *Innovations in multi-agent systems and applications-1* 310 (2010), pp. 183–221.

[179] Stefano Albrecht. "Utilising Policy Types for Effective Ad Hoc Coordination in Multiagent Systems." PhD thesis. The University of Edinburgh, Nov. 2015.

[180] Dan Ariely. *Predictably Irrational: The Hidden Forces That Shape Our Decisions*. New York: Harper Perennial, 2008.

[181] Oliver Hart. *Incomplete Contracts and Control, Prize Lecture*. Dec. 2016.

[182] Finnegan Southey, Michael Bowling, Bryce Larson, Carmelo Piccione, Neil Burch, Darse Billings, and Chris Rayner. "Bayes' Bluff: Opponent Modelling in Poker." In: *Proceedings of the 21st Annual Conference on Uncertainty in Artificial Intelligence (UAI)*. 2005, pp. 550–558.

[183] Koen Hindriks and Dmytro Tykhonov. "Opponent Modelling in Automated Multi-Issue Negotiation Using Bayesian Learning." In: *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems*. Vol. 1. AAMAS '08. May 2008, pp. 331–338.

[184] Samuel Barrett, Peter Stone, and Sarit Kraus. "Empirical Evaluation of Ad Hoc Teamwork in the Pursuit Domain." In: *The 10th International Conference on Autonomous Agents and Multiagent Systems*. Vol. 2. AAMAS '11. Taipei, Taiwan, 2011, pp. 567–574.

[185] Stefano V. Albrecht and Peter Stone. "Autonomous Agents Modelling Other Agents: A Comprehensive Survey and Open Problems." In: *Artificial Intelligence* 258 (May 2018), pp. 66–95.

[186] Martin Treiber, Ansgar Hennecke, and Dirk Helbing. "Congested Traffic States in Empirical Observations and Microscopic Simulations." In: *Physical review E* 62.2 (Aug. 2000), pp. 1805–1824.

[187] Samuel Barrett, Peter Stone, Sarit Kraus, and Avi Rosenfeld. "Teamwork with Limited Knowledge of Teammates." In: *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*. AAAI'13. AAAI Press, 2013, pp. 102–108.

[188] Piotr Franciszek Orzechowski, Annika Meyer, and Martin Lauer. "Tackling Occlusions & Limited Sensor Range with Set-based Safety Verification." In: *International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 1729–1736.

[189] David Lenz, Markus Rickert, and Alois Knoll. "Heuristic Search in Belief Space for Motion Planning under Uncertainties." In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 2659–2665.

[190] Adam Bry and Nicholas Roy. "Rapidly-Exploring Random Belief Trees for Motion Planning under Uncertainty." In: *2011 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2011, pp. 723–730.

[191] Judea Pearl. "Causal Inference in Statistics: An Overview." In: *Statistics Surveys* 3 (2009), pp. 96–146.

[192] J. Andrew Bagnell, Andrew Y. Ng, and Je G. Schneider. *Solving Uncertain Markov Decision Processes*. Tech. rep. Carnegie Mellon University, 2001.

[193] Arnab Nilim and Laurent El Ghaoui. "Robust Control of Markov Decision Processes with Uncertain Transition Matrices." In: *Operations Research* 53.5 (2005), pp. 780–798.

[194] Shihui Li, Yi Wu, Xinyue Cui, Honghua Dong, Fei Fang, and Stuart Russell. "Robust Multi-Agent Reinforcement Learning via Minimax Deep Deterministic Policy Gradient." In: *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*. July 2019, pp. 4213–4220.

[195] Shiau Hong Lim, Huan Xu, and Shie Mannor. "Reinforcement Learning in Robust Markov Decision Processes." In: *Advances in Neural Information Processing Systems*. Ed. by C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger. Vol. 26. Curran Associates, Inc., 2013, pp. 701–709.

[196] Aviv Tamar, Shie Mannor, and Huan Xu. "Scaling Up Robust MDPs Using Function Approximation." In: *Proceedings of the 31st International Conference on Machine Learning*. Ed. by Eric P. Xing and Tony Jebara. Vol. 32. Proceedings of Machine Learning Research 2. PMLR, June 2014, pp. 181–189.

[197] Esther Derman, Daniel J. Mankowitz, Timothy A. Mann, and Shie Mannor. "A Bayesian Approach to Robust Reinforcement Learning." In: *Proceedings of the 35th Conference on Uncertainty in Artificial Intelligence (UAI)*. Tel Aviv, Israel, 2019.

[198] Marek Petrik and Reazul Hasan Russel. "Beyond Confidence Regions: Tight Bayesian Ambiguity Sets for Robust MDPs." In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett. Vol. 32. Curran Associates, Inc., 2019, pp. 7047–7056.

[199] Michele Aghassi and Dimitris Bertsimas. "Robust Game Theory." In: *Mathematical Programming* 107.1–2 (June 2006), pp. 231–273.

[200] Ronald Bjarnason, Alan Fern, and Prasad Tadepalli. "Lower Bounding Klondike Solitaire with Monte-Carlo Planning." In: *Proceedings of the Nineteenth International Conference on International Conference on Automated Planning and Scheduling*. ICAPS'09. Thessaloniki, Greece: AAAI Press, 2009, pp. 26–33.

[201] P. I. Cowling, E. J. Powley, and D. Whitehouse. "Information Set Monte Carlo Tree Search." In: *IEEE Transactions on Computational Intelligence and AI in Games* 4.2 (June 2012), pp. 120–143.

[202] Joelle Pineau, Geoffrey Gordon, and Sebastian Thrun. "Point-Based Value Iteration: An Anytime Algorithm for POMDPs." In: *International Joint Conference on Artificial Intelligence (IJCAI)*. Vol. 18. Aug. 2003, pp. 1025–1032.

[203] David Silver and Joel Veness. "Monte-Carlo Planning in Large POMDPs." In: *Advances in Neural Information Processing Systems 23 (NIPS)*. 2010, pp. 2164–2172.

[204] Hanna Kurniawati and Vinay Yadav. "An Online POMDP Solver for Uncertainty Planning in Dynamic Environment." In: *Robotics Research: The 16th International Symposium ISRR*. Ed. by Masayuki Inaba and Peter Corke. Cham: Springer International Publishing, 2016, pp. 611–629.

[205] Adhiraj Somani, Nan Ye, David Hsu, and Wee Sun Lee. "DESPOT: Online POMDP Planning with Regularization." In: *Advances in Neural Information Processing Systems*. Ed. by C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger. Vol. 26. Curran Associates, Inc., 2013, pp. 1772–1780.

[206] Adrien Couëtoux, Jean-Baptiste Hoock, Nataliya Sokolovska, Olivier Teytaud, and Nicolas Bonnard. "Continuous Upper Confidence Trees." In: *Learning and Intelligent Optimization*. Ed. by Carlos A. Coello Coello. Springer Berlin Heidelberg, 2011, pp. 433–445.

[207] Philippe Morere, Roman Marchant, and Fabio Ramos. "Continuous State - Action - Observation POMDPs for Trajectory Planning with Bayesian Optimisation." In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Oct. 2018, pp. 8779–8786.

[208] Sebastian Brechtel, Tobias Gindele, and Rüdiger Dillmann. "Solving Continuous POMDPs: Value Iteration with Incremental Learning of an Efficient Space Representation." In: *Journal of machine learning research : JMLR* 28.3 (2013), pp. 370–378.

[209] Haoyu Bai, David Hsu, Mykel Kochenderfer, and Wee Sun Lee. "Unmanned Aircraft Collision Avoidance Using Continuous-State POMDPs." In: *Robotics: Science and Systems VII*. Robotics: Science and Systems Foundation, June 2011.

[210] David Auger. "Multiple Tree for Partially Observable Monte-Carlo Tree Search." In: *Applications of Evolutionary Computation*. Ed. by Cecilia Di Chio, Stefano Cagnoni, Carlos Cotta, Marc Ebner, Anikó Ekárt, et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 53–62.

[211] Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. "The Nonstochastic Multiarmed Bandit Problem." In: *SIAM Journal on Computing* 32.1 (Jan. 2003), pp. 48–77.

[212] Mohammad Shafiei, Nathan Sturtevant, and Jonathan Schaeffer. "Comparing UCT versus CFR in Simultaneous Games." In: *Proceedings of the IJCAI-09 Workshop on General Game Playing (GIGA'09)*. 2009, pp. 75–82.

[213] Martin Zinkevich, Michael Johanson, Michael Bowling, and Carmelo Piccione. "Regret Minimization in Games with Incomplete Information." In: *Advances in Neural Information Processing Systems*. Ed. by J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis. Vol. 20. Curran Associates, Inc., 2008, pp. 1729–1736.

[214] Michael Johanson, Nolan Bard, Marc Lanctot, Richard Gibson, and Michael Bowling. "Efficient Nash Equilibrium Approximation through Monte Carlo Counterfactual Regret Minimization." In: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*. International Foundation for Autonomous Agents and Multiagent Systems, 2012, pp. 837–846.

[215] Michael Bowling, Neil Burch, Michael Johanson, and Oskari Tammelin. "Heads-up Limit Hold'em Poker Is Solved." In: *Science* 347.6218 (2015), pp. 145–149.

[216] Matej Moravčík, Martin Schmid, Neil Burch, Viliam Lisý, Dustin Morrill, Nolan Bard, Trevor Davis, Kevin Waugh, Michael Johanson, and Michael Bowling. "DeepStack: Expert-level Artificial Intelligence in Heads-up No-Limit Poker." In: *Science* 356.6337 (2017), pp. 508–513.

[217] Arthur Guez, David Silver, and Peter Dayan. "Efficient Bayes-adaptive Reinforcement Learning Using Sample-based Search." In: *Proceedings of the 25th International Conference on Neural Information Processing Systems (NIPS)*. Vol. 1. NIPS'12. Curran Associates Inc., 2012, pp. 1025–1033.

[218] Arthur Guez, Nicolas Heess, David Silver, and Peter Dayan. "Bayes-Adaptive Simulation-based Search with Value Function Approximation." In: *Advances in Neural Information Processing Systems 27*. Ed. by Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger. Curran Associates, Inc., 2014, pp. 451–459.

[219] Marc Ponsen, Geert Gerritsen, and Guillaume Chaslot. "Integrating Opponent Models with Monte-Carlo Tree Search in Poker." In: *Proceedings of the 3rd AAAI Conference on Interactive Decision Theory and Game Theory*. AAAIWS'10-03. AAAI Press, 2010, pp. 37–42.

[220] T. Sarratt, D. V. Pynadath, and A. Jhala. "Converging to a Player Model in Monte-Carlo Tree Search." In: *2014 IEEE Conference on Computational Intelligence and Games*. Aug. 2014, pp. 1–7.

[221] Levente Kocsis and Csaba Szepesvári. "Bandit Based Monte-Carlo Planning." In: *Machine Learning: ECML 2006*. Vol. 4212. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 282–293.

[222] Olivier Teytaud and Sébastien Flory. "Upper Confidence Trees with Short Term Partial Information." In: *Applications of Evolutionary Computation*. Ed. by Cecilia Di Chio, Stefano Cagnoni, Carlos Cotta, Marc Ebner, Anikó Ekárt, et al. Springer Berlin Heidelberg, 2011, pp. 153–162.

[223] Albert Rizaldi, Jonas Keinholz, Monika Huber, Jochen Feldle, Fabian Immler, Matthias Althoff, Eric Hilgendorf, and Tobias Nipkow. "Formalising and Monitoring Traffic Rules for Autonomous Vehicles in Isabelle/HOL." In: *Integrated Formal Methods*. Ed. by Nadia Polikarpova and Steve Schneider. Cham: Springer International Publishing, 2017, pp. 50–66.

[224] Albert Rizaldi, Fabian Immler, and Matthias Althoff. "A Formally Verified Checker of the Safe Distance Traffic Rules for Autonomous Vehicles." In: *NASA Formal Methods Symposium*. June 2016, pp. 175–190.

[225] Eitan Altman. *Constrained Markov Decision Processes*. Chapman and Hall/CRC, 1999.

[226] Tetsuro Morimura, Masashi Sugiyama, Hisashi Kashima, Hirotaka Hachiya, and Toshiyuki Tanaka. "Parametric Return Density Estimation for Reinforcement Learning." In: *CoRR* abs/1203.3497 (2012).

[227] Marc G. Bellemare, Will Dabney, and Rémi Munos. "A Distributional Perspective on Reinforcement Learning." In: *CoRR* abs/1707.06887 (2017).

[228] Will Dabney, Mark Rowland, Marc G. Bellemare, and Rémi Munos. "Distributional Reinforcement Learning with Quantile Regression." In: *CoRR* abs/1710.10044 (2017).

[229] Will Dabney, Georg Ostrovski, David Silver, and Remi Munos. "Implicit Quantile Networks for Distributional Reinforcement Learning." In: *35th International Conference on Machine Learning (ICML)*. Vol. 80. Proceedings of Machine Learning Research. Stockholmsmässan, Stockholm Sweden: PMLR, July 2018, pp. 1096–1105.

[230] Haruki Nishimura, Boris Ivanovic, Adrien Gaidon, Marco Pavone, and Mac Schwager. "Risk-Sensitive Sequential Action Control with Multi-Modal Human Trajectory Forecasting for Safe Crowd-Robot Interaction." In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2020, pp. 11205–11212.

[231] K. Chatterjee, P. Novotný, G. Pérez, J. Raskin, and Dorde Zikelic. "Optimizing Expectation with Guarantees in POMDPs." In: *AAAI Conference on Artificial Intelligence (AAAI)*. 2017.

[232] K. Chatterjee, Adrián Elgyütt, P. Novotný, and Owen Rouillé. "Expectation Optimization with Probabilistic Guarantees in POMDPs with Discounted-Sum Objectives." In: *Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI)*. 2018.

[233] Yue Wang, Swarat Chaudhuri, and Lydia E. Kavraki. "Bounded Policy Synthesis for POMDPs with Safe-Reachability Objectives." In: *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. AAMAS '18. Stockholm, Sweden: International Foundation for Autonomous Agents and Multiagent Systems, 2018, pp. 238–246.

[234] Majid Khonji, Ashkan Jasour, and Brian Williams. "Approximability of Constant-Horizon Constrained POMDP." In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, July 2019, pp. 5583–5590.

[235] A. Wang, A. Jasour, and B.C. Williams. "Non-Gaussian Chance-Constrained Trajectory Planning for Autonomous Vehicles under Agent Uncertainty." In: *IEEE Robotics and Automation Letters* 5.4 (2020), pp. 6041–6048.

[236] C. Dawson, A. Jasour, A. Hofmann, and B. Williams. "Provably Safe Trajectory Optimization in the Presence of Uncertain Convex Obstacles." In: *2020 IEEE International Conference on Intelligent Robots and Systems (IROS)*. 2020, pp. 6237–6244.

[237] Pedro Santana, Sylvie Thiébaux, and Brian Williams. "RAO*: An Algorithm for Chance-constrained POMDP's." In: *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*. AAAI'16. AAAI Press, 2016, pp. 3308–3314.

[238] Jongmin Lee, Geon-hyeong Kim, Pascal Poupart, and Kee-Eung Kim. "Monte-Carlo Tree Search for Constrained POMDPs." In: *Advances in Neural Information Processing Systems 31*. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett. Curran Associates, Inc., 2018, pp. 7923–7932.

[239] Peter Geibel and Fritz Wysotzki. "Risk-Sensitive Reinforcement Learning Applied to Control Under Constraints." In: *Journal of Artificial Intelligence Research* 24.1 (July 2005), pp. 81–108.

[240] Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. "Constrained Policy Optimization." In: *Proceedings of the 34th International Conference on Machine Learning (ICML)*. Vol. 70. ICML'17. Sydney, NSW, Australia: JMLR.org, 2017, pp. 22–31.

[241] P. Poupart, Aarti Malhotra, P. Pei, Kee-Eung Kim, Bongseok Goh, and Michael Bowling. "Approximate Linear Programming for Constrained Partially Observable Markov Decision Processes." In: *AAAI Conference on Artificial Intelligence (AAAI)*. 2015.

[242] Joshua D. Isom, Sean P. Meyn, and Richard D. Braatz. "Piecewise Linear Dynamic Programming for Constrained POMDPs." In: *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 1*. AAAI'08. Chicago, Illinois: AAAI Press, 2008, pp. 291–296.

[243] Aditya Undurti and Jonathan How. "An Online Algorithm for Constrained POMDPs." In: *2010 IEEE International Conference on Robotics and Automation (ICRA)*. June 2010, pp. 3966–3973.

[244] X. Jiang, S. Chen, J. Yang, H. Hu, and Z. Zhang. "Finding the Equilibrium for Continuous Constrained Markov Games under the Average Criteria." In: *IEEE Transactions on Automatic Control* 65.12 (2020), pp. 5399–5406.

[245] Julian Bernhard, Robert Gieselmann, Klemens Esterle, and Alois Knoll. "Experience-Based Heuristic Search: Robust Motion Planning with Deep Q-Learning." In: *2018 IEEE 21st International Conference on Intelligent Transportation Systems (ITSC)*. 2018, pp. 3175–3182.

[246] Julian Bernhard, Stefan Pollok, and Alois Knoll. "Addressing Inherent Uncertainty: Risk-Sensitive Behavior Generation for Automated Driving Using Distributional Reinforcement Learning." In: *2019 IEEE Intelligent Vehicles Symposium (IV)*. 2019, pp. 2148–2155.

[247] Mark H. M. Winands and Yngvi Björnsson. "Evaluation Function Based Monte-Carlo LOA." In: *Advances in Computer Games*. Ed. by H. Jaap van den Herik and Pieter Spronck. Springer Berlin Heidelberg, 2010, pp. 33–44.

[248] Guillaume Chaslot, Mark Winands, H Herik, Jos Uiterwijk, and Bruno Bouzy. "Progressive Strategies for Monte-Carlo Tree Search." In: *New Mathematics and Natural Computation* 04 (Nov. 2008), pp. 343–357.

[249] J. A. M. Nijssen and Mark H. M. Winands. "Enhancements for Multi-Player Monte-Carlo Tree Search." In: *Computers and Games*. 2010.

[250] Sylvain Gelly, Yizao Wang, and Olivier Teytaud. *Modification of UCT with Patterns in Monte-Carlo Go*. Technical Report RR-6062 32. Jan. 2006, pp. 30–56.

[251] Sylvain Gelly and David Silver. "Combining Online and Offline Knowledge in UCT." In: *Proceedings of the 24th International Conference on Machine Learning - ICML '07*. Corvalis, Oregon: ACM Press, 2007, pp. 273–280.

[252] David Silver, Richard S. Sutton, and Martin Müller. "Sample-Based Learning and Search with Permanent and Transient Memories." In: *Proceedings of the 25th International Conference on Machine Learning - ICML '08*. Helsinki, Finland: ACM Press, 2008, pp. 968–975.

[253] Xiaoxiao Guo, Satinder Singh, Honglak Lee, Richard L. Lewis, and Xiaoshi Wang. "Deep Learning for Real-Time Atari Game Play Using Offline Monte-Carlo Tree Search Planning." In: *Advances in Neural Information Processing Systems*. Vol. 27. Curran Associates, Inc., 2014, pp. 3338–3346.

[254] J.-B. Grill, F. Altché, Y. Tang, T. Hubert, M. Valko, I. Antonoglou, and R. Munos. "Monte-Carlo Tree Search as Regularized Policy Optimization." In: *37th International Conference on Machine Learning (ICML)*. Vol. PMLR 119. 2020, pp. 3769–3778.

[255] B. Kucharski, A. Deihim, and M. Ergezer. "Machine Learning Based Heuristic Search Algorithms to Solve Birds of a Feather Card Game." In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 2019, pp. 9656–9661.

[256] Alba Cotarelo, Vicente García-Díaz, Edward Rolando Núñez-Valdez, Cristian González García, Alberto Gómez, and Jerry Chun-Wei Lin. "Improving Monte Carlo Tree Search with Artificial Neural Networks without Heuristics." In: *Applied Sciences* 11.5 (2021).

[257] T. Papagiannis, G. Alexandridis, and A. Stafylopatis. "Applying Gradient Boosting Trees and Stochastic Leaf Evaluation to MCTS on Hearthstone." In: *19th IEEE International Conference on Machine Learning and Applications, ICMLA*. 2020, pp. 157–162.

[258] Guangli Li, Guancheng Wang, Qiang Wang, Fan Fei, Shuai Lü, and Degui Guo. "ANN: A Heuristic Search Algorithm Based on Artificial Neural Networks." In: *Proceedings of the 2016 International Conference on Intelligent Information Processing*. ICIIP '16. Wuhan, China: ACM, 2016, 51:1–51:9.

[259] Nahas Pareekutty, Francis James, Balaraman Ravindran, and Suril V. Shah. "RRT-HX: RRT With Heuristic Extend Operations for Motion Planning in Robotic Systems." In: *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. Vol. 5A: 40th Mechanisms and Robotics Conference. ASME, Aug. 2016.

[260] Mohak Bhardwaj, Sanjiban Choudhury, and Sebastian Scherer. "Learning Heuristic Search via Imitation." In: *Proceedings of the 1st Annual Conference on Robot Learning*. Vol. 78. Proceedings of Machine Learning Research. PMLR, Nov. 2017, pp. 271–280.

[261] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[262] Markus Wulfmeier, Dominic Zeng Wang, and Ingmar Posner. "Watch This: Scalable Cost-Function Learning for Path Planning in Urban Environments." In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 2089–2095.

[263] Karin Frlic. "Learning Value Functions for Accelerating Risk-Constrained, Interactive Planning in Autonomous Driving." MA thesis. Munich, Germany: Technische Universität München, 2021.

[264] Markus Enzenberger and Martin Müller. "A Lock-Free Multithreaded Monte-Carlo Tree Search Algorithm." In: *Advances in Computer Games*. Ed. by H. Jaap van den Herik and Pieter Spronck. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 14–20.

[265] Richard B. Segal. "On the Scalability of Parallel UCT." In: *Computers and Games*. Ed. by H. Jaap van den Herik, Hiroyuki Iida, and Aske Plaat. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 36–47.

[266] Julian Bernhard, Klemens Esterle, Patrick Hart, and Tobias Kessler. "BARK: Open Behavior Benchmarking in Multi-Agent Environments." In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2020, pp. 6201–6208.

[267] H. Tanveer, M.M. Mubasher, and S.W. Jaffry. "Integrating Human Panic Factor in Intelligent Driver Model." In: *2020 3rd International Conference on Advancements in Computational Sciences (ICACS)*. 2020, pp. 1–6.

[268] Martin Treiber. *Traffic Flow Dynamics*. Jan. 2013.

[269] R. E. Wilson. "Gipps' Model of Highway Traffic." In: *Progress in Industrial Mathematics at ECMI 2000*. Ed. by Angelo Marcello Anile, Vincenzo Capasso, and Antonio Greco. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 293–297.

[270] Y. Peng, S. Liu, and D.Z. Yu. "An Improved Car-Following Model with Consideration of Multiple Preceding and Following Vehicles in a Driver's View." In: *Physica A: Statistical Mechanics and its Applications* 538 (2020).

[271] Arne Kesting, Martin Treiber, and Dirk Helbing. "General Lane-Changing Model MOBIL for Car-Following Models." In: *Transportation Research Record* 1999 (Jan. 2007), pp. 86–94.

[272] Sama Kyle and Yoichi Morales. "Driving Feature Extraction and Behavior Classification Using an Autoencoder to Reproduce the Velocity Styles of Experts." In: *2018 IEEE 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 1337–1343.

[273] Wei Zhan, Liting Sun, Di Wang, Haojie Shi, Aubrey Clausse, et al. "INTERACTION Dataset: An INTERnational, Adversarial and Cooperative moTION Dataset in Interactive Driving Scenarios with Semantic Maps." In: *arXiv:1910.03088 [cs, eess]* (2019). arXiv: 1910.03088 [cs, eess].

[274] Julian Bernhard, Patrick Hart, Amit Sahu, Christoph Schöller, and Michell Guzman Cancimance. "Risk-Based Safety Envelopes for Autonomous Vehicles under Perception Uncertainty." In: *CoRR* abs/2107.09918 (2021).

[275] Gerald Tesauro, V T Rajan, and Richard Segal. "Bayesian Inference in Monte-Carlo Tree Search." In: *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*. UAI'10. AUAI Press, 2010, pp. 580–588.

[276] Jia Guo and Xiaoping Du. "Sensitivity Analysis with Mixture of Epistemic and Aleatory Uncertainties." In: *AIAA Journal* 45.9 (2007), pp. 2337–2349.

[277] Robert Lieck, Vien Ngo, and Marc Toussaint. "Exploiting Variance Information in Monte-Carlo Tree Search." In: *Heuristics and Search for Domain-independent Planning (HSDIP)*. June 2017.

[278] Piero Baraldi and Enrico Zio. "A Combined Monte Carlo and Possibilistic Approach to Uncertainty Propagation in Event Tree Analysis." In: *Risk Analysis* 28.5 (2008), pp. 1309–1326.

[279] Thomas M Moerland, Joost Broekens, and Catholijn M Jonker. "Efficient Exploration with Double Uncertain Value Networks." In: *31st Conference on Neural Information Processing (NIPS)*. 2017, p. 18.

[280] Stefanie Manzinger, Marion Leibold, and Matthias Althoff. "Driving Strategy Selection for Cooperative Vehicles Using Maneuver Templates." In: *2017 IEEE Intelligent Vehicles Symposium (IV)*. 2017, pp. 647–654.

[281] Annkathrin Krämmer, Christoph Schöller, Franz Kurz, Dominik Rosenbaum, and Alois Knoll. "Vorausschauende Wahrnehmung Für Sicheres Automatisiertes Fahren: Validierung Intelligenter Infrastruktursysteme Am Beispiel von Providentia." In: *Internationales verkehrswesen* 72.1 (Mar. 2020), pp. 26–31.

[282] International Organization for Standardization. *ISO 26262:2011(E) – Road Vehicles – Functional Safety*. 2011.

[283] Truls Nyberg, Christian Pek, Laura Col, Christoffer Norén, and Jana Tumova. "Risk-Aware Motion Planning for Autonomous Vehicles with Safety Specifications." In: *2021 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2021, pp. 1016–1023.

[284] David Silver, Satinder Singh, Doina Precup, and Richard S. Sutton. "Reward Is Enough." In: *Artificial Intelligence* 299 (2021), p. 103535.