2019 IEEE Intelligent Vehicles Symposium (IV)
Paris, France. June 9-12, 2019

Addressing Inherent Uncertainty: Risk-Sensitive Behavior Generation for Automated Driving using Distributional Reinforcement Learning

Julian Bernhard[1], Stefan Pollok[1] and Alois Knoll[2]

***Abstract*— For highly automated driving above SAE level 3, behavior generation algorithms must reliably consider the inherent uncertainties of the traffic environment, e.g. arising from the variety of human driving styles. Such uncertainties can generate ambiguous decisions, requiring the algorithm to appropriately balance low-probability hazardous events, e.g. collisions, and high-probability beneficial events, e.g. quickly crossing the intersection. State-of-the-art behavior generation algorithms lack a distributional treatment of decision outcome. This impedes a proper risk evaluation in ambiguous situations, often encouraging either unsafe or conservative behavior. Thus, we propose a two-step approach for risk-sensitive behavior generation combining offline distribution learning with online risk assessment. Specifically, we first learn an optimal policy in an uncertain environment with Deep Distributional Reinforcement Learning. During execution, the optimal risk-sensitive action is selected by applying established risk criteria, such as the Conditional Value at Risk, to the learned state-action return distributions. In intersection crossing scenarios, we evaluate different risk criteria and demonstrate that our approach increases safety, while maintaining an active driving style. Our approach shall encourage further studies about the benefits of risk-sensitive approaches for self-driving vehicles.**

I. Introduction

For highly automated driving above SAE level 3, behavior generation algorithms must reliably consider the inherent uncertainties of the traffic environment, e.g. arising from the variety of driving styles of other participants. Such uncertainties can generate ambiguous decisions, requiring the algorithm to appropriately balance low-probability hazardous events, e.g. collisions, and high-probability beneficial events, e.g. quickly crossing the intersection. A single, numeric value measuring the outcome of a decision does not appropriately characterize such an ambiguous situation, since it neglects the probability of events. Instead of using an expectation-based utility measure, humans resolve ambiguity by minimizing an adequate risk-metric over an outcome distribution [1]. Such risk-metrics better evaluate the potential harm of an action with respect to the probability of occurence.

However, state-of-the-art behavior generation algorithms still lack a distributional treatment of risk. On the one hand, frequently used problem definitions for behavior generation, e.g. MDPs† [2, 3], POMDPs† [4, 5] or MAMDPs† [6–8], adhere to expectation-based return calculation as it is the conventional definition of optimality for such problems.

[1]Julian Bernhard and Stefan Pollok are with fortiss GmbH, An-Institut Technische Universität München, Munich, Germany
[2]Alois Knoll is with Chair of Robotics, Artificial Intelligence and Real-time Systems, Technische Universität München, Munich, Germany
†MDP: Markov Decision Process, POMDP: Partially-Observable MDP, MAMDP: Multi-Agent MDP

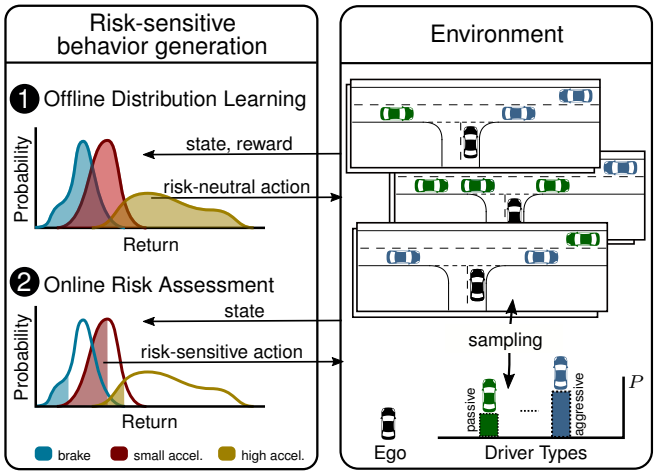


Fig. 1. Our approach deals with an unknown *episode-specific* driver type of a participant sampled from a known *environment-specific* driver type distribution (right). Risk-neutral, state-action return distributions are trained offline for this environment and evaluated online regarding collision risk (left). This two-step risk-sensitive behavior generation approach increases safety in the face of behavioral uncertainty.

On the other hand, most problem solvers, e.g. Deep Q-Learning [2, 9, 10] and Monte Carlo Tree Search [6, 7] for MDPs or Adaptive Belief Tree [4] for POMDPs, output the expected return instead of the return distribution.

Interestingly, recent variants of Deep Reinforcement Learning enable learning of state-action return distributions [11–13], motivating an approach for risk-sensitive behavior generation. Our two-step approach combines offline learning of the return distribution with online risk assessment (Fig. 1). It demonstrates the advantages of using risk-sensitive return metrics to increase safety in the face of behavioral uncertainty. Specifically, we use Deep Distributional Q-Learning to learn the risk-neutral, state-action return distributions in environments with an unknown *episode-specific* behavior type of a participant sampled from a known *environment-specific* behavior type distribution. During execution, the optimal action is then selected based on a distortion risk metric applied to the learned state-action return distributions.

The main contributions of this work are:

- A risk-sensitive behavior generation approach combining offline Deep Distributional Reinforcement Learning with online risk assessment.
- A benchmark of continuous observation spaces suitable for Deep Q-Learning in intersection scenarios.
- An evaluation of risk metrics applicable for behavior generation of autonomous vehicles.

978-1-7281-0560-4/19/$31.00 ©2019 IEEE

- A demonstration of the safety benefits of risk-sensitive behavior generation in environments with behavioral uncertainty.

This work is structured as follows: First, we present related work and introduce our approach. Next, we present the experiment setup with a benchmark of neural network observation spaces, followed by a qualitative and quantitative evaluation of our method.

II. Related Work

A behavior generation algorithm should consider the interactions between participants to successfully navigate in congested traffic, e.g. crowded intersections. Different variants exist to model interactive human behavior.

Cooperative approaches assume that all agents optimize a global cost function. Solving this multi-agent MDP either with optimization- [8, 14] or search-based methods [7, 15] yields a globally optimal solution defining also the ego agent's behavior. However, the equilibrium assumption neglects the uncertainty inherent to human interactions. *Probabilistic approaches* model the uncertainty about the behavior of other participants as hidden state in a POMDP and solve the problem mainly using sampling-based approaches either offline [5] or online [4, 16]. Both, cooperative and probabilistic approaches, face the problem of a combinatorially increasing number of maneuvering options with a growing number of participants. To achieve real-time capability, these algorithms limit the planning horizon [4], consider only interactions with the nearest participants [7, 8] and apply computationally simple traffic prediction models [4, 7], e.g. the Intelligent Driver Model. Menéndez-Romero *et al.* [17] define a *probabilistic cooperative approach* for highway merging. However, their approach assumes a discrete formulation of the participants' intentions.

Deep Reinforcement Learning (DRL) promises interaction-aware decision making at lower computational cost. It learns the expected return of an action by interacting with other participants in simulation. During online planning, the agent exploits this experience. Isele *et al.* [10] apply Deep Q-Networks (DQN) to intersection crossing and extend it to occlusion handling in [9]. Wolf *et al.* [18] evaluate semantic state space definitions for DQNs in highway scenarios. Both approaches apply deterministic traffic models. Yet, even with deterministic models frequently a small percentage of collisions remains. This *epistemic or parametric uncertainty* arises from imperfect information of the learning algorithm about the problem [19], e.g. coming from insufficient exploration or inexact minimization of the loss function. To overcome epistemic uncertainty when using reinforcement learning for autonomous driving behavior generation, one can combine DRL with a search process to allow escaping from local optima of the learned policy [20, 21] or add an additional safety layer to avoid insecure actions [2, 22, 23].

In contrast to epistemic uncertainty, our work deals with *inherent or aleatoric uncertainty* in the environment, e.g. arising from uncertainty about the behavior of other participants. To avoid unsafe decisions in such domains, risk-sensitive reinforcement learning employs an optimization criterion balancing the return and the risk of an action [24]. Such risk criteria and their application to the field of robotics are discussed in [1]. Dabney *et al.* [13] combine a novel, non-parametric approach for return distribution estimation using Deep Distributional Reinforcement Learning (DDRL) with risk-sensitive action selection. They outperform previous DDRL approaches in the domain of Atari games. The risk preferences of humans in driving scenarios are evaluated in [25] using Inverse Reinforcement Learning. To deal with behavioral uncertainty in DRL, Bouton *et al.* [3] add safety rules that block actions when they violate a safety constraint with certain probability. The probabilistic safety measure is calculated separately for each participant in a discretized state space. At an intersection with two participants their approach yields zero collisions. However, the approach neglects interactions between other participants and does not scale efficiently to more complex scenarios.

To the best of our knowledge, our work is the first which addresses the inherent uncertainty of traffic environments with risk-sensitive optimization criteria. Our algorithm avoids a rule-based formulation of safety or discretization of the state space. It is solely based on the reward definition and easily interpretable risk evaluation metrics. Further, we demonstrate the advantages of Distributional Reinforcement Learning for autonomous vehicle behavior generation.

III. Problem Definition

There exist various types of inherent uncertainties. We focus on the inherent uncertainty that arises from the interaction with other traffic participants of varying driving styles. We formulate the problem as Stochastic Bayesian Game (SBG) [26]. The SBG models other agents' behaviors based on a behavior type space and distribution. We can adopt this notion to our domain: A behavior type corresponds to a human driving style, e.g. "aggressive" or "passive". The type distribution models the occurrence frequencies of the driving styles in an environment. This SBG consists of:

- environment state space S with fully observable kinodynamic states s_i of the participants.
- N traffic participants; for each participant $i \in N$:
  - action set A_i of motion primitives
  - behavior type space Θ_i modeling the driving styles, e.g. $\Theta_i = \{$"aggressive,"passive"$\}$.
  - reward function $\mathcal{R}_i : S \times A \times \Theta_i \to \mathbb{R}$ defining the reward after executing the joint action $a \in A$.
  - stochastic policy $\pi_i : \mathbb{H} \times A_i \times \Theta_i \to [0, 1]$ over the sets of state-action histories $\mathbb{H}$, actions A_i and behavior types Θ_i, e.g. $\pi_i(H^t, a,$"aggressive"$)$.
- state transition function $T : S \times A \times S \to [0, 1]$.
- type distribution $\Delta : \mathbb{N}_0 \times \Theta \to [0, 1]$ over the sets of participants' indices $\mathbb{N}_0$ and types Θ. In the above example, it reflects the percentage of drivers showing "aggressive" or "passive" behavior.

Before each episode of the SBG, the type θ_i for each participant i is sampled from Θ_i with probability $\Delta(\theta_i)$. Based on the state-action history H^t at time step t, each participant repeatedly chooses an action according to its behavior $\pi_i(H^t, a, \theta_i)$ until a terminal environment state occurs. The ego-vehicle, $i{=}0$, knows a priori the type distribution Δ and space Θ, and the behaviors π_i; in our approach via inference during training in simulation. The episode-specific, sampled types θ_i of the other participants are unknown.

A behavior generation algorithm must solve the presented SBG. It shall find the optimal driving policy of the ego-vehicle $\pi_0(\cdot,\cdot,\cdot)$, maximizing positive return while considering the risk of negative return due to the uncertainty about the episode-specific driving styles of other participants.

IV. Method

We propose a risk-sensitive behavior generation approach to deal with the presented problem. It encompasses the following two steps visualized in Fig. 1:

1) **Offline Distribution Learning:** The random return variable R depending on action a in environment state s is distributed according to the state-action return distribution $Z(r|s,a)$. Using Distributional Reinforcement Learning [11], we learn $Z^*(r|s,a)$ in simulation for a fixed behavior type space and distribution. It encodes the optimal policy of the ego-agent for such an environment.
2) **Online Risk Assessment:** We deviate from using the standard, expectation-based selection of the optimal action with $a^*{=}\operatorname{argmax}_a \mathbb{E}_{r\sim Z^*}[R]$. Instead, we quantify the collision risk with distortion risk metrics [1] applied to the learned state-action value distribution. The optimal action is then selected based on the measured risk of each action.

Next, we depict the presented approach in detail.

A. Distributional Reinforcement Learning

Reinforcement learning finds an optimal policy for a Markov Decision Process (MDP). The Bellman equation defines the optimal Q-function

$$Q^*(s,a) = \mathbb{E}_{s'}\left[r(s,a,s') + \gamma \max_{a'} Q^*(s',a')|s,a\right] \quad (1)$$

representing the expected return, taking action a in state s and from thereon following the optimal policy $\pi^*(s){=}a^*{=}\operatorname{argmax}_a Q^*(s,a)$. The discount factor γ defines how future rewards $r_t \sim \mathcal{R}$ contribute to the current state-action value. Mnih *et al.* [27] introduced Deep Q-Networks (DQN) enabling Q-learning for problems with higher-dimensional, continuous state space. Double Deep Q-Networks (DDQN) [28] and prioritized experience replay [29] improved convergence and optimality of DQN.

Distributional reinforcement learning models the return R as a random variable with probability distribution $Z(r|s,a)$ and the Q-value being the expected return $Q(s,a){=}\mathbb{E}_{r\sim Z}[R]$. Bellemare *et al.* [11] introduced Deep Distributional Reinforcement Learning to learn $Z(r|s,a)$ non-parametrically in a continuous state space. They proved that the Distributional Bellman equation

$$\begin{aligned} Z(r|s,a) &\overset{\mathrm{D}}{:=} \mathcal{R}(s,a) + \gamma Z(r|s',a') \\ s' &\sim T(\cdot|s,a), a' \sim \pi^*(\cdot|s') \end{aligned} \quad (2)$$

has a unique fix point $Z^*(s,a)$ that minimizes the maximal form of the Wasserstein metric, a distance between two probability distributions. Their proposed algorithm, C51, approximately minimizes this distance to learn the return distribution $Z^*(s,a)$ from which the optimal policy is obtained greedily with $\pi^*(s) = \operatorname{argmax}_a \mathbb{E}[Z^*(s,a)]$.

Quantile Regression Deep Q-Learning (QRDQN) improves the performance of the C51 algorithm by truly minimizing the Wasserstein metric [12]. It approximates the inverse cumulative distribution function (c.d.f) or quantile function F_Z^{-1} at discrete probabilities.

B. Training Process

We apply the QRDQN algorithm to learn the state-action distribution $Z^*(r|s,a)$ of the ego agent for a *fixed* behavior type space and distribution in simulation.

Before the start of a training episode, we sample episode-specific behavior types θ_i for all other participants i from the fixed environment-specific type distribution Δ. The sampled types remain constant for the rest of the episode. The simulated participants then behave according to $\pi_i(\cdot,\cdot,\theta_i)$. By seeing a multitude of episodes with different behavior types, the learning agent infers the type distribution and space, and learns a risk-neutral, optimal policy for the given SBG. After learning, $Z^*(r|s,a)$ expresses the inherent uncertainty about the actual behavior types appearing at a specific episode.

C. Risk Assessment

During execution, we quantify the risk of an action based on the learned distribution $Z^*(s,a)$ using risk metrics. Learning of the state-action distribution occurred risk-neutral with expectation-based action selection. Now, during execution of the learned behavior, we quantify the action risks with a distortion risk metric applied to the learned risk-neutral distribution.

Distortion risk metrics comply with six mathematical axioms and emerged from the field of finance. Their application as risk metric in robotics is discussed by Majumdar and Pavone [1]. Sequential decision making may be temporally inconsistent when risk evaluation is not applied already during training [30]. However, the advantage of assessing risk based on the risk-neutral distribution is that the most suitable risk estimator and its parameters could be adapted online to the encountered traffic scene.

We evaluate two distortion risk metrics. For better readability, we denote $Z = Z^*(r|s,a)$ in the following:

- **Conditional Value at Risk** (CVaR) [1]:

$$\rho_{\mathrm{CVaR}}[Z] = \mathbb{E}_{r\sim Z}[R|R < \mathrm{VaR}_\alpha] \quad (3)$$

with probability parameter α and the value at risk $\mathrm{VaR}_\alpha{:=}F_Z^{-1}(\alpha)$. Thus, α is the cumulative probability

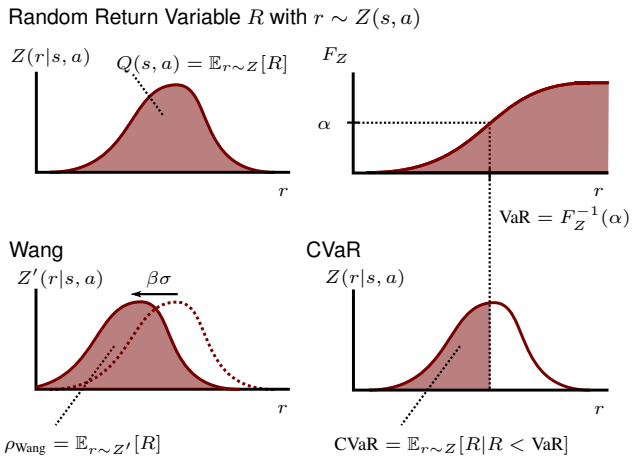


Fig. 2. Graphical representation of the calculation of Wang and CVaR risk metrics: *Wang* distorts the original distribution and calculates an expectation over the resulting distribution. It shifts the mean for a normal-like distribution. *CVaR* considers only returns below the Value at Risk (VaR).

of returns smaller than VaR_α. The CVaR_α is the mean over this section of the return distribution.

- **Wang** [31]: Wang distorts the original cumulative distribution and uses the expectation of the resulting distribution:

$$F_{Z'} = \Phi[\Phi^{-1}(F_Z) + \beta]$$
$$\rho_{\text{Wang}}[Z] = \mathbb{E}_{r\sim Z'}[R] \tag{4}$$

where Φ is the standard normal c.d.f and β a real-valued parameter. For a normal distribution, this metric shifts its mean with $\mu'{=}\mu{+}\beta\sigma$.

The calculation of the risk metrics is represented graphically in Fig. 2.

Action selection is greedy. For each action a_i, we calculate its expected return under the risk metric and select with

$$a^*(s_t) = \underset{a_i\in\mathcal{A}}{\text{argmax}}\, \rho_{\text{xxx}}(Z(r|s_t,a_i)). \tag{5}$$

the optimal, risk-sensitive action a^* in state s_t.

V. Experiment Setup

We evaluate our approach on four turning scenarios in the T-intersection given in Fig. 3. Below, we describe the experiment setup in detail.

A. Scenario

We consider four turning scenarios, with either left or right turn and a varying number of participants. The other participants have right of way, but react to the ego-vehicle. At the beginning of an episode, the ego-vehicle starts at the same point of the intersection with zero initial velocity. It succeeds when reaching the end of the turning lane without collision. We limit the velocity for all participants to 54 km/h and the maximum acceleration to $5\,\text{m}\,\text{s}^{-2}$ and $-4\,\text{m}\,\text{s}^{-2}$. The time step of the simulation is 200 ms.

B. Behavior Modeling

We define two deterministic driving styles "passive" and "aggressive". Stochastic behavior types are planned in future work. The corresponding policies $\pi(\cdot,\cdot,\text{"passive"})$ and $\pi(\cdot,\cdot,\text{"agressive"})$ use an Intelligent Driver Model (IDM) that reacts also to turning vehicles. $\pi(\cdot,\cdot,\text{"agressive"})$ accelerates to the desired velocity and keeps the gap to other IDM vehicles. But, it does not react and brake, if the ego-vehicle occupies the lane.

As we want to evaluate performance with and without behavioral uncertainty, we consider two type definitions:

- **Single:** All the other drivers behave aggressively, thus $\Theta_{\text{single}}{=}\{\text{"aggressive"}\}$ with $\Delta_{\text{single}}(\text{"aggressive"}){=}1.0$.
- **Mixed:** Drivers act with equal percentage passively or aggressively, thus $\Theta_{\text{mixed}}{=}\{\text{"passive"},\text{"aggressive"}\}$ with uniform type distribution $\Delta_{\text{mixed}}(\text{"passive"}){=}0.5$ and $\Delta_{\text{mixed}}(\text{"agressive"}){=}0.5$.

Before an episode, a single type is sampled from the selected type distribution. It is then used by all other participants.

C. Deep Reinforcement Learning

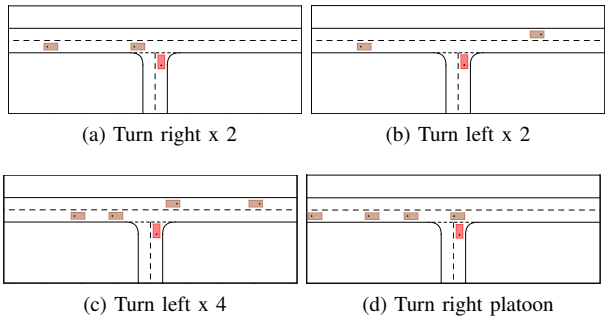
We train DQN and QRDQN agents separately for each scenario for both type definitions "single" and "mixed". We employ the standard DQN [27] and QRDQN [12] architectures with fully connected layers ($4{\times}300$ ReLUs), outputting a single value for each action, respectively $N{=}200$ quantiles for each action. The input consists of the concatenated observations of all participants. Observation and action space are given below. We use prioritized experience replay [29] and Double DQN [28] for both DQN and QRDQN.

Rewards are defined for the ego-agent. The other participants do not adhere to reward maximization. They are fully controlled by their policies π. The ego-agent receives a positive reward $\mathcal{R}_{goal}{=}100$ for reaching the goal, and $\mathcal{R}_{collision}{=}-1000$ for collisions. Every action costs additionally $\mathcal{R}_{step} = -5$. The discount factor γ is set to 0.95.

D. Training & Test Data

For each combination of scenario and type definition, we define a fixed training and test data set consisting of 100 000 episode definitions, respectively. Each episode definition contains the environment state, specifying the initial kinodynamic vehicle states of all participants at the beginning of the episode, and the applied behavior type. The distribution of behavior types in the data set complies with the selected type distribution Δ_{single} or Δ_{mixed}.

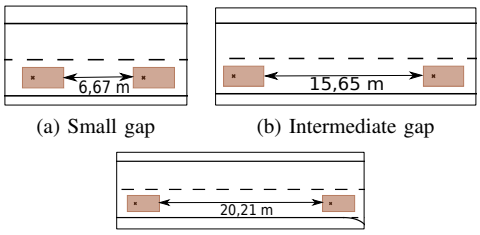
To improve generalization, we vary the number of participants up to the maximum count of the scenario and



(a) Turn right x 2 (b) Turn left x 2

(c) Turn left x 4 (d) Turn right platoon

Fig. 3. Intersection scenarios considered in the evaluation.



(a) Small gap (b) Intermediate gap

(c) Large gap

Fig. 4. Different gap sizes in the training and test data sets.

shuffle the order of vehicle state positions in the concatenated observation space. The other participants have a random velocity between $29\,\mathrm{km\,h^{-1}}$ and $36\,\mathrm{km\,h^{-1}}$. Further, the data sets contain different initial gap sizes between two vehicles in the same lane as depicted in Fig. 4.

E. Evaluation Metrics and Significance Testing

The following metrics are used for evaluation:

- **Success/Collision rate** [%]: percentage of runs the ego-vehicle reached the end of the turning lane/collided.
- **Max. time rate** [%]: percentage of runs exceeding maximum allowed crossing time ($14\,\mathrm{s}$).
- **Crossing time** [s]: time to reach the goal averaged over successful runs.

We use a fixed set of $10\,000$ test runs per approach and scenario to calculate the metrics, employing the best performing training checkpoint after a fixed number of training steps.

To check for significance in performance differences, we use paired statistical tests in which the test set index is the independent variable. We perform for the binomial variables (success, collision, max. time) a Cochran's Q test followed by pair-wise McNemar tests. For the crossing time, we use a repeated measures ANOVA followed by pair-wise dependent t-tests. Confidence level is 0.95 with Bonferroni correction for the pair-wise tests.

F. Action Space

We use only longitudinal actions $A_{\text{ego}}{=}(-3, 0, 2, 5)$ in $\mathrm{m\,s^{-2}}$ along predefined left or right turning paths to facilitate an empirical analysis of the benefits of our method. A generally applicable driving policy including lateral actions is deferred to future work. The other participants have a continuous action space defined by their behavior model.

G. Observation Space

To find an appropriate observation space of the intersection scenario applied as input to DQN and QRDQN, we benchmarked different observation spaces. We chose a right turn scenario with a single other participant and type distribution Θ_{single}, and compared the success rates of a DQN agent for the following observation spaces:

1) Cartesian coordinates and velocity

$$(x_{\text{ego}}, y_{\text{ego}}, v_{\text{ego}}, x_1, y_1, v_1, \ldots, x_k, y_k, v_k)$$

2) Cartesian coordinates, velocity and binary value

$$(x_{\text{ego}}, y_{\text{ego}}, v_{\text{ego}}, x_1, y_1, v_1, c_1, \ldots, x_k, y_k, v_k, c_k)$$

TABLE I

RESULTS OF THE OBSERVATION SPACE BENCHMARK.

Observation Representation	Collision Rate [%]	Max. Time Rate [%]	Crossing Time [s]
1) x, y, v	9.8	0.0	4.53
2) x, y, v, c	2.6	0.0	4.70
3) $\Delta x, \Delta y, \Delta v_x \Delta v_y$	23.8	0.0	3.31
4) $\Delta x, \Delta y, \Delta v_x \Delta v_y$, $\text{state}_{\text{ego}}$	1.4	0.0	4.06
5) $d, \Delta\phi, v$, TTC	31.9	0.0	3.24
6) d, v_ϕ, lane	2.6	0.0	4.79

3) Relative features to ego-vehicle

$$(\Delta x_1, \Delta y_1, \Delta v_{x,1}, \Delta v_{y,1}, \ldots, \Delta x_k, \Delta y_k, \Delta v_{x,k}, \Delta v_{y,k})$$

4) Relative features and ego-vehicle state

$$(x_{\text{ego}}, y_{\text{ego}}, v_{\text{ego}}, \Delta x_1, \Delta y_1, \Delta v_{x,1}, \Delta v_{y,1}, \ldots, \Delta x_k, \Delta y_k, \Delta v_{x,k}, \Delta v_{y,k})$$

5) Distance, orientation, velocity and TTC (inspired by [10])

$$(x_{\text{ego}}, y_{\text{ego}}, v_{\text{ego}}, d_1, \Delta\phi_1, v_1, \text{TTC}_1, \ldots, d_k, \Delta\phi_k, v_k, \text{TTC}_k)$$

6) Distance, signed velocity and lane

$$(d_1, v_{\phi,1}, \text{lane}_1, \ldots, d_k, v_{\phi,k}, \text{lane}_k)$$

All real-valued numbers are normalized to the range -1 to 1. The Δ sign denotes the value difference, d_k the Euclidean distance and TTC_k is the Time-To-Collision between vehicle k and the ego-vehicle. The binary value c is one, if the vehicle is present in the scene, otherwise, it is zero. The signed velocity $v_{\phi,k}$ is positive when driving from left to right or bottom to top, and negative when driving right to left. The lane index lane_k can take a value between one and four to indicate the position on one of the four available lanes. If the number of vehicles is lower than the scenario maximum, we calculate missing features based on a vehicle with a zeroed state, not interfering with the drivable space of the intersection, TTC$=-1$ and lane$=0$.

Table I compares the results, in this preliminary evaluation, without significance testing. The TTC-based representation lead to an aggressive driving behavior with high collision rate. Interestingly, sparse observation spaces such as 2) and 6) achieved an acceptable overall performance. However, relative state information in combination with the ego-vehicle state (4) outperformed the other representations in terms of collision rate and achieved a medium crossing time. Thus, we decided to use this representation in the evaluation.

VI. EVALUATION

In our final evaluation, we compare the performance of the DQN baseline [2, 10, 18] with QRDQN, and QRDQN with risk-sensitive policy evaluation in environments with single and mixed behavior type definitions.

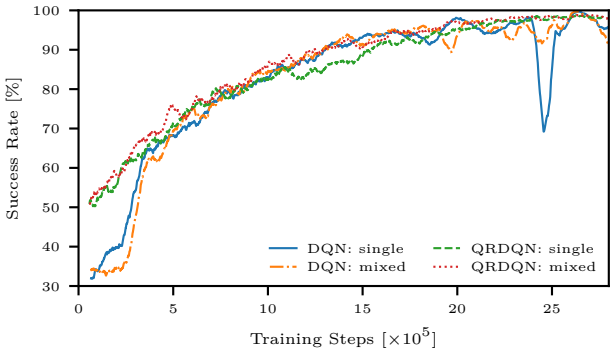


Fig. 5. Success rates during training of DQN and QRDQN for the single and mixed behavior type space in the "Turn left x 2" scenario.

A. Exemplary Training Results

Fig. 5 compares the training data success rates of DQN and QRDQN over the course of training, exemplarily for the "Turn left x 2" scenario with single or mixed behavior type space. QRDQN converged smoothly in both the single and mixed setting. In contrast, DQN fluctuated strongly from the 20×10^5 episode on in the mixed setting. As expected, QRDQN showed thus improved stability in the training process for stochastic environments.

B. Risk Metric Parameterization

In a preliminary study, we coarsely evaluated the influence of the risk metric parameters α of CVaR and β of Wang using the trained QRDQN agents. We considered the average performance over all scenarios in the training data set. We discovered that Wang was very sensitive to parameter changes and started to yield conservative driving behavior for $\beta < -0.4$. In contrast, CVaR was more robust to parameter changes. For the following evaluation, we set $\alpha = 0.7$ and $\beta = -0.2$.

C. Quantitative Analysis

First, we qualitatively compare the different approaches. We highlight in bold the best/worst result of a group, if the group test and all pair-wise tests within the group were significant as described in Sec. V-E.

First, we discuss the advantage of Distributional Reinforcement Learning over standard Deep Q-Learning. Table II depicts the performance of these two algorithms for "single" and "mixed" behavior type space averaged over *all scenarios*. For Θ_{single}, QRDQN achieved a slightly higher success rate

TABLE II

COMPARISON OF ALGORITHMS AVERAGED OVER ALL SCENARIOS.

Θ_{others}	Algorithm	% Collisions	% Max. Time	Crossing Time [s]
single	QRDQN + CVaR	**1.18**	0.00	5.43
	DQN	3.09	0.00	6.12
	QRDQN	2.10	0.00	**5.27**
mixed	QRDQN + CVaR	**0.70**	0.00	6.19
	DQN	6.98	**24.75**	7.45
	QRDQN	1.68	0.00	**5.16**

TABLE III

COMPARISON OF RISK METRICS IN THE "MIXED" TYPE SPACE.

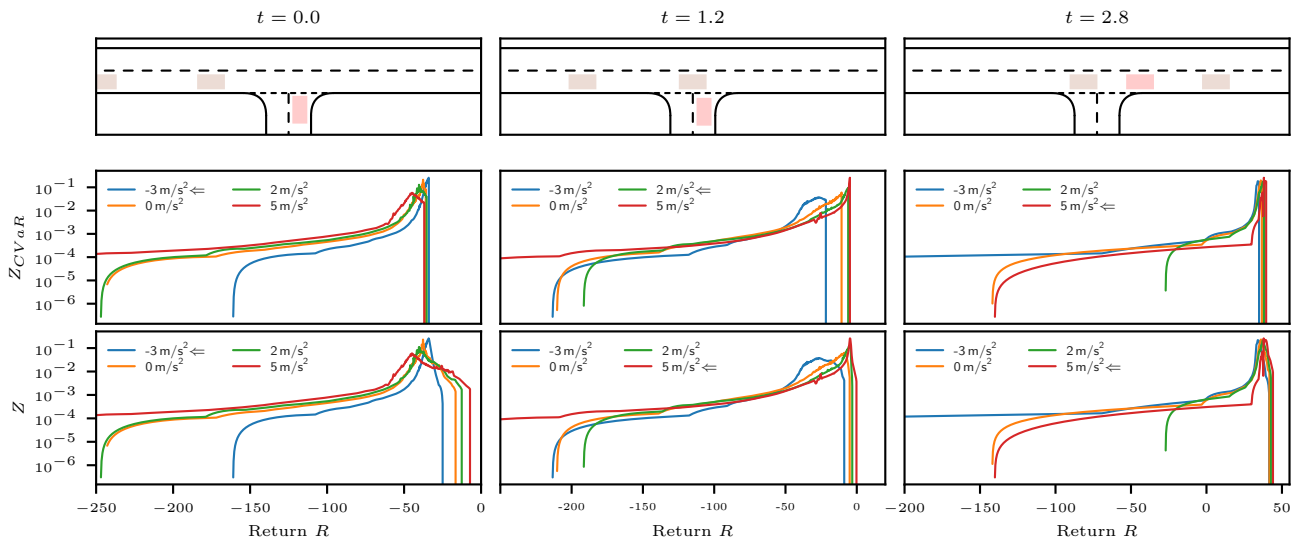
Scenario	Risk Measure	% Collisions	% Max. Time	Crossing Time [s]
left x 4	CVaR	1.08	0.00	5.43
	None	**2.06**	0.00	**5.34**
	Wang	0.88	0.00	5.46
right platoon	CVaR	0.00	0.00	**10.83**
	None	**1.15**	0.00	6.98
	Wang	-	**100.00**	-
left x 2	CVaR	0.08	0.00	4.84
	None	**0.61**	0.00	**4.76**
	Wang	0.09	0.00	4.88
right x 2	CVaR	1.64	0.00	3.67
	None	**2.90**	0.00	**3.57**
	Wang	1.59	0.00	3.70

than standard DQN. However, for Θ_{mixed}, when uncertainty about the behavior of others was given, QRDQN reduced collisions by 5%. There, a local minima in the learned policy of DQN led to a large max. time rate. The crossing time decreased in the "single" and "mixed" case with QRDQN. These results underline the benefits of learning state-action *distributions* $Z(s, a)$ in uncertain environments. State-action *values* $Q(s, a)$ do not dissolve the subtle return nuances of such domains. A detailed, more general discussion of the benefits of the distributional approach is given in [11].

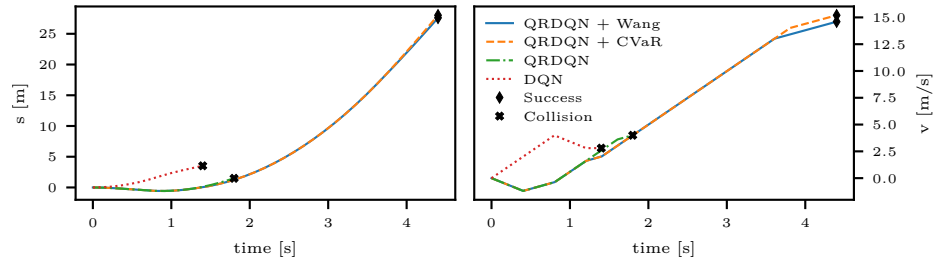
Yet, a collision rate of 1.68% remained when using QRDQN with Θ_{mixed}. Table II provides the results for QRDQN combined with the best performing risk measure CVaR. Applying risk assessment during online planning outperformed the QRDQN approach *significantly*, *halving* the collision rate from 1.68% to 0.7%. The crossing time increased slightly due to conservative driving in the "Turn right platoon" scenario. Risk assessment led in that case to longer waiting times at the entrance of the intersection. When no inherent uncertainty was present (Θ_{single}), the learned distributions represent only model uncertainties, e.g. arising due to insufficient exploration or loss minimization. Risk assessment was still beneficial in that case.

Next, we compared the CVaR and Wang risk measures. For the different scenarios, the results in the "mixed" setting are depicted in Table III. Pair-wise significance was found against the QRDQN approach (risk measure "none"); not between CVaR and Wang. Still, we detect a tendency: Wang reduced collisions in *three* scenarios compared to QRDQN. In the "right platoon" scenario, Wang led to conservative driving, failing to cross the intersection within the maximum episode duration. In contrast, CVaR reduced collisions in *all* cases. The crossing time is comparable to QRDQN. In the platoon scenario, CVaR drives conservatively too, increasing crossing time noticeably. But in contrast to Wang, it still managed to cross the intersection in all cases.

We conclude that CVaR is a suitable metric to evaluate risk in behavior generation algorithms of autonomous vehicles. Regarding the remaining collisions, further studies should investigate the effects of different risk metric parameterizations and evaluate the influence of epistemic uncertainties on



(a) Learned return distributions Z and modified return distributions Z_{CVaR} with returns pruned above the VaR at specific time points of the scenario. An arrow indicates the optimal action obtained with either $a^* = \text{argmax}_a \, \mathbb{E}[Z]$ or $a^* = \text{argmax}_a \, \mathbb{E}[Z_{\text{CVaR}}]$.



(b) Longitudinal position s and velocity v of the ego-agent over the course of the scenario.

Fig. 6. For a "Turn right x 2" episode with passive behavior of other participants, we show the evaluation of the scene and the corresponding distributions at specific timepoints and compare velocity and longitudinal position for all algorithms. Risk-sensitive action selection with CVaR yields a more moderate acceleration at time point $t = 1.2\,\text{s}$ in comparison to the learned optimal action of QRDQN choosing highest acceleration. This subtle difference made the risk-sensitive approach successfully complete this episode whereas DQN and QRDQN failed.

the reliability of the proposed approach.

D. Qualitative Analysis

We pick out a single episode to qualitatively examine the reasons for better performance with risk assessment. We consider a "Turn right x 2" episode with "passive" behavior of other participants. In this case, risk assessment with CVaR or Wang resulted in successful intersection crossing, and DQN and QRDQN collided.

Fig. 6b depicts the longitudinal position s and velocity v over the course of the scenario. Slight backwards movements occurred with the distributional approaches, since, with our reward definition, the policy was optimized solely for safety. We postpone comfort constraints to later work. For specific time points in this scenario, Fig. 6a displays the current traffic situation, and for all actions, the corresponding learned distributions Z and the modified distributions Z_{CVaR} with returns pruned above the VaR. An arrow "⇐" highlights the selected, optimal action, respectively.

Overall, we see that the distributions mainly differ in the length of a tail with lower-probability negative returns and only marginally with respect to higher probability positive returns. At the beginning of the scenario at t=0 s, risk assessment with CVaR did not change the optimality of the learned action. Braking remained optimal, since the other actions' long negative tails dominate their distributions even after pruning returns above the VaR. The time point t=1.2 s, however, was critical for the final outcome of the episode. The learned policy of QRDQN chose highest possible acceleration, primarily as the positive return probability in its distribution outweighs the large negative tail. In contrast, considering only the return values below the VaR for decision making yielded risk-averse action selection with the optimal action being a lower acceleration value. This avoided the collision occurring with QRDQN and led to a successful completion of the scenario ($t = 2.8\,\text{s}$).

This example clarifies the benefits of the CVaR metric for behavior generation in uncertain environments: Due to the inherent uncertainty about a hazardous event in the environment, the distributions of riskier actions consist of a longer, low-probability tail at negative returns facing higher probability peaks at more positive returns. The decision becomes ambiguous. The CVaR risk measure decides based on the VaR which returns to prune from the distribution, strengthening the contribution of less likely negative outcomes. Overall, this removes the ambiguity of a decision, oc-

curring with riskier actions, and yields a saver driving policy.

A video comparing the performance of the evaluated algorithms and risk measures at selected episodes is found under `https://youtu.be/PSDFEG5d1xg`.

VII. Conclusion and Future Work

We proposed a two-step approach for risk-sensitive behavior generation, evaluating the risk of actions online, based on return distributions learned offline with Deep Distributional Reinforcement Learning. We evaluated two distortion risk metrics and demonstrated that our approach increases safety in environments with inherent uncertainty about other participants' behaviors while avoiding too conservative driving.

Majumdar and Pavone [1] discussed the application of risk metrics, emerging from finance, to robotics. Our approach presents now a step forward in applying risk-sensitive behavior generation for autonomous driving. Yet, a distributional consideration of risk in other methods, e.g search-based methods, would broaden our understanding of its benefits and challenges.

To achieve a high level of safety under inherent *and* epistemic uncertainties, we plan to combine the approach with methods reducing epistemic uncertainty in the future, e.g an additional search process [20].

References

[1] Majumdar, A. and Pavone, M., "How Should a Robot Assess Risk? Towards an Axiomatic Theory of Risk in Robotics," *CoRR*, vol. abs/1710.11040, 2017.

[2] Mirchevska, B., Pek, C., *et al.*, "High-Level Decision Making for Safe and Reasonable Autonomous Lane Changing Using Reinforcement Learning," en, in *21st International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2018.

[3] Bouton, M., Karlsson, J., *et al.*, "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.

[4] Hubmann, C., Schulz, J., *et al.*, "A Belief State Planner for Interactive Merge Maneuvers in Congested Traffic," in *21st International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2018.

[5] Bouton, M., Cosgun, A., *et al.*, "Belief state planning for autonomously navigating urban intersections," in *Intelligent Vehicles Symposium (IV)*, IEEE, 2017.

[6] Kurzer, K., Zhou, C., *et al.*, "Decentralized Cooperative Planning for Automated Vehicles with Hierarchical Monte Carlo Tree Search," in *Intelligent Vehicles Symposium (IV)*, Jun. 2018.

[7] Lenz, D., Kessler, T., *et al.*, "Tactical cooperative planning for autonomous highway driving using Monte-Carlo Tree Search," in *Intelligent Vehicles Symposium (IV)*, IEEE, 2016.

[8] Burger, C. and Lauer, M., "Cooperative Multiple Vehicle Trajectory Planning using MIQP," in *21st International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2018.

[9] Isele, D., Rahimi, R., *et al.*, "Navigating Occluded Intersections with Autonomous Vehicles Using Deep Reinforcement Learning," in *International Conference on Robotics and Automation (ICRA)*, IEEE, 2018.

[10] ——, "Navigating Occluded Intersections with Autonomous Vehicles using Deep Reinforcement Learning," May 2017.

[11] Bellemare, M. G., Dabney, W., *et al.*, "A distributional perspective on reinforcement learning," *CoRR*, vol. abs/1707.06887, 2017.

[12] Dabney, W., Rowland, M., *et al.*, "Distributional Reinforcement Learning with Quantile Regression," *CoRR*, vol. abs/1710.10044, 2017.

[13] Dabney, W., Ostrovski, G., *et al.*, "Implicit Quantile Networks for Distributional Reinforcement Learning," in *35th International Conference on Machine Learning (ICML)*, vol. 80, PMLR, Jul. 2018.

[14] Sadigh, D., Sastry, S., *et al.*, "Planning for Autonomous Cars that Leverages Effects on Human Actions," in *Proceedings of the Robotics: Science and Systems Conference (RSS)*, Jun. 2016.

[15] Kurzer, K., Engelhorn, F., *et al.*, "Decentralized Cooperative Planning for Automated Vehicles with Continuous Monte Carlo Tree Search," *CoRR*, vol. abs/1809.03200, 2018.

[16] Bai, H., Cai, S., *et al.*, "Intention-aware online POMDP planning for autonomous driving in a crowd," in *Robotics and Automation (ICRA)*, IEEE, 2015.

[17] Menéndez-Romero, C., Sezer, M., *et al.*, "Courtesy Behavior for Highly Automated Vehicles on Highway Interchanges," in *Intelligent Vehicles Symposium (IV)*, IEEE, Jun. 2018.

[18] Wolf, P., Kurzer, K., *et al.*, "Adaptive Behavior Generation for Autonomous Driving using Deep Reinforcement Learning with Compact Semantic States," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2018.

[19] Dilokthanakul, N. and Shanahan, M., "Deep Reinforcement Learning with Risk-Seeking Exploration," in *From Animals to Animats 15*, Springer International Publishing, 2018.

[20] Bernhard, J., Gieselmann, R., *et al.*, "Experience-Based Heuristic Search: Robust Motion Planning with Deep Q-Learning," in *21st International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2018.

[21] Paxton, C., Raman, V., *et al.*, "Combining neural networks and tree search for task and motion planning in challenging environments," in *International Conference on Intelligent Robots and Systems*, IEEE, Sep. 2017.

[22] Mukadam, M., Cosgun, A., *et al.*, "Tactical Decision Making for Lane Changing with Deep Reinforcement Learning," in *Conference on Neural Information Processing (NIPS)*, 2017.

[23] Shalev-Shwartz, S., Shammah, S., *et al.*, "Safe, Multi-Agent, Reinforcement Learning for Autonomous Driving," *CoRR*, vol. abs/1610.03295, 2016.

[24] García, J. and Fernández, F., "A Comprehensive Survey on Safe Reinforcement Learning," *Journal of Machine Learning Research*, vol. 16, 2015.

[25] Majumdar, A., Singh, S., *et al.*, "Risk-sensitive inverse reinforcement learning via coherent risk models," in *Robotics: Science and Systems*, 2017.

[26] Albrecht, S. V., Crandall, J. W., *et al.*, "Belief and Truth in Hypothesised Behaviours," en, *Artificial Intelligence*, vol. 235, Jun. 2016.

[27] Mnih, V., Kavukcuoglu, K., *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, Feb. 2015.

[28] Van Hasselt, H., Guez, A., *et al.*, "Deep Reinforcement Learning with Double Q-Learning," in *30th AAAI Conference on Artificial Intelligence*, AAAI Press, 2016.

[29] Schaul, T., Quan, J., *et al.*, "Prioritized experience replay," in *International Conference on Learning Representations (ICLR)*, 2016.

[30] Ruszczyński, A., "Risk-averse dynamic programming for Markov decision processes," *Mathematical Programming*, vol. 125, no. 2, Oct. 2010.

[31] Wang, S. S., "A Class of Distortion Operators for Pricing Financial and Insurance Risks," *The Journal of Risk and Insurance*, vol. 67, no. 1, 2000.