Ensemble-based Uncertainty Estimation with overlapping alternative Predictions

Dirk Eilers
Fraunhofer Institute for Cognitiv Systems
Fraunhofer Gesellschaft
Munich, Germany
dirk.eilers@iks.fraunhofer.de

Felippe Schmoeller da Roza
Fraunhofer Institute for Cognitiv Systems
Fraunhofer Gesellschaft
Munich, Germany

Karsten Roscher
Fraunhofer Institute for Cognitiv Systems
Fraunhofer Gesellschaft
Munich, Germany

Abstract

A reinforcement learning model will predict an action in whatever state it is - even if there is no distinct outcome due to unseen states the model may not indicate that. Different methods for uncertainty estimation can be used to indicate this. Although a known approach in AI, most of the available uncertainty estimation methods are not able to deal with the choice overlap that happens in states where multiple actions can be taken by a reinforcement learning agent with a similar performance outcome. In this work, we investigate uncertainty estimation on simplified scenarios in a gridworld environment. Using ensemble-based uncertainty estimation we propose an algorithm based on action count variance to deal with discrete action spaces and delta to ID action variance calculation to handle overlapping alternative predictions. To visualize the expressiveness of the model uncertainty we create heatmaps for different in-distribution and out-of-distribution (OOD) scenarios and propose an indicator for uncertainty. We can show that the method is able to indicate potentially unsafe states when the agent is facing novel elements in the OOD scenarios while capable to distinguish uncertainty resulting from OOD instances from uncertainty caused by the overlapping of alternative predictions.

1 Introduction

In order to apply Machine Learning in today's and future's real-world applications it is necessary to take into account the given system boundaries of an application, e.g. resource and/or timing requirements and safety constraints. There is a high incentive to use Reinforcement Learning (RL) in parts of these system approaches, as to the high flexibility and self-handling in decision-making tasks. However, this also comes with a high risk in safety-critical applications. As an RL agent will predict an action in whatever state it will find itself, it is of high importance for the application controller to be aware of the certainty and confidence of its own decisions in all situations it could face. This includes situations from inside as from outside the training data.

To overcome this problem in the larger field of safe RL this paper focuses on estimating uncertainty from data during deployment (test data) that is out of the distribution of the training data (distributional shift) as one aspect of the problem solution. Distributional shift in data science is widely understood as the distributional difference between the training and test data of an algorithm. This distributional shift originates from different sources, such as perturbations to the data-set due to system complexity

Deep RL Workshop at the 36th Conference on Neural Information Processing Systems (NeurIPS 2022).

and a lack of training data. In machine learning, a distributional shift often leads to degraded performance during testing, because the probability distribution over data (or state-action pairs in RL) is shifted and a classifier or an agent may therefore predict wrong or sub-optimal classes (or actions). In general, when the testing distribution differs from the training distribution, machine learning systems may not only demonstrate poor performance but also wrongly assume that their performance is good.

This issue is also present in system solutions based on Reinforcement Learning. To overcome this limitation, safe Reinforcement Learning (Safe RL) solutions must be capable of detecting and handling the uncertainty in the decision-making process. Therefore, it is necessary to deal with uncertainty and potentially improve given algorithms to optimize these with respect to safety constraints. For instance, uncertainty estimation can detect a lack of generalization due to insufficient training and unseen states during training (OOD, epistemic uncertainty) as well as uncertainty resulting from randomness in the environment (aleatoric uncertainty). For epistemic uncertainty, estimates can be obtained from the variance over the prediction of a set of agents trained independently. In states where the agents are unsure due to insufficient training, different agents are prone to diverge in their action predictions. This can be utilized to indicate uncertainty, however, in some optimization problems, multiple paths could be taken to get to the goal and the agents can deviate within the possible alternative predictions. Therefore, it is necessary to differentiate between the two effects.

2 Related work

There is a lot of work in recent years with respect to OOD in the classical image classification domain as well as some work in the RL domain. Pimentel *et al.* [2014] define novelty as samples that differ in some respect from the data used during training. Insufficient training data will make the model less capable to generalize to data that conceptually belong to the same distribution but were not experienced during training. Hendrycks and Gimpel [2018] define OOD as testing samples drawn from a distribution that differs from the training data and describe OOD detection as a threshold–based process. Furthermore, a differentiation is posed between OOD instances that are near or far from the training data distribution. The closer the OOD sample is to the training data distribution, the easiest it is to mitigate the issue by training with more data, whereas the further away it gets it will most likely represent conceptual or semantic gaps that are not easily overcome, e.g., samples which are not included in the known classes. In this regard, DNNs tend to be overconfident in predictions on unseen data and can give unpredictable results for far-from-distribution test data Lütjens *et al.* [2019].

A lot of prior work focuses on OOD as a concept of samples from outside the class set. If samples are outside the set, classifications for that cannot be learned, even with more/unlimited training. A recurrent approach is to specify a separate OOD class to train the model on Pimentel *et al.* [2014]. Similarly, DeVries and Taylor [2018], Mohseni *et al.* [2020] define OOD samples as examples of classes different from those in the ID dataset. Lee *et al.* [2018] describe ID as a trained distribution by a classifier and OOD as sufficiently different from it. Also, Schwaiger *et al.* [2020] follow the approach to consider strong conceptual differences between training and test data to be OOD. Yu and Aizawa [2019] go as far as to define OOD by the gap in between classified ID data. They propose to maximize the discrepancy between the decision boundaries of e.g. two classifiers to push OOD samples outside. They also follow the concept of contrasting samples being near and far from the known distribution.

Furthermore, it is important to understand the difference between epistemic and aleatoric uncertainty for uncertainty estimation as a proxy for OOD detectors. Epistemic uncertainty arises out of a lack of sufficient data to exactly infer the underlying system Sedlmeier *et al.* [2019]. Clements *et al.* [2020] claim epistemic uncertainty to stem from limited data. Epistemic uncertainty can indicate samples that reside far away from the data distribution, as well as the capabilities of a model to generalize to data close to it Postels *et al.* [2020]. In contrast, aleatoric uncertainty arises from stochastic environments and must be accounted for in risk-sensitive applications Clements *et al.* [2020], Chua *et al.* [2018]. Aleatoric uncertainty cannot be solved just by more training. For safety reasons, aleatoric uncertainty is a significant indicator in RL and should also be considered. However, in this paper, we focus on the detection of epistemic uncertainty, as we focus on distributional shift and OOD detection.

Kahn *et al.* [2017] present an uncertainty-aware model-based learning algorithm that estimates the probability of collision together with a statistical estimate of uncertainty. The predictive model is based on bootstrapped neural networks using dropout. In regions of high uncertainty, their risk-averse cost function causes the robot to revert to a cautious low-speed strategy. In Da Silva *et al.* [2020] they propose an action-advising framework where the agent asks for advice when its epistemic uncertainty is high for a certain state to accelerate reinforcement learning. They add as a last layer multiple heads estimating separately expected values for each action, as done in Bootstrapped DQN. As the learning algorithm updates the network, their predictions get closer to the real underlying function, and close to the each other. Sedlmeier *et al.* [2019] use uncertainty-based OOD, using Q-value uncertainty in DQN Algorithm. They compare MC-Dropout, Bootstrapped and Bootstrapped with prior functions. They also address the problem of alternative valid predictions of the model agent. Unfortunately, they do not dig into details when it comes to the uncertainty estimation in those cases but rather calculate an overall estimate for the epoch.

Hoel *et al.* [2020] estimate uncertainty for RL based on ensembles with randomized prior functions (RPF). They are based on Osband *et al.* [2018] and propose a criterion function. They choose safe actions in unknown situations far from the training distribution. In Hoel *et al.* [2021] they also utilize an ensemble of DQN agents to estimate Q-value uncertainty to switch back to a fallback policy in uncertain situations given a certain threshold. An ensemble is trained on bootstrapped data, which provides a distribution over the estimated Q-values to provide a Bayesian estimation of the epistemic uncertainty. The epistemic uncertainty estimate is then used to choose less risky actions in unknown situations. However, they don't take into account a potential overlapping uncertainty due to possible alternative actions. Rotman *et al.* [2020] investigate different UE methods for detection during deployment, which they call the online safety assurance problem (OSAP). The method for policy-based UE via agent ensembles is also comparable to our baseline UE method.

3 Background

3.1 Reinforcement Learning and MDP

In Reinforcement Learning, we consider an agent that sequentially interacts with an environment modeled as a Markov Decision Process (MDP). An MDP is a tuple $\mathcal{M} := (\mathcal{S}, \mathcal{A}, R, P, \mu_0)$ composed by the set of states $\mathcal{S}$, the set of actions $\mathcal{A}$, the reward function $\mathcal{R}$, the reward function $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto \mathbb{R}$, the transition probability function $P : (\mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto [0, 1]$ which describes the system dynamics, and the starting state distribution μ_0. The transition probability function $P(s_{t+1}|s_t, a_t)$ models the probability of transitioning to a state s_{t+1} given a previous state s_t and taking the action a_t. The reward function represents the return as sum of the discounted reward with γ^k being the discount factor at time steps k, given by

$$R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}. \tag{1}$$

In the MDP framework, at each timestep, the agent observes the current state, takes an action, transitions to the next state drawn from the distribution, and receives a reward. The action-value function, also known as the Q-value function, where Q^π represents the expected return when following a policy π (which basically maps states into actions, as shown below.

$$Q^\pi(s, a) = \mathbb{E}[R_t|s_t = s, a_t = a, \pi]. \tag{2}$$

Q-learning, in which a policy is learned using Q-values, is a popular model-free method. Deep Q-networks (DQNs) are an extension of Q-learning that work with neural networks and optimization algorithms commonly used in deep learning, such as gradient descent. To do so, the temporal-different error δ_t can be derived from the Q-value function using the Bellman operator, resulting in the equation below, whereas θ^- and θ are the DQN parameters from the target and the prediction network, respectively.

$$\delta_t = r_t + \gamma \max_a Q\left(s_{t+1}, a; \theta^-\right) - Q\left(s_t, a_t; \theta\right). \tag{3}$$

3.2 Distributional shift and OOD

Distributional shift and OOD are two concepts that are closely related. For instance, Hendrycks and Gimpel [2018] correlate distributional shift to OOD by describing the latter as samples drawn from a different distribution than the training data. When transferring this definition to RL problems, it is important to distinguish distributions that are closer or further away from the training distribution. It is expected that an RL agent would be able to perform well in scenarios that are slightly different from the training ones, i.e., it should be able to generalize, but when the situation is too dissimilar (perhaps at a semantic level) the agent might have its ability to make proper decisions severely affected. A model or agent which is not able to detect a distributional shift may tend to be overconfident in predictions on unseen data and can give unpredictable results for far-from-distribution test data Lütjens *et al.* [2019].

Some authors also show that epistemic uncertainty can be used as a proxy for detecting distributional shifts, e.g., Sedlmeier *et al.* [2019]. Epistemic uncertainty is usually associated with a lack of sufficient data to better infer the underlying system. Clements *et al.* [2020] claim that epistemic uncertainty stems from limited data. However, besides detecting a model's lack of generalization to data close to it epistemic uncertainty can also point to samples that reside far away from the data distribution known to the trained model, as shown in Postels *et al.* [2020].

Defining distributional shift within the RL domain is not trivial Haider *et al.* [2021]. In this paper we assume that distributional shift can be characterized by changes in the system dynamics, more specifically the shift of the distribution over the state transitions given state action pairs between training and test in MDPs, as shown below:

$$\mathrm{P}_{train}[s_t + 1|s_t, a_t] \neq P_{test}[s_t + 1|s_t, a_t]. \tag{4}$$

Additionally, when considering partially observable MDPs (POMDPs) where the system's state cannot be assessed but rather an observation o_t is available to the agent, the shift of the distribution over observations given states has to be taken into account:

$$\mathrm{P}_{train}[o_t|s_t] \neq P_{test}[o_t|s_t]. \tag{5}$$

3.3 Ensemble based uncertainty estimation

As discussed previously, uncertainty estimation often is the utilized method to detect a distributional shift. Detecting a distributional shift for the sake of exploration during training overlaps with the concept of novelty detection. In contrast, applying distributional shift detection during test time is considered as OOD detection in the classical image classification domain, where OOD describes the distributional shift of the data distribution between training and test. This paper focuses on ensemble-based epistemic uncertainty estimation to detect distributional shift and OOD data during test time. An ensemble of trained agents on a subset of the available data will estimate with little variance in well-trained states. When the ensemble members face little trained states, the estimates vary naturally across the members and give a distribution over the estimated Q-values. The variance of the estimated Q-values can be used to quantify the epistemic uncertainty of a decision.

An ensemble of bootstrapped data over DQNs provides a distribution over the estimated Q-values to provide a Bayesian estimation of the epistemic uncertainty. The Q-values will converge to the real values in situations the agent learned sufficiently. In untrained situations, the Q-value estimates will still diverge and the variance will therefore give an estimate of the epistemic uncertainty. In Osband *et al.* [2018] they use random prior functions to introduce diversity in an ensemble of agents trained on bootstrapped data. The expected return is then given by

$$Q_k(s, a) = f\,(s, a; \theta_k) + \beta p(s, a; \hat{\theta}_k), \tag{6}$$

where Q_k is the Q-function of the kth ensemble member, $\hat{\theta}$ are the parameters of the prior function and β is a factor to weight the impact of the prior function.

In Hoel *et al.* [2020] and Hoel *et al.* [2021] they use the variance of the Q-values of the ensemble estimate to derive an uncertainty estimation threshold to invoke a backup policy. With the variance $Var_k[Q_k(s, a)] < \sigma^2$ the policy with threshold can be calculated by

$$\pi_\sigma(s) = \begin{cases} \arg\max_a \mathbb{E}_k[Q_k(s, a)] & \text{if } Var_k[Q_k(s, a)] < \sigma^2, \\ \pi_{backup}(s) & \text{otherwise.} \end{cases} \tag{7}$$

4 Ensemble Uncertainty estimation based on action count variance and delta to ID

4.1 Action count variance uncertainty estimation

The Q-value is a continuous variable where high variance in the predictions means high uncertainty of the ensemble. However, when given encapsulated agents or due to other reasons, where the Q-values are not accessible, it is possible to use the action prediction result from the ensemble members and derive an uncertainty estimate. Instead, it is possible to take the deviation over the proposed actions of an ensemble, to indicate uncertainty. In cases where the action space is continuous, the variance can be directly calculated as action variance like with the Q-values. However, with discrete action spaces, this will lead to false results, as the actions themselves are orthogonal and e.g. a mean action can not be calculated. Therefore, in cases where the action space is discrete, we propose to calculate an action count on each action over the ensemble given a certain state and then calculate the variance of that action count (ACV - action count variance). When the ACV is low, there is a balance in the proposed different actions over the ensemble and the uncertainty is therefore high. In contrast, when the action count variance is high, there is a concentration of one or more actions in the ensemble and the uncertainty is low. The higher the ACV gets, the lower the uncertainty. Given $N_{actions}$ as the number of alternative actions and $K_{ensemble}$ as the number of ensemble members and 8 the minimum of the ACV gets to equation 9 and the maximum to equation 10.

$$Var[AC(s,a)] = \frac{1}{N_{actions}} \sum_n \left(AC_n - \frac{K_{ensemble}}{N_{actions}} \right)^2 \tag{8}$$

$$Var_{min}[AC(s,a)] = 0 \tag{9}$$

$$Var_{max}[AC(s,a)] = \frac{K^2_{ensemble}}{N^2_{actions}} \left(N_{actions} - 1 \right) \tag{10}$$

A backup policy can then be chosen based on the ACV calculation like given in equation 11.

$$\pi_\sigma(s) = \begin{cases} \arg\max_a \mathbb{E}_k[Q_k(s,a)] & \text{if } Var_k[AC_k(s,a)] > Var_{threshold}, \\ \pi_{backup}(s) & \text{otherwise.} \end{cases} \tag{11}$$

4.2 Delta to ID uncertainty estimation

One problem with uncertainty estimation within reinforcement learning is the fact, that there often are alternative decisions to take in an MDP. These can be called alternative possible actions. When an agent is in a state with alternative possible actions, the ensemble may already deviate in its prediction, although trained sufficiently in this state (ID - in-distribution). This means alternative possible actions will sort of overlay the uncertainty from a lack of training (OOD). Given an overlay of such kind will make it hard to decide if real uncertainty, only alternative possible actions or both are in place. Therefore, we propose to "compare" the given (potentially OOD) situation to the nearest ID situation (IDD - ID delta), to identify "normal" deviation versus uncertainty deviation. To get a comparison, we propose to subtract ID ACV from the given AVC and use the result as a cleaned version of the ACV for uncertainty indication.

$$Var_{delta}[AC(s,a)] == Var_{OOD}[AC(s,a)] - Var_{ID}[AC(s,a)]. \tag{12}$$

There can be different approaches to obtaining the nearest ID from a given OOD scenario. To simplify, we stick to an OOD scenario with one dedicated OOD Object. We propose two variations as a straightforward approach. At first, the OOD object in the given OOD scenario (observation) will be exchanged with a corresponding ID object - like a known obstacle. Then, the given OOD object will just be removed from the scenario without replacement - or with zero replacement, so to say.

$$Obs_{ID_{obstacle}}(pos_{OODobstacle}) == ID_{obstacle}, \tag{13}$$

$$Obs_{ID_{zero}}(pos_{OODobstacle}) == ID_{zero}. \tag{14}$$

A high delta of the ACV will indicate high uncertainty and a low delta is associated with low uncertainty in both cases, respectively. To decide on an uncertain situation in a given state, we propose to use a threshold to mask out insignificant variance-delta to ID. This threshold can be used in future work to switch to a backup policy.

5 Experimental results

5.1 Setup and Training

We trained complete agents in parallel with a randomly placed set of 10 hazards singly placed in a gridwold of 10x10 positions and training runs of 1 million steps for each agent. For testing, we set up different scenarios with hazards used for training as ID and added a single novel obstacle as ODD. For the results shown below, we focused solely on ID scenarios with a line of known hazards in the middle of the grid and the goal at the end of the line.

For the visualization of the uncertainty estimation, we calculated heatmaps over the grid showing each resulting uncertainty estimation for each position of the agent in the grid given the overall scenario.

5.2 Uncertainty heatmaps

For the depicted results the uncertainty calculation based on the action count variance of the ensemble members is used. Figure 1 is the base scenario with the known ID hazard line in blue and the goal in green. As we use variance in the action count, a higher brighter color means more concentration on fewer actions (and therefore more certainty) and darker color means less concentration in the actions or more equally distributed actions (and therefore higher uncertainty. As one can see in the base scenario - due to possible alternative action predictions there are some "uncertainties" along the diagonals to the goal, as these coordinates have equal probabilities vertically and horizontally to approach the goal. This applies e.g. also for the point on the left of the hazard line, as the probabilities for up and down are equally distributed.

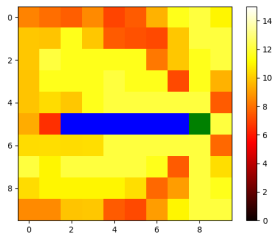
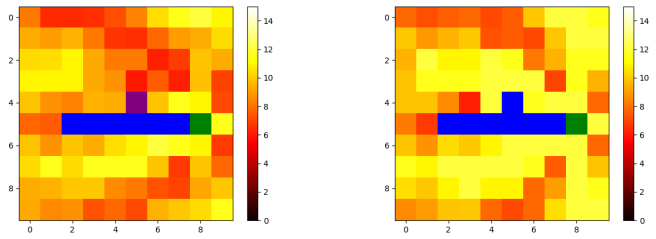


Figure 1: ID Hazard Line as baseline ID scenario.

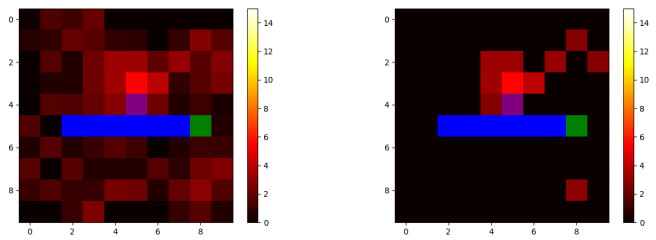
Figure 2a shows the predictions with one unknown obstacle inserted in the middle direct on top of the line shown in purple. One can see the tendency of many of the areas, and especially the area around the unknown obstacle, to shown an increase in uncertainty. Nevertheless, the uncertainty indication is superposed by the already given "uncertainty" of the possible alternative predictions from the baseline ID scenario. In contrast, figure 2b shows the predictions with a known obstacle inserted in the middle direct on top of the line shown in blue, instead of the OOD obstacle. Now, the uncertainty indication is nearer to the baseline ID scenario.

Our idea presented in 4.2 proposes to subtract the baseline variance from the OOD variance and therefore try to eliminate the baseline variance resulting from the possible alternative predictions. In Figure 3 the results are depicted for the delta to the ID baseline without and with a threshold (3a and 3b) and the delta to ID with the known obstacle (3c and 3d). It seems feasible to indicate the given OOD hotspot when masking out a dedicated threshold, like given in 3b and 3d, although the indication is not totally sharp.

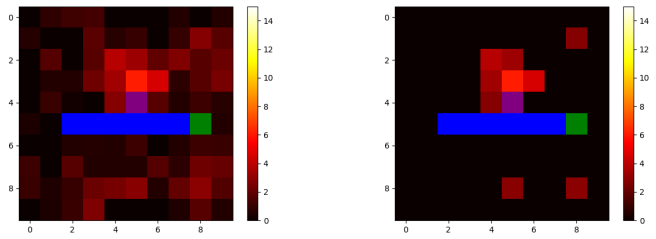


(a) OOD obstacle at the top (b) ID Hazard at the top

Figure 2: ID Hazard Line with obstacle in the middle



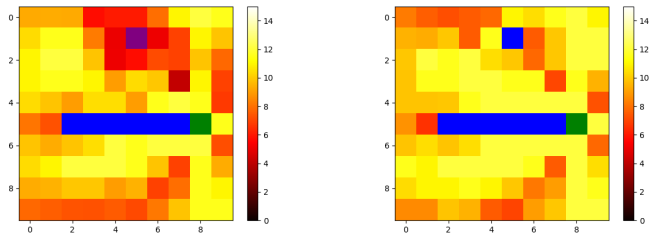
(a) Delta to ID baseline (b) Delta to ID with threshold



(c) Delta to ID known obstacle (d) Delta to ID known obstacle with threshold

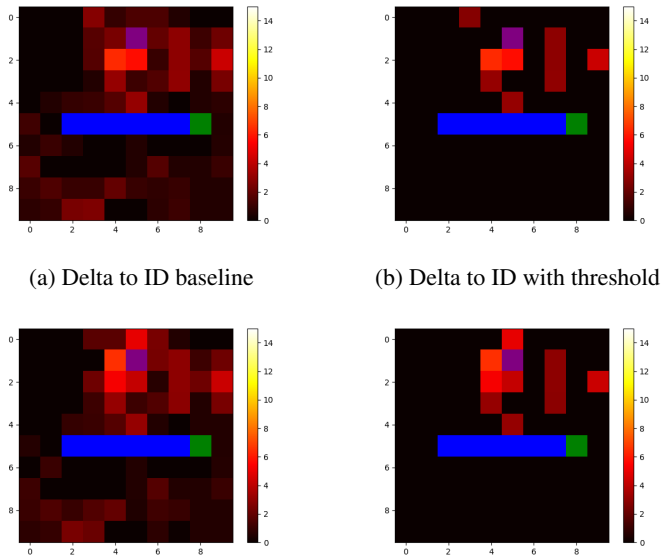
Figure 3: Delta OOD to ID with obstacle in the middle

In the given scenario, the OOD hotspot is lying direct in an area of low uncertainty (the yellow area on top of the blue line). One can think of a special situation, where the ID subtraction is convenient to isolate the hotspot. Therefore, we also ran setups, where the hotspot is lying in an area of already given "uncertainty" from possible alternative predictions - like in the upper middle section. See Figure 4.



(a) OOD obstacle at the top (b) ID Hazard at the top

Figure 4: ID Hazard Line with obstacle at the top



(a) Delta to ID baseline

(b) Delta to ID with threshold

(c) Delta to ID known obstacle

(d) Delta to ID known obstacle with threshold

Figure 5: a) ID Hazard Line with obstacle at the top

And Figure 5 shows the resulting indication for the delta OOD to ID baseline in 5a and 5b - and the delta to ID with known hazard in 5c and 5d. Again, the given threshold seems to isolate the hotspot from the OOD obstacle quite well. But here, the delta to ID baseline fails, as the top and left fields from the obstacle are not indicated. The already higher "uncertainty" from the possible alternative predictions in that area subtracts out the indication. But actually, these fields should be indicated - because they lie in the diagonal path to the goal and would therefore be conflicting due to the unknown obstacle. In contrast, the delta to ID with known obstacle replacement performs better and indicates the respective fields.

6 Conclusion

Within Reinforcement Learning for a decision-making agent aleatoric and epistemic uncertainty can be distinguished. Aleatoric uncertainty refers to the inherent randomness of an outcome whereas epistemic uncertainty arises due to a lack of knowledge. In this paper, we investigated ensemble-based epistemic uncertainty estimation on gridworld scenarios with discrete action spaces and overlapping alternative predictions.

We exhibited the problem obstacle with discrete action spaces and variance calculation and proposed to use action count variance (ACV) instead of action variance. We depicted the conflict of uncertainty estimation due to a lack of training with overlapping alternative solutions. As a viable solution, we proposed to calculate the delta to ID (IDD) for the action count variance and utilized a threshold to indicate uncertainty due to a lack of training, only. With experiments utilizing representative gridworld test scenarios, we could show that action count variance with delta to ID is able to indicate uncertain states based on a threshold calculation with high probability. Therefore, a decision for a backup policy based on that indication seems to be a feasible solution.

In future work, we want to investigate feasible methods to determine a sufficient near ID scenario for a given OOD scenario and extend the approach to more general and realistic environments. We also want to focus on strategies to deal with an uncertainty indication and compare the resulting performance to other given approaches.

References

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep Reinforcement Learning in a Handful of Trials using Probabilistic Dynamics Models. *arXiv:1805.12114*, 2018.

William R. Clements, Bastien Van Delft, Benoît-Marie Robaglia, Reda Bahi Slaoui, and Sébastien Toth. Estimating Risk and Uncertainty in Deep Reinforcement Learning. September 2020.

Felipe Leno Da Silva, Pablo Hernandez-Leal, Bilal Kartal, and Matthew E Taylor. Uncertainty-aware action advising for deep reinforcement learning agents. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 5792–5799, 2020.

Terrance DeVries and Graham W Taylor. Learning confidence for out-of-distribution detection in neural networks. *arXiv preprint arXiv:1802.04865*, 2018.

Tom Haider, Felippe Schmoeller Roza, Dirk Eilers, Karsten Roscher, and Stephan Günnemann. Domain shifts in reinforcement learning: Identifying disturbances in environments. In *AISafety@ IJCAI*, 2021.

Dan Hendrycks and Kevin Gimpel. A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks. *arXiv:1610.02136 [cs]*, October 2018.

Carl-Johan Hoel, Krister Wolff, and Leo Laine. Tactical Decision-Making in Autonomous Driving by Reinforcement Learning with Uncertainty Estimation. April 2020.

Carl-Johan Hoel, Krister Wolff, and Leo Laine. Ensemble quantile networks: Uncertainty-aware reinforcement learning with applications in autonomous driving. 2021.

Gregory Kahn, Adam Villaflor, Vitchyr Pong, Pieter Abbeel, and Sergey Levine. Uncertainty-Aware Reinforcement Learning for Collision Avoidance. February 2017.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks. *arXiv:1807.03888 [cs, stat]*, October 2018.

Björn Lütjens, Michael Everett, and Jonathan P How. Safe reinforcement learning with model uncertainty estimates. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8662–8668. IEEE, 2019.

Sina Mohseni, Mandar Pitale, Jbs Yadawa, and Zhangyang Wang. Self-Supervised Learning for Generalizable Out-of-Distribution Detection. In *AAAI*, 2020.

Ian Osband, John Aslanides, and Albin Cassirer. Randomized Prior Functions for Deep Reinforcement Learning. 2018.

M. Pimentel, D. Clifton, L. Clifton, and L. Tarassenko. A review of novelty detection. *Signal Process.*, 2014.

Janis Postels, Hermann Blum, Yannick Strümpler, Cesar Cadena, Roland Siegwart, Luc Van Gool, and Federico Tombari. The Hidden Uncertainty in a Neural Networks Activations. 2020.

Noga H. Rotman, Michael Schapira, and Aviv Tamar. Online safety assurance for deep reinforcement learning, 2020.

Adrian Schwaiger, Poulami Sinhamahapatra, Jens Gansloser, and Karsten Roscher. Is Uncertainty Quantification in Deep Learning Sufficient for Out-of-Distribution Detection? In *Proc. AISafety@IJCAI2020*, volume 2640 of *CEUR Workshop Proceedings*, page 8, 2020.

Andreas Sedlmeier, Thomas Gabor, Thomy Phan, Lenz Belzner, and Claudia Linnhoff-Popien. Uncertainty-based out-of-distribution classification in deep reinforcement learning. *arXiv preprint arXiv:2001.00496*, 2019.

Qing Yu and Kiyoharu Aizawa. Unsupervised Out-of-Distribution Detection by Maximum Classifier Discrepancy. *arXiv:1908.04951 [cs]*, August 2019.