Safety Assurance with Ensemble-based Uncertainty Estimation and overlapping alternative Predictions in Reinforcement Learning

Dirk Eilers[1,*,1], Simon Burton[1,1], Felippe Schmoeller Roza[1] and Karsten Roscher[1]

[1]Fraunhofer Institute for Cognitive Systems IKS, Fraunhofer Gesellschaft, Munich, Germany

Abstract

A number of challenges are associated with the use of machine learning technologies in safety-related applications. These include the difficulty of specifying adequately safe behaviour in complex environments (specification uncertainty), ensuring a predictably safe behaviour under all operating conditions (technical uncertainty) and arguing that the safety goals of the system have been met with sufficient confidence (assurance uncertainty). An assurance argument is therefore required that demonstrates that the effects of these uncertainties do not lead to an unacceptable level of risk during operation. A reinforcement learning model will predict an action in whatever state it is in - even in previously unseen states for which a valid (safe) outcome cannot be determined due to lack of training. Uncertainty estimation is a well understood approach in machine learning to identify states with a high probability of an invalid action due a lack of training experience, thus addressing technical uncertainty. However, the impact of alternative possible predictions which may be equally valid (and represent a safe state) in estimating uncertainty in reinforcement learning is not so clear and to our knowledge, not so well documented in current literature. In this paper we build on work where we investigated uncertainty estimation on simplified scenarios in a gridworld environment. Using model ensemble-based uncertainty estimation we proposed an algorithm based on action count variance to deal with discrete action spaces whilst considering in-distribution action variance calculation to handle the overlap with alternative predictions. The method indicates potentially unsafe states when the agent is near out-of-distribution elements and can distinguish it from overlapping alternative, but equally valid predictions. Here, we present these results within the context of a safety assurance framework and highlight the activities and evidences required to build a convincing safety argument. We show that our previous approach is able to act as an external observer and can fulfil the requirements of an assurance argumentation for systems based on machine learning with ontological uncertainty.

Keywords

Safe Reinforcement Learning (Safe RL), Safety Assurance Argumentation, Distributional Shift, Ensemble-based Uncertainty Estimation, Out-of-Distribution (OOD) detection

1. Introduction

The application of Machine Learning (ML) to safety-critical cyber-physical systems such as industrial robots and automated vehicles has the potential for greatly increasing the level of automation in complex environments. However, the use of ML is met with many practical challenges, in particular regarding resource, timing and performance constraints. The most dominant obstacle to the deployment of such systems is the difficulty in demonstrating the absence of unreasonable risk of unsafe actions due to erroneous outputs of the ML model. These errors are caused by a combination of insufficiencies due by epistemic uncertainty in the model and the occurrence of inputs and states that uncover these insufficiencies, themselves subject to aleatoric uncertainty. [1] argued that a causal understanding of insufficiencies can be used to reduce uncertainties in the performance of ML in an iterative manner. Based on a specification of safety acceptance criteria, a measurement of the error rate of the ML function is used to evaluate the impact and potential causes of ML insufficiencies. This analysis is used to derive design-time and operation-time measures to reduce residual safety risk (Figure 1). Design-time measures reduce the occurrence of insufficiencies in the model, e.g. by restricting the scope of the operating environment, optimizing the ML technique and architecture or redefining training conditions. Operation-time measures reduce the impact of residual insufficiencies in the model, e.g. through plausibility analysis or heterogeneously redundant calculations of the target function.

Reinforcement learning (RL) is well suited to systems operating in complex environments with high demands on flexibility such as in route planing or motion control of mobile robots. As an RL agent will predict an action in whatever state it finds itself, the application can benefit from an awareness of the certainty and confidence in its own decisions. This includes situations that fall both within as well as outside of the distribution of previously seen training data. This paper focuses on uncertainty estimation as an operation-time measure to detect states which could lead to errors in the ML model. Specifically,

The AAAI-23 Workshop on Artificial Intelligence Safety (SafeAI 2023)

*Corresponding author.

dirk.eilers@iks.fraunhofer.de (D. Eilers)

https://iks.fraunhofer.de/ (D. Eilers)

[1]These authors contributed equally.

Copyright © 2023 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

CEUR-WS.org/Vol-3381/35.pdf

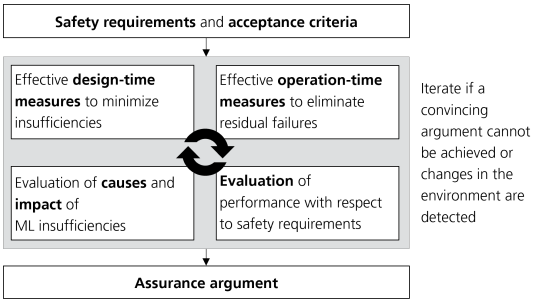


Figure 1: Safety Assurance Framework for machine learning systems (adapted from [1])

we evaluate the detection of out-of-distribution (OOD) inputs to address the impact of distributional shift.

Distributional shift in data science is widely understood as the distributional difference between training and test data (respectively data used during the inference or deployment phase) [2][3][4][5]. Distributional shift can have different causes, such as natural perturbations to the data-set due to aleatoric uncertainty as well as evolving conditions in the environment. In machine learning, a shift in the probability distribution over state-action pairs often leads to degraded performance in the inference phase, leading the agent to propose wrong or sub-optimal actions. When the testing distribution differs from the training distribution, machine learning systems may not only demonstrate poor performance, but also have false confidence in the validity of their actions.

To overcome this limitation, safe reinforcement learning (safe RL) solutions must be capable of detecting and handling the uncertainty in the decision-making process. 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, a set of alternatively trained agents (ensemble) can be used. In states with high uncertainty due to a lack of training, the different agents will likely predict different actions, due to a lack of substance of the prediction. This variance can be utilized to indicate uncertainty. However, there may be states with various, equally valid actions that would also result in a variance in the outputs of the ensemble. Therefore, it is necessary to differentiate between the two effects.

2. Related work

Recent work has addressed OOD detection in the classical image classification domain as well as some work in the RL domain. [6] define novelty detection as the classification of test data that differ in some respect from data available during training. A model with insufficient training data is less able to generalize based on unseen test data even if this data does not contain "novel" concepts. Similarly, [2] characterize OOD as test data, which are from a different distribution as the training data and describe OOD detection as a threshold–based process. The closer the data are to the training data distribution the more likely it is that this is caused by a lack of training data only. OOD data further away from the training data distribution are more likely to represent conceptual or semantic differences, such as samples which are completely outside of the given classifications. Deep neural networks (DNNs) tend to be overconfident in predictions on unseen data and can give unpredictable results for far-from-distribution test data [3].

Prior work focused on OOD as a concept of samples that fall outside the defined set of classes. If samples are from outside this set, correct classifications for these samples, by definition, cannot be learned, even with unlimited training. To address this issue, it is common to specify a separate OOD class to train the model on [6]. Similarly, [7] and [8] define OOD samples as examples for classes different from those in the in-distribution (ID) dataset. [4] describe ID as a distribution trained by a classifier and OOD as sufficiently different from it. Also, [9] follow the approach of considering a strong difference between training and test data to be OOD. They describe ID data as conceptually similar to training data and OOD data as differing strongly from training data. [10] go as far as to define OOD by the distributional gap in between classified ID data sets. 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 near and far from the distribution.

Furthermore, it is important to understand the difference of 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 [11]. It can indicate samples that reside far away as well as close to the data distribution [5]. In contrast, aleatoric uncertainty arises from stochastic environments and must be accounted for in risk-sensitive applications [12], [13]. Aleatoric uncertainty cannot be solved just by more training. The impact of aleatoric uncertainty is therefore a significant factor in arguing the safety of RL-based safety-critical applications. In [14] the authors propose ensemble quantile networks (EQN) based on the work of [15] where they combine implicit quantile networks (IQN) for aleatoric uncertainty detection and utilize random prior functions (RPF) [16] as an ensemble based method for epistemic uncertainty estimation. As epistemic uncertainty originates from model insufficiencies, a model ensemble will output a distribution over different estimates in an uncertain state

(distribution over outputs). Aleatoric uncertainty arises from the randomness in the environment and causes a distribution over returns from the environment to the models input (distribution over returns/inputs). However, in this paper we focus on the detection of epistemic uncertainty, as we focus on distributional shift and OOD detection.

[17] investigate a distributional RL algorithm D3PG, which models the uncertainty in the form of a return distribution in which the expected value is the Q-value. Different actions might be used when the distribution is bimodal or multimodal depending on the application scenario. [18] 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 [19] the authors 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 deep Q-learning (DQN). As the learning algorithm updates the network, their predictions get closer to the real function, and one close to the others. [11] 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 overlapping alternative (equally valid) predictions of the model agent. Unfortunately, they do not dig into detail, when it comes to the uncertainty estimation in those cases but rather calculate an overall estimate for the epoch.

[20] estimate uncertainty for RL based on ensembles with randomized prior functions (RPF). They are based on [16] and propose a criterion function. They choose safe actions in unknown situations far from the training distribution. In [14] 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 be 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.

One of the key questions to be answered from a safety assurance perspective is how does the OOD detection as an operation-time measure really help to prevent hazardous conditions? This requires OOD detection to be able to detect novel data that relates to an increase of risk, and to do so with sufficient accuracy and timeliness so that the system can be brought into a safe state before the risk becomes unacceptable.

The interaction between the development of ML specific methods for optimizing performance and safety assurance was not observed in much previous work. In [21] the authors describe a collaborative and iterative process where ML method developers are supported by safety engineers to ensure the method contributes to the overall system safety assurance argument. This includes the systematic argumentation of the effectiveness of design and operation-time measures, an evaluation of the performance of the ML function against quantitative safety acceptance criteria and an analysis of the causes of insufficiencies in the model in order to derive more effective design and operation-time methods. Nevertheless, uncertainties in the assurance of the safety of ML functions will remain.

The closed-box nature of ML algorithms and the consequent reliance on observational evidence coupled with the inherent epistemic uncertainty of the models (compared to traditional software) whilst operating within an environment with high aleatoric uncertainty lead to a lack of confidence in our statements about the safety of the resulting system (assurance uncertainty). [1] refers to this challenge as the need to *infer* certain safety claims based on incomplete observations and defines a set of conditions to formalise this statement. This requires the use of rigorous argumentation to justify why an acceptable level of safety can be asserted despite the inherent limitations in the available evidence. In [22], based on [23], Baconian probability is proposed as a concept for estimating confidence in assurance arguments [24] based on how many possible assurance deficits (known as "defeaters") of an argument can be eliminated. Confidence claim patterns were introduced which aim to identify all possible defeaters and demonstrate that they are either unlikely or not of significance. In section 4 we return to the challenge of arguing the safety contribution of an uncertainty estimation-based OOD detector by highlighting some of the possible defeaters to the safety argument can be identified and addressed as part of an iterative process.

3. Ensemble uncertainty estimation based on action count variance and delta to ID

In [25] we proposed uncertainty estimation with action count variance (ACV) and delta to ID (IDD). In the next section we will give a slightly reduced repetition.

3.1. Background

3.1.1. Reinforcement learning and MDP

In RL, the goal is to find the best policy for an agent that makes sequential decisions while interacting with an environment modeled as a Markov decision process (MDP). An MDP is defined as 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 $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto \mathbb{R}$, the transition probability function $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto [0, 1]$, and the starting state distribution μ_0. The transition probability function $P(s_{t+1}|s_t, a_t)$ models the system dynamics by mapping the probability of transitioning from a previous state s_t to the state s_{t+1} when 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) extend Q-learning with the usage of neural networks as function approximators. To do so, the temporal-difference error δ_t can be derived from the Q-value function using the Bellman operator, resulting in the equation below.

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

where θ^- and θ are the DQN parameters from the target and the prediction network as defined in [26], respectively.

3.1.2. Distributional shift and OOD

Distributional shift and OOD are two concepts that are closely related, but 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 those used in training, as it should be able to generalize. However, when the situation is too dissimilar (perhaps at a semantic level) the agent might have its ability to make proper decisions severely affected. Epistemic uncertainty can be used as a proxy for detecting distributional shifts and is usually associated with a lack of sufficient data to better infer the underlying system.

Defining distributional shift within the RL domain is not trivial [27]. 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:

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

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:

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

3.1.3. Ensemble-based uncertainty estimation

We focus on ensemble-based epistemic uncertainty estimation to detect distributional shift and OOD data during test time respectively during the inference or deployment phase. An ensemble of trained agents on a subset of the available data will estimate with low variance in well trained states. When the ensemble members face too few 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 on 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 sufficiently learned. In untrained situations, the Q-value estimates will still diverge and the variance will therefore give an estimate of the epistemic uncertainty.

Random prior functions can be used to introduce diversity in an ensemble of agents trained on bootstrapped data [16]. The expected return is then given by

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

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

The variance of the Q-values of the ensemble estimates can be used to derive an uncertainty estimation threshold to invoke a backup policy [20] [14] that ensures a safe state. With the variance $Var_k[Q_k(s, a)] < \sigma^2$ the policy

with threshold can be calculated by

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

3.2. 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 when the Q-values are not accessible due to other reasons, it is possible to take the deviation over the proposed actions of the ensemble members, 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 a mean action can not be calculated. Therefore, in cases where the action space is discrete, we proposed in [25] 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. A backup policy can then be chosen based on the ACV calculation as given in equation 8.

$$\pi_\sigma(s) = \begin{cases} \arg\max_a \mathbb{E}_k[Q_k(s,a)] & \text{if } Var_k[AC_k(s,a)] \\ & > Var_{threshold}, \\ \pi_{backup}(s) & \text{otherwise.} \end{cases} \tag{8}$$

3.3. Delta to ID uncertainty estimation

One problem with uncertainty estimation within reinforcement learning is that often multiple decisions are equally valid in a given state. These can be called alternative possible actions - or more generally alternative predictions. When an agent is in a state with alternative possible actions, the ensemble may already deviate in its prediction, although it might be trained sufficiently in this state. This means, alternative possible actions will pose high uncertainty and might falsely flag an OOD instance. Traditional methods fail to distinguish these two cases and, therefore, in [25] an alternative solution was proposed. This method, called Delta to ID (IDD), consists in comparing the given (and potentially OOD) situation to its nearest ID counterpart to differentiate high uncertainty resulting from these "ambiguous" states from distributional shifts. To get a comparison, it was proposed to subtract the ID uncertainty (represented by the ACV) from the given OOD uncertainty and use the result as a cleaned (delta) version of the ACV for uncertainty indication. Because of the (1-x) characteristic of the ACV to the uncertainty, we actually subtract $(Const_{maxVar} - ACV_{OOD}) - (Const_{maxVar} - ACV_{ID})$ which inverses the ACV characteristic to match the uncertainty's and results for the subtraction in:

$$Var_{delta}[AC(s,a)] = Var_{ID}[AC(s,a)] - Var_{OOD}[AC(s,a)]. \tag{9}$$

There can be different approaches to get a nearest ID from a given OOD scenario. To simplify here, we stick to an OOD scenario with one dedicated OOD obstacle. The OOD obstacle in the given OOD scenario will then be exchanged with a corresponding ID obstacle. The observation function $Obs()$ changes as given in equation 10.

$$Obs_{ID_{nearest\_with\_obstacle}}(pos_{OOD_{obstacle}}) = ID_{obstacle}. \tag{10}$$

A high delta of the ACV will indicate high uncertainty and a low delta low uncertainty in both cases, respectively. To decide on an uncertain situation in a given state, we proposed 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 as operation-time measure for safety assurance methods e.g. in an iterative causal model like proposed in [1] and we will come back to in section 4.

4. Safety assurance

4.1. Background

The use of ML for highly automated safety-critical applications leads to a number of safety assurance challenges. These challenges are related to the complexity and unpredictability of the operating environment (aleatoric uncertainty), as well as the complexity of the technical system and task itself. A complex system can be defined as system that exhibits behaviours that are *emergent* properties of the interactions between the parts of the system, where the behaviours would not be predicted based on *knowledge* of the parts and their interactions alone. This definition is closely related to the general concept of uncertainty, defined as *any deviation from the unachievable*

ideal of completely deterministic knowledge of the relevant system [28].

For safety-critical autonomous systems, uncertainty manifests itself in various forms not restricted to the narrow definitions used in ML. *Specification uncertainty* is the uncertainty in the appropriateness and completeness of safety acceptance criteria and the definition of acceptably safe behavior in all situations that can reasonably be anticipated to occur within the target environment. Incomplete, or otherwise insufficient training data can be seen as a consequence of specification uncertainty. *Technical uncertainty* stems from a lack of predictability in the performance of the technical components of a system. An example of which is the unpredictable reaction of the system to previously unseen events, or differences in the system behavior despite similar input conditions (epistemic uncertainty in the trained model). *Assurance uncertainty* relates to lack of confidence in claims regarding safety properties of the ML system. This can include a insufficient integrity of evidence supporting the assurance arguments as well as the chain of reasoning itself. Safety assurance for ML-based systems must therefore minimise these uncertainties and thus maximise the confidence that the system fulfils its safety expectations. The approaches described in [1, 21] and summarised in Figure 1 are designed to iteratively minimise these uncertainties and thereby safety risk as part of a continuous assurance process based on an understanding of the environment, insufficiencies in the ML system and potential deficits in the safety assurance argumentation. To support this approach an assurance argument is proposed to support a systematic evaluation that well defined safety claims are supported by evidence and that all assumptions are explicitly stated and validated.

Complexity and unpredictability of the operational domain and of the system itself lead to semantic gaps, which indicate discrepancies between the intended and specified functionality, also known as specification insufficiencies. In safety-critical systems this can lead to hazardous systemic failures. From our consideration, specification uncertainty is also a problem in RL, for example when inappropriate reward functions are used. This might manifest itself in a manner that appears to be epistemic uncertainty, the root cause is however subtly different to, for example, a lack of training data.

To better understand the characteristics and impact of uncertainty, one can differentiate between statistical, scenario and ontological uncertainty. Statistical uncertainty can be expressed in quantitative statistical terms, such as confidence intervals expressed over probability distributions. Scenario uncertainty can only be described using qualitative scenarios, which are potentially multiple plausible states of the system and its environment. Ontological uncertainty [29] defines a lack of awareness that the knowledge about the system itself is incomplete - this requires an external perspective to resolve. Ontological uncertainty is a specific cause of specification insufficiencies, which in turn will lead to epistemic uncertainty in the trained model. In this paper, we describe an operation-time measure to mitigate the effects of this uncertainty by introducing an observer external to the ML component to detect the conditions where previously unseen inputs might impact the safety requirements.

4.2. Safety assurance argumentation using ensemble-based uncertainty estimation and IDD

In this section we discuss the impact of the Ensemble-based uncertainty estimation from the following perspectives. First we discuss the role of the uncertainty estimation as an operation-time measure for mitigating the impact of residual errors in the ML component and how this supports a safety assurance argument for the function. Second we examine issues of uncertainty in the assurance argument itself and how confidence in the argument can be increased.

Figure 2 shows a simplified and incomplete excerpt (inspired by [30]) of a safety assurance argument described using the Goal Structuring Notation (GSN) [31][32] for the claim that the residual risk of the system colliding with obstacles is sufficiently low. GSN is a graphical notation that represents the elements of an assurance argument and the relationships between them. It shows how goals (claims) can be broken into sub-goals until they can be supported by direct references to evidence. It documents argumentation strategies as well as the context information, including assumptions and justifications. The assurance strategy illustrated here is based on an identification of potential causes of insufficiencies in the function and measures for reducing their impact during development and operation.

Uncertainty estimation is one of a number of complementary measures used to form a broad argument for safety. However, as mentioned above, the complexity of the system can undermine the confidence in the argument. [23] describes confidence in assurance arguments in terms of trust in assertions related to the evidence, context (including assumptions) and inference (or structure of the argument itself). For each of these aspects a number of defeaters could potentially be identified that undermine the argument [22]. For the example argumentation in Figure 2 these can include an incomplete definition of operating environment or incorrect assumptions regarding the performance of the perception components (asserted context), as well as the validity of test results demonstrating the generalisation performance of the trained function due to the difficulty in covering previously unknown corner cases (asserted evidence).

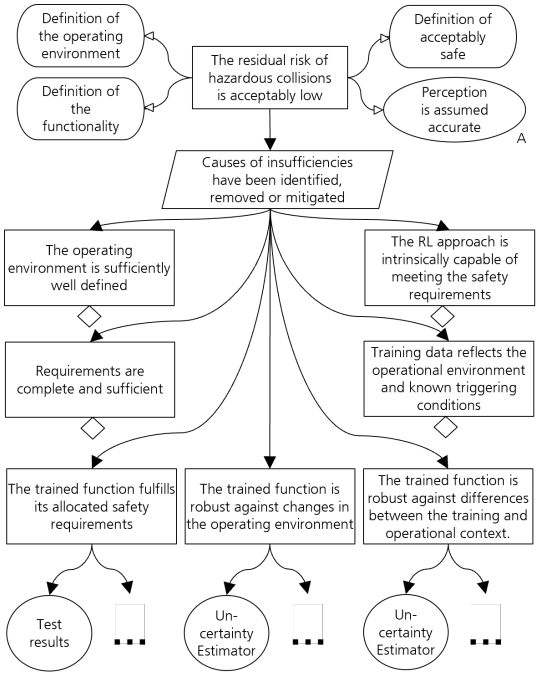


Figure 2: Partial GSN description of the assurance argumentation for of the gridworld agent with IDD uncertainty estimator

Furthermore, the assertion that all possible causes of insufficiencies have been addressed could also be incorrect (asserted inference).

As proposed in [1], it is advantageous to iterate through the assurance process when dealing with ontological uncertainties - and we can show in the following, that this also is beneficial even with our simplified case study. The addition of the uncertainty estimator was the initial step to mitigate against the residual uncertainties in the assurance argument with the extension of the IDD a further step to increase confidence in the effectiveness of the uncertainty estimation itself.

5. Experimental results

In [25] we presented the results from more extended experiments. Here, we summarise the results and conduct additional experiments to argue the safety assurance.

Setup and Training: We trained complete agents in parallel with a randomly placed set of 10 obstacles singly placed in a gridworld of 10x10 positions and training runs of 1 million steps each agent. For testing we set up different scenarios with previously seen obstacles as ID and added a single dedicated obstacle not seen during training as an OOD condition. For the paper we focused on an ID scenario with a line of known obstacles 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.

Uncertainty heatmaps: For the depicted results, the uncertainty calculation based on the action count variance of the ensemble members is used. Figure 3 is the base scenario with the known ID obstacle line in blue and the goal in green. As we use variance in the action count, a higher brighter colour means more concentration on fewer actions (and therefore more certainty) and darker colour means a less concentration in the actions or more equally distributed action (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, since the action space only allows for up/down and left/right movement and cannot realize a diagonal path directly. This shows the limits of the uncertainty metric here as well - the actions along the diagonal are no more or less dangerous but they are monitoring a high “uncertainty”. This consideration applies e.g. also for the point on the left of the obstacle line, as the probabilities for up and down are equally distributed.

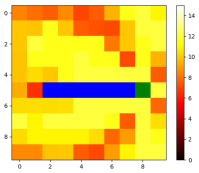
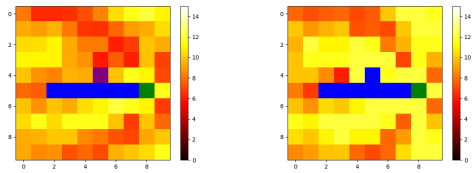


Figure 3: Obstacle line ID scenario as heatmap over agent positions

Figure 4a shows the predictions with one unknown obstacle inserted in the middle direct on top of the line shown in purple. There is increased uncertainty, especially in the area surrounding the unknown obstacle. Nevertheless, the uncertainty indication is superposed by the already given "uncertainty" of the possible alternative predictions from the base ID scenario. In contrast, figure 4b 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 much closer to the base ID scenario.

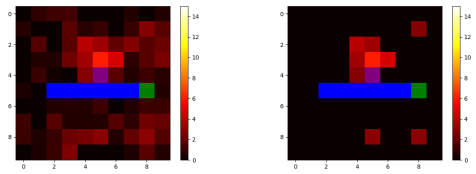
Our approach proposed to subtract the base variance from the OOD variance and therefore try to eliminate the base variance resulting from the possible alternative predictions. In Figure 5 the results are depicted for the delta to ID with the known obstacle without and with threshold (5a and 5b). It seems to feasibly indicate a



(a) OOD obstacle in the middle (b) ID obstacle in the middle

Figure 4: ID obstacle line with OOD obstacle in the middle

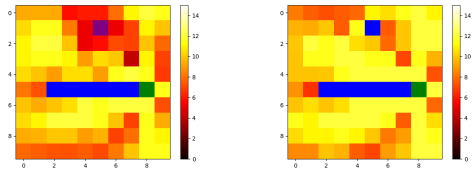
given OOD hotspot considering a dedicated threshold, although the indication is not totally sharp.



(a) Delta to ID known obstacle (b) Delta to ID known obstacle with threshold

Figure 5: Delta OOD to ID with obstacle in the middle

In the given scenario, the OOD hotspot lies directly in an area of low uncertainty (the yellow area on top of the blue line). In order to validate that the approach generalizes to different scenarios, we also ran setups where the hotspot lies in an area of previously known uncertainty from possible alternative predictions - such as in the upper middle section (see Figure 6).

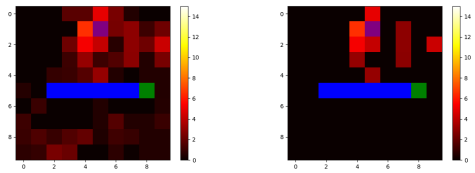


(a) OOD obstacle at the top (b) ID obstacle at the top

Figure 6: ID obstacle line with OOD obstacle at the top

Figure 7 shows the resulting indication for the delta to ID with known obstacle in 7a and 7b.

False-Positive and False-Negative rates: In order to argue the safety assurance, we set up an additional experiment and measure the probability of an agent without an observer to hit the unknown obstacle and compared this to the probability of an agent with only baseline uncertainty estimation (UE-BL) and the probabilities with the IDD uncertainty estimator as external observers. The



(a) Delta to ID known obstacle (b) Delta to ID known obstacle threshold

Figure 7: Delta OOD to ID with obstacle at the top

agent without an observer will get no uncertainty estimation (UE) indications which is equivalent to false-negatives (FN). For the two with external observers we assume the agent will follow an alternative route and not hit the unknown obstacle when the uncertainty estimator indicates uncertainty above a given threshold.

We iterate over all possible positions of the unknown obstacle and all possible positions of the agent without introducing randomness in the setup, to focus on the demonstration of the effects here. We calculate the mean probabilities for false-positive (FP) indications (which slow the agent down) and the false-negatives (FN) (which result in hazards). As varying hyper-parameters we use different ACV-thresholds for IDD and UE-BL, a variable sized bounding box around the OOD position wherein each indication is TP (true-positive), for FN the percentage threshold of consent of the ensemble to hit in the next state, absolute amount for the delta to ID vs. a cut-off under zero, and for IDD a substitution with a known obstacle vs. an empty space. The approaches are compared in table 1 where the hyper-parameters are tuned for equal FP probabilities to achieve directly comparable FN probabilities, and as a ROC (Receiver Operating Characteristic) curve in figure 8.

	w/o UE	UE-BL	UE-IDD
mean P(FN) (false-negative)	41.61e-3	3.86e-3	2.45e-3
mean P(FP) (false-positive)	-	11.49e-2	11.22e-2

Table 1
Impact of the uncertainty estimators on the overall false-negative and false-positive probabilities

IDD significantly reduces FNs compared to the agent with the baseline uncertainty estimator (about factor 1.75) and the agent without UE (about factor 20). This comes with the cost of an increasing FP rate for the UE agents, whereas the agent without UE naturally has no FPs.

When mapping the experimental results to the safety assurance argumentation from Section 4.2 and the iterative causal analysis model, it becomes clear that the safety

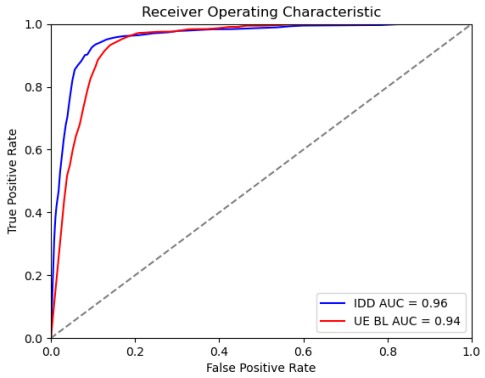


Figure 8: ROC curve for UE-BL (red) and IDD (blue)

claim may not be met without an uncertainty estimator and would then be improved within the 1st iteration when introducing the UE-BL. The results show a significant improvement for the FN, but assuming an even higher safety claim of e.g. FN less then 3‰, a 2nd iteration identified additional measures to further reduce the FN. The 2nd iteration with IDD as an additional measure then reached the required claim. When then looking at the high remaining FP and a potential additional claim in respect to that, a 3rd iteration could address this aspect. However, this will be the target of future work.

Finally, whether all possible scenarios have been considered and whether the safety assurance achieved is sufficient as rigorous evidence for certification purposes needs further investigation on more realistic applications in future work.

6. Conclusion

This paper investigated the safety assurance argumentation for an ensemble based epistemic uncertainty estimation on gridworld scenarios with discrete action spaces and overlapping alternative predictions.

We build on previous work with discrete actions spaces and variance calculation based on action count variance (ACV) and a delta to ID (IDD) approach to deal with overlapping alternative predictions, where we showed that action count variance with IDD is able to indicate uncertain states based on a threshold calculation with high probability. As utilizing a backup policy based on that indication can be a feasible solution, we established a safety assurance argumentation in this paper. With the definition of the assurance case and an iterative assurance approach, we demonstrated that the IDD-enhanced uncertainty estimator can be utilized as an operation-time measure as external observer to indicate ontological uncertainty.

Future work will address to reduce the FP rate of the observer, investigate methods to determine a sufficient near ID scenario for a given OOD scenario and extend the approach to more general and realistic environments and applications. Further, it will focus on rigorous argumentation and elaboration of the experimental results for the safety assurance and on strategies to react upon the uncertainty estimation during operation to reduce situational risk.

Acknowledgments

This work was funded by the Bavarian Ministry for Economic Affairs, Regional Development and Energy as part of a project to support the thematic development of the Institute for Cognitive Systems.

References

[1] S. Burton, A causal model of safety assurance for machine learning, arXiv:2201.05451 (2022).

[2] D. Hendrycks, K. Gimpel, A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks, arXiv:1610.02136 [cs] (2018).

[3] B. Lütjens, M. Everett, J. P. How, Safe reinforcement learning with model uncertainty estimates, in: 2019 International Conference on Robotics and Automation (ICRA), IEEE, 2019, pp. 8662–8668.

[4] K. Lee, K. Lee, H. Lee, J. Shin, A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks, arXiv:1807.03888 [cs, stat] (2018).

[5] J. Postels, H. Blum, Y. Strümpler, C. Cadena, R. Siegwart, L. Van Gool, F. Tombari, The Hidden Uncertainty in a Neural Networks Activations, arXiv:2012.03082 (2020).

[6] M. Pimentel, D. Clifton, L. Clifton, L. Tarassenko, A review of novelty detection, Signal Process. (2014). doi:10.1016/j.sigpro.2013.12.026.

[7] T. DeVries, G. W. Taylor, Learning confidence for out-of-distribution detection in neural networks, arXiv preprint arXiv:1802.04865 (2018).

[8] S. Mohseni, M. Pitale, J. Yadawa, Z. Wang, Self-supervised learning for generalizable out-of-distribution detection, Proceedings of the AAAI Conference on Artificial Intelligence 34 (2020). doi:10.1609/AAAI.V34I04.5966.

[9] A. Schwaiger, P. Sinhamahapatra, J. Gansloser, K. Roscher, Is Uncertainty Quantification in Deep Learning Sufficient for Out-of-Distribution Detection?, in: Proc. AISafety@IJCAI2020, volume 2640 of *CEUR Workshop Proceedings*, 2020, p. 8.

[10] Q. Yu, K. Aizawa, Unsupervised Out-of-Distribution Detection by Maximum Classifier Discrepancy, arXiv:1908.04951 [cs] (2019).

[11] A. Sedlmeier, T. Gabor, T. Phan, L. Belzner, C. Linnhoff-Popien, Uncertainty-based out-of-distribution classification in deep reinforcement learning, arXiv preprint arXiv:2001.00496 (2019).

[12] W. R. Clements, B. Van Delft, B.-M. Robaglia, R. B. Slaoui, S. Toth, Estimating Risk and Uncertainty in Deep Reinforcement Learning, arXiv:1905.09638 [cs, stat] (2020).

[13] K. Chua, R. Calandra, R. McAllister, S. Levine, Deep Reinforcement Learning in a Handful of Trials using Probabilistic Dynamics Models, arXiv:1805.12114 (2018).

[14] C.-J. Hoel, K. Wolff, L. Laine, Ensemble quantile networks: Uncertainty-aware reinforcement learning with applications in autonomous driving, arXiv:2105.10266 (2021).

[15] W. Dabney, G. Ostrovski, D. Silver, R. Munos, Implicit quantile networks for distributional reinforcement learning, arXiv:1806.06923 (2018).

[16] I. Osband, J. Aslanides, A. Cassirer, Randomized Prior Functions for Deep Reinforcement Learning, arXiv:1806.03335 (2018).

[17] P. Wang, Y. Li, S. Shekhar, W. F. Northrop, Uncertainty Estimation with Distributional Reinforcement Learning for Applications in Intelligent Transportation Systems: A Case Study, in: 2019 IEEE Intelligent Transportation Systems Conference (ITSC), 2019, pp. 3822–3827. doi:10.1109/ITSC.2019.8917429.

[18] G. Kahn, A. Villaflor, V. Pong, P. Abbeel, S. Levine, Uncertainty-Aware Reinforcement Learning for Collision Avoidance, arXiv:1702.01182 (2017).

[19] F. L. Da Silva, P. Hernandez-Leal, B. Kartal, M. E. Taylor, Uncertainty-aware action advising for deep reinforcement learning agents, in: Proceedings of the AAAI conference on artificial intelligence, volume 34, 2020, pp. 5792–5799.

[20] C.-J. Hoel, K. Wolff, L. Laine, Tactical Decision-Making in Autonomous Driving by Reinforcement Learning with Uncertainty Estimation, arXiv:2004.10439 (2020).

[21] S. Burton, C. Hellert, F. Hüger, M. Mock, A. Rohatschek, Safety assurance of machine learning for perception functions, in: Deep Neural Networks and Data for Automated Driving, Springer, Cham, 2022, pp. 335–358.

[22] P. J. Graydon, Defining baconian probability for use in assurance argumentation, NASA/TM–2016–219341 (2016).

[23] R. Hawkins, T. Kelly, J. Knight, P. Graydon, A new approach to creating clear safety arguments, Advances in systems safety, pp. 3-23. Springer, London (2011).

[24] J. Goodenough, C. Weinstock, A. Klein, Toward a Theory of Assurance Case Confidence, Technical Report CMU/SEI-2012-TR-002, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, 2012. URL: http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=28067.

[25] D. Eilers, F. S. Roza, K. Roscher, Ensemble-based uncertainty estimation with overlapping alternative predictions, Deep RL Workshop at the 36th Conference on Neural Information Processing Systems (NeurIPS) (2022).

[26] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, M. Riedmiller, Playing atari with deep reinforcement learning, arXiv:1312.5602 (2013).

[27] T. Haider, F. S. Roza, D. Eilers, K. Roscher, S. Günnemann, Domain shifts in reinforcement learning: Identifying disturbances in environments., AISafety@ IJCAI (2021).

[28] W. E. Walker, P. Harremoës, J. Rotmans, J. P. Van Der Sluijs, M. B. Van Asselt, P. Janssen, M. P. Krayer von Krauss, Defining uncertainty: a conceptual basis for uncertainty management in model-based decision support, Integrated assessment 4 (2003) 5–17.

[29] R. Gansch, A. Adee, System theoretic view on uncertainties, in: 2020 Design, Automation & Test in Europe Conference & Exhibition (DATE), IEEE, 2020, pp. 1345–1350.

[30] S. Burton, I. Kurzidem, A. Schwaiger, P. Schleiss, M. Unterreiner, T. Graeber, P. Becker, Safety assurance of machine learning for chassis control functions, in: International Conference on Computer Safety, Reliability, and Security, Springer, 2021, pp. 149–162.

[31] Goal structuring notation community standard version 2, Technical Report, Assurance Case Working Group (ACWG), https://scsc.uk/r141B:1?t=1, accessed on 04/05/2019, 2018.

[32] J. Spriggs, GSN - The Goal Structuring Notation: A Structured Approach to Presenting, 2012.