Benchmark for Out-of-Distribution Detection in Deep Reinforcement Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

Reinforcement Learning (RL) based solutions are being adopted in a variety of domains including robotics, health care and industrial automation. Most focus is given to when these solutions work well, but they fail when presented with out of distribution inputs. RL policies share the same faults as most machine learning models. Out of distribution detection for RL is generally not well covered in the literature, and there is a lack of benchmarks for this task. In this work we propose a benchmark to evaluate OOD detection methods in a Reinforcement Learning setting, by modifying the physical parameters of non-visual standard environments or corrupting the state observation for visual environments. We discuss ways to generate custom RL environments that can produce OOD data, and evaluate three uncertainty methods for the OOD detection task. Our results show that ensemble methods have the best OOD detection performance with a lower standard deviation across multiple environments.

1 Introduction

Reinforcement learning (RL) is one of the paradigms of machine learning. It involves training an agent to solve tasks by interacting with the environment and learning from its experience. In the recent years, RL has been successful in a variety of domains including robotics [12], game playing [27] and even agricultural applications [6]. Researchers are using it on a daily basis to make important decisions. The performance of trained agents is highly dependent on the experience or data seen during training. It is usually assumed that the test data follows the same distribution as the training data. However, this assumption does not hold true in many real world applications. The samples or observations that do not conform to the underlying distribution of the training data are referred to as out-of-distribution (OOD) samples.

Recent RL algorithms are making use of deep neural networks which are known to be sensitive to OOD data [9]. This can result in incorrect decisions which in turn can have significant costs. When developing new algorithms, researchers usually focus on the performance of the models calculated by metrics like accuracy and mean squared error and often ignore to report the models' uncertainty in its predictions. The uncertainty of the model can be directly associated with the trust in its predictions. Predictions with higher uncertainty can be rejected or can be an indication for the need of human processing instead of automation.

This work focuses on detecting OOD samples in the context of reinforcement learning policies. Several approaches have come up in the recent years to identify OOD data. But, most of them focus on detecting OOD samples for image classification problems. This is due to the fact that the other

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

domains suffer from the availability of OOD or adversarial examples. This work aims to extend the detection of OOD data to deep reinforcement learning and provide a benchmark for future researchers to test their methods.

Our contributions are a benchmark for OOD detection in reinforcement learning, by creating OOD environments that have corrupted observed states or modified physical parameters, which enable the evaluation of OOD detection methods in reinforcement learning. We provide initial results using Dropout, DropConnect, and ensembles, finding that ensembles work best for this task.

2 Related Work

To the best of our knowledge, there is currently no benchmark available for evaluating out-of-distribution detection in reinforcement learning. However, there has been a surge in interest in this field in the recent years. In this section, we discuss the prior work that has been done for out-of-distribution detection in general. We also discuss the research that has been done for out-of-distribution detection in a reinforcement learning setting.

Various available methods for out-of-distribution or anomaly detection can be categorized based on the availability of anomaly labels into supervised, unsupervised and semi-supervised techniques [4] . The supervised methods used uncertainty measures based on the gradient of the negative log-likelihood [23], Mahalanobis distance from different layers [17], Long short-term memory (LSTM) based binary detectors[5]. Some of best semi-supervised methods used Likelihood ratio [25], Probably Approximately Correct (PAC) based algorithm [19] and a two-head Convolutional Neural Network (CNN) [29] for anomaly detection.

Unsupervised techniques include using predicted softmax probability [10], Temperature scaling [18], and Generative Adversarial Network (GAN) based architecture [16]. Overall, supervised methods tend to perform better than the other methods as ground-truths are available. However, having the examples and labels for a full spectrum of anomalies may not be possible in all the cases and this might result in overfitting. Unsupervised methods are flexible and can be applied to a variety of domains as they don't rely on the labels and anomalous data. However, they are highly sensitive to noise. Semi-supervised methods have the flexibility of unlabelled data along with the accuracy from the labelled data. However, they tend to overfit in unseen anomalous situations. One of the challenges in anomaly detection in deep learning is to define the boundary between normal and anomalous examples with complex feature spaces.

Uncertainty estimation provides good results for Independent and Identically Distributed (IID) samples. However, most of these methods tend to fail when there is even a mild change in the dataset distribution. [24] focuses on understanding the quality of uncertainty estimates in the case of distributional shift along with IID setting. A set of probabilistic deep learning methods like Maximum softmax probability, Monte-Carlo Dropout [8], Ensembles [14], Temperature Scaling, Stochastic Variational Bayesian Inference (SVI) [28] were evaluated on images, text and MNIST data. In addition to the classification accuracy, metrics like Brier score [2], Negative Log-Likelihood and Expected Calibration Error (ECE) are calculated. For MNIST, the accuracy of all the models degrade as the shift in the data increases. The Brier score differentiates the evaluated methods more clearly. All methods have a better Brier score than the state-of-the-art temperature scaling method. Even though SVI achieves the worst accuracy, it outperforms all the other methods when the data shift is significant. Most of the methods show high confidence in their predictions on entirely OOD data. CIFAR-10 [13] and ImageNet [7] datasets were used to study the predictive uncertainty on image data. Ensembles had the best performance across most of the metrics. The performance of all the methods follows the same order in both the image datasets. However, the order is not the same as SVI performs worse than vanilla method on the shifted datasets. The 20newsgroups [15] dataset is used to evaluate the predictive uncertainty on text data. Similar to the performance on image data, ensembles outperform all the other methods in terms of accuracy and uncertainty estimation. The uncertainty does not change significantly with temperature scaling even for significantly shifted data. On fully OOD data, vanilla method had better performance than dropout and SVI methods. Overall,

ensembles outperformed all the other methods in all the tasks with a better trade-off between accuracy and confidence.

While [4] and [24] provide a large-scale comparison of the OOD methods and an extensive benchmark for evaluating uncertainty estimates respectively, all the discussed methods were evaluated on image or text data. Recently, [26] presented an OOD detection method applicable for reinforcement learning problems. The solution involves modeling the OOD detection problem as a classification problem with two classes i.e. one for in-distribution data and the other for OOD data. The authors propose a framework called UBOOD [26] for uncertainty-based OOD detection. It is based on the principle that the epistemic uncertainty is lower for in-distribution (observed during training) samples than for the OOD samples. Two environments were developed for evaluating the proposed framework. One of the environments is a simple grid-based world with a discrete state-space and the other has a continuous state-space based on OpenAI's Lunar lander. Three different versions of the proposed framework have been evaluated based on their F1 scores, with each version based on Monte-Carlo Concrete Dropout network, Bootstrap network, Bootstrap-prior network respectively. The UBOOD framework with Bootstrap-prior network performs the best in detecting the OOD samples on both the environments. It was observed that the F1 score is better for the environment which differs the most from the training environment. Similarly, [20] proposes a risk sensitive reinforcement learning approach that can be combined with a RL policy to make it sensitive to novel data. This work specifically focusses on dynamic obstacle avoidance problem in novel scenarios. The agent can simultaneously observe its goal and the position and velocity of the obstacle. The probabilities of collision for each motion are calculated by LSTM networks. A distribution of the predictions are calculated by MC-Dropout and Bootstrapping. Predictions are used to calculate the mean and variance for each motion primitive. The time to reach the goal after every motion primitive is also estimated in parallel using a simple model. At each time step, the motion primitive with the least collision probability is selected and the process is repeated. This model is evaluated against an uncertainty unaware model. The results show that the uncertainty aware model is more robust to novel obstacles. However, the uncertainty values in novel scenarios did not increase significantly.

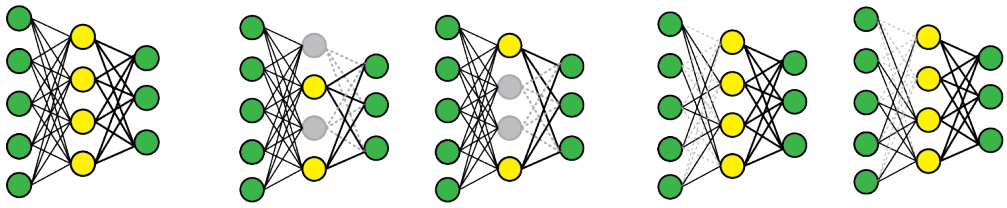
While [26] and [20] explore the implementation of out-of-distribution detection in reinforcement learning tasks, the authors are forced to create their own environments for experiments. This is due to the lack of available benchmark for out-of-distribution or anomaly detection in reinforcement learning. This highlights the need and importance of benchmark tasks for pushing the research on OOD detection in RL even further.

3 Out-of-Distribution Detection and Uncertainty

Deep learning is being used to solve complex problems across a variety of domains including autonomous vehicles, industrial automation, health care and surveillance. It has shown to perform as good as or even better than humans in some of these tasks. However, most of the learning methods assume the test data to be from the same distribution as of the training data. This assumption does not hold true in many real world applications. The samples that deviate from the underlying distribution of the training data are referred to as out-of-distribution (OOD) samples or anomalies. Deep learning methods in general are known to be sensitive to OOD data and lead to incorrect results.

An example specific to reinforcement learning is the autonomous control of industrial robots. These robots are typically deployed in a human-robot collaborative environment. In the absence of an OOD detection mechanism, any new work setting which has not been seen during training can make the robot to take actions that could be fatal to the human and other resources that are in its vicinity. This makes out-of-distribution detection extremely important for the safety of the humans and the environment in which the models are deployed.

Out-of-distribution detection corresponds to the task of identifying samples or observations where the model is uncertain about its output. Different methods used to estimate uncertainty in this work are Monte Carlo Dropout [8], Monte Carlo DropConnect [22] and Ensembles [14]. Dropout is a regularization technique in neural networks. During training, some of the activations are randomly



(a) Standard network (b) Network after applying dropout (c) Network after applying dropconnect

Figure 1: Figure showing the effect of using dropout layers in the network [11]

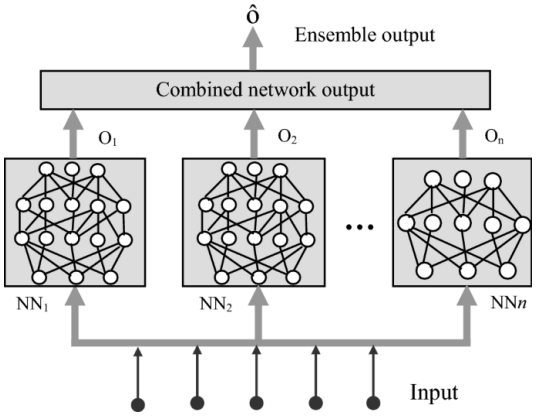


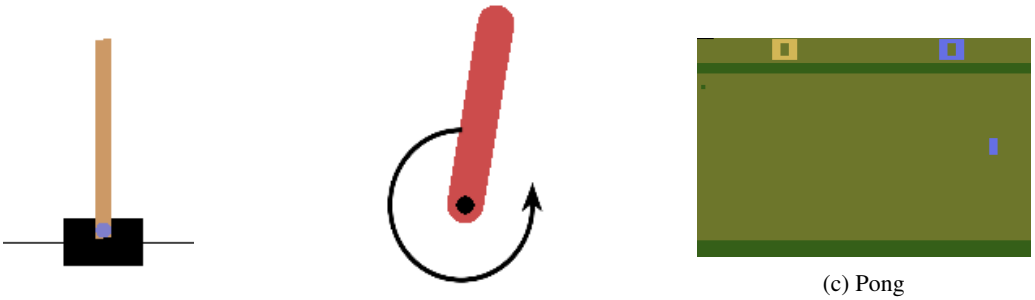
Figure 2: Training an Ensemble network [1]

dropped out. This has proven to be a simple yet effective method to avoid overfitting. Monte Carlo Dropout is the method of enabling dropout at inference which has proven to be an approximation of the predictive posterior distribution. If the same input is applied to the network multiple times, an empirical distribution can be estimated and the parameters like mean and variance can be obtained. This variance serves as a measure of the model's uncertainty. The variance is expected to be low in the input areas where there was enough training data and can be high where there was no or little training data.

DropConnect [22] is a variation of Dropout where the weights are dropped out instead of the activations of a layer. It has also proven to produce an approximation of the predictive posterior distribution. While implementing dropout is simple, implementation of DropConnect requires new layers that use DropConnect layers internally. MC DropConnect is sometimes seen to perform better than MC Dropout in both learning the task and the uncertainty quantification. Ensemble corresponds to training multiple instances of the same model but randomly drawn initial weights and then combining the predictions. They have better prediction performance than any single member model. They have also shown to exhibit excellent uncertainty estimation properties.For classification tasks, the entropy can be used as a measure of uncertainty. For regression tasks, the standard deviation of the output can be used.

4 Experimental Setup

In this section, various tasks that were used to evaluate the performance of out-of-distribution detection methods are described. As real world applications of reinforcement learning are in both visual and non-visual based environments, a combination of tasks that cover these aspects are chosen. These tasks include cartpole, pendulum and pong. Cartpole is a non-visual physics based environment and has a discrete action space. Pendulum is also a non-visual physics based environment but with a continuous action space. Pong is a visual based environment with a discrete action space. These



(a) Cartpole

(b) Pendulum

(c) Pong

Figure 3: Figures showing the different environments/tasks used.

Task	Algorithm	OOD detection method
Cartpole	DQN	MC Dropout, MC DropConnect, Ensemble
Cartpole	PPO	MC Dropout, MC DropConnect, Ensemble
Pendulum	DDPG	MC Dropout, MC DropConnect, Ensemble
Pong	DQN	MC DropConnect, Ensemble

Table 1: Combinations of tasks, RL algorithms and OOD detection methods evaluated.

environments have been taken from OpenAI Gym suite [3] which is a framework for the development and comparison of reinforcement learning algorithms.

To generate custom versions of the cartpole environment, physical variables like gravity, mass of the cart, mass of the pole, length of the pole and the magnitude of the force to be applied to the cart are assigned new values. A grid of values are chosen in multiples of the default values for each parameter. For example, the force magnitude has a default value of 10. The custom versions have values ranging from 1.0 to 100.0 as a multiple (x/10, x/9, x/8,..., x/2, 2x, 3x,...,10x) of the default value. Custom versions of the pendulum environment are generated in the same way as the cartpole environment by assigning new values of physical variables like gravity, mass of the pole, length of the pole along with the speed and the torque to be applied. Unlike the physical environments like cartpole and pendulum, custom versions of the pong environment are generated by corrupting the observations (images). For this, we have used the imagecorruptions [21] package, that supports various corruption types including gaussian noise, impulse noise, motion blur, and zoom blur. We can also adjust the severity of the corruption using this package.

The performance of the different out-of-distribution detection methods is evaluated in the following way.

1. A task is trained using a deep RL algorithm.
2. The trained model is evaluated on multiple custom versions of the environment.
3. The custom environment versions where the trained model fails are identified.
4. Different out-of-distribution detection methods are evaluated on these custom environments.
5. The above steps are repeated for different tasks.

5 Experimental Results and Analysis

In this section, we analyze the OOD detection performance of various methods on cartpole, pendulum and the pong environments.

OOD detection method	Custom configuration	Best AUC score
MC Dropout	Force: 2.5	**0.780**
MC Dropout	Gravity: 49	0.759
MC Dropout	Mass of the cart: 3	0.719
MC DropConnect	Force: 1	0.921
MC DropConnect	Gravity: 78.4	**0.992**
MC DropConnect	Length of the pole: 2	0.987
MC DropConnect	Mass of the cart: 9	0.887
Ensemble	Gravity: 98	0.833
Ensemble	Length of the pole: 2	**0.993**

Table 2: Best AUC scores achieved by MC Dropout, MC DropConnect and ensemble method on different custom versions of the cartpole environment. The overall best AUC score of each OOD method is highlighted.

Table 2 lists the different OOD methods along with their best AUC scores on the specific custom versions of the cartpole environment. MC Dropout achieves its best AUC score of 0.78 on the custom version having the force parameter of 2.5. On the other hand, MC DropConnect achieves its best AUC score of 0.992 on the custom version having the gravity of 78.4. However, the best overall AUC score of 0.993 on the custom cartpole environments is achieved by the ensemble method on the version with the length of the pole as 2. Apart from the best AUC scores, the lowest standard deviation of the AUC scores across trials is also achieved by the ensemble method followed by the MC Dropout and then the MC DropConnect which has the largest standard deviation values for the custom cartpole environments. This shows that the Ensemble method is the best performing OOD detection method for the cartpole environment.

Figure 4: Comparison of Ensemble mean and standard deviation of the Q values produced by DQN along with the threshold for OOD detection for original cartpole environment and a custom environment with length of pole of 2

Figure 4 show the mean and standard deviation of the Q values during one episode for both the original environment and the custom version with a length of 2. As seen from the figure, the standard deviation is almost always less than the OOD threshold for the original environment over the course of the entire episode. However, for the custom environment, the standard deviation is more than the OOD threshold right from the initial steps of the episode. This behavior is seen for both the actions. This also highlights the good OOD detection performance of the ensemble method.

Table 3 lists the different OOD methods along with their best AUC scores on the specific custom versions of the pendulum environment. MC Dropout achieves its best AUC score of 0.727 on the custom version having the gravity of 50. On the other hand, MC DropConnect achieves its best AUC

OOD detection method	Custom configuration	Best AUC score
MC Dropout	Gravity: 50	**0.727**
MC Dropout	Length of the pole: 5	0.656
MC Dropout	Mass of the cart: 5	0.726
MC DropConnect	Gravity: 50	0.602
MC DropConnect	Length of the pole: 5	**0.726**
MC DropConnect	Mass of the cart: 5	0.715
Ensemble	Length of the pole: 0.1	**0.619**
Ensemble	Mass of the cart: 0.05	0.596

Table 3: Best AUC scores achieved by MC Dropout, MC DropConnect and ensemble method on different custom versions of the pendulum environment. The overall best AUC score of each OOD method is highlighted.

score of 0.726 on the custom version having the length of the pole of 5. Similarly, the ensemble method achieves its best overall AUC score of 0.619 on the custom pendulum environment having the length of the pole as 0.1. Apart from the best AUC scores, the lowest standard deviation of the AUC scores across trials is achieved by the ensemble method followed by the MC Dropout and then the MC DropConnect which has the largest standard deviation values for the custom pendulum environments. This shows that there is a trade off between the best possible performance and consistency for the OOD detection methods. Overall, the MC Dropout method can be considered the best performing OOD detection method for the pendulum environment with an acceptable level of reproducibility in performance.

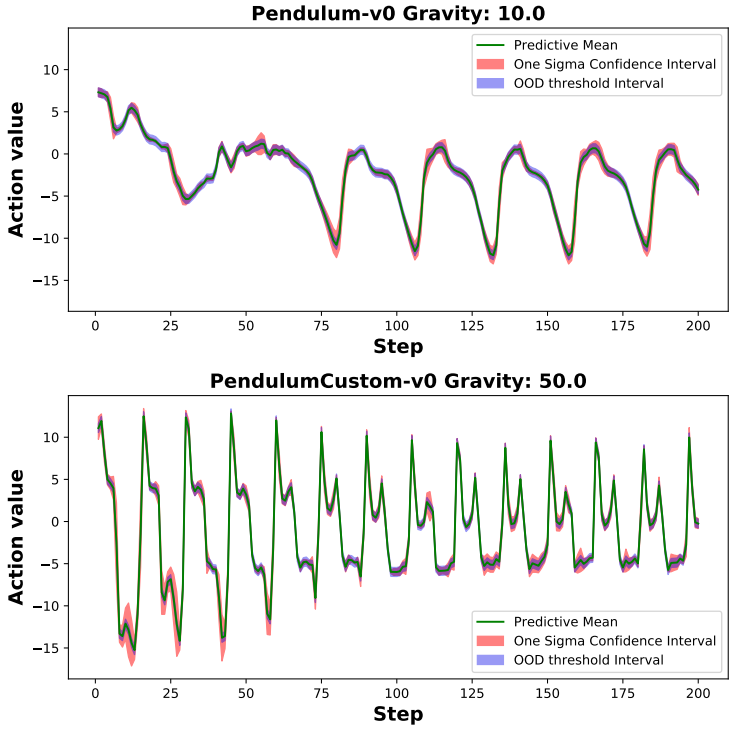


Figure 5: Comparison of action values obtained by MC Dropout on DDPG for original pendulum environment with gravity of 10 and the custom version with gravity of 50. The plot shows the mean of the action value obtained using the five predictors along with the standard deviation and the best threshold interval computed to distinguish the ID and OOD observations.

Custom configuration	Best AUC score	
	MC DropConnect	Ensemble
Gaussian noise: 3	0.699	0.647
Gaussian noise: 4	0.749	0.735
Gaussian noise: 5	**0.828**	0.831
Impulse noise: 3	0.712	0.656
Impulse noise: 4	0.775	0.769
Impulse noise: 5	0.822	0.836
Motion blur: 3	0.707	**0.910**
Motion blur: 4	0.682	0.861
Motion blur: 5	0.685	0.841
Pixelate: 3	0.606	0.823
Pixelate: 4	0.565	0.634

Table 4: Best AUC scores achieved by MC DropConnect and Ensemble methods on different custom versions of the pong environment.

Figure 5 shows the progress of the action values along with their standard deviations for a duration of one episode obtained during the best performing trail of MC Dropout model for the original pendulum environment with gravity of 10 and the custom version with gravity of 50. It can be seen that the standard deviation values obtained for the original pendulum environment are higher than the OOD threshold value especially in the middle of the episode. Similarly, the standard deviation values obtained for the custom environment are higher than the OOD threshold value in the beginning of the episode and also occasionally in the later part of the episode. This shows that MC Dropout is not very efficient in distinguishing between ID and OOD observations for the pendulum environment.

Table 4 lists the different OOD methods along with their best AUC scores on the specific custom versions of the pong environment. MC DropConnect achieves its best AUC score of 0.828 on the custom pong environment corrupted with Gaussian noise of severity level 5. On the other hand, the ensemble method achieves its best AUC score of 0.91 on the custom version corrupted using motion blur with a severity of 3. When the performance of both the methods is compared based on the corruption type, the ensemble method achieves better AUC scores than MC DropConnect across all corruptions. The ensemble method also has lower standard deviations across trials than MC DropConnect. This shows that the ensemble method has the best OOD detection performance for the pong environment.

6 Conclusions and Future Work

In this work, the OOD detection performance of different uncertainty estimation methods i.e. MC Dropout, MC DropConnect and ensemble is compared across a range of control tasks. The tasks included two physics based environments i.e. cartpole, which has a discrete action space and pendulum, which has a continuous action space. The OOD detection methods were also evaluated on pong, which is a visual based environment. The difference in the performance of the trained models between the original environment and the custom versions highlight the sensitivity of the models to the changes in the environment. The models trained on visual based environment were, in general, more sensitive to changes in the environment than the models trained on physics based environments. One of the major challenges faced during training the dropout and dropconnect models was to identify the appropriate level of dropout probability that the models can still learn to solve the original versions of the tasks.

Ensemble methods achieved the best OOD detection performance on cartpole and pong environments while MC Dropout performed the best on the pendulum environment. The ensemble method also had the lowest variation in its performance over multiple trials across all the environments. The MC

DropConnect has a good OOD detection performance across all the environments, however, it is not consistent. This work also highlights the effect of the RL algorithm on the performance of the OOD detection methods. The overall AUC scores obtained using DQN based models are higher than the ones obtained by DDPG. Nonetheless, more experiments are needed to confirm this behavior.

Overall, the experiments show that MC Dropout, MC DropConnect and the ensemble method were successful in detecting OOD observations in deep reinforcement learning. This is especially true in the case of the visual based environments, where the trained models failed on almost all the custom versions of the environment but were able to detect OOD observations to a large extent especially with higher levels of corruption. Future work can be done in developing methods that not only estimate the uncertainty but also learn from the OOD observations to create more robust models. Researchers are also encouraged to test the OOD detection methods on more complex environments.

References

[1] K.M.R. Alam, N. Siddique, and H Adeli. A dynamic ensemble learning algorithm for neural networks. *Neural Comput and Applic 32, 8675–8690*, 2020.

[2] G. W. Brier. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78:1–3, 1950.

[3] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016. Last accessed: 13.07.2021.

[4] S. Bulusu, B. Kailkhura, B. Li, P. K. Varshney, and D. Song. Anomalous example detection in deep learning: A survey. *IEEE Access*, 8:132330–132347, 2020.

[5] Fabio Carrara, Rudy Becarelli, Roberto Caldelli, Fabrizio Falchi, and Giuseppe Amato. Adversarial examples detection in features distance spaces. In Laura Leal-Taixé and Stefan Roth, editors, *Computer Vision – ECCV 2018 Workshops*, pages 313–327, Cham, 2019. Springer International Publishing.

[6] Mengting Chen, Yuanlai Cui, Xiaonan Wang, Hengwang Xie, Fangping Liu, Tongyuan Luo, Shizong Zheng, and Yufeng Luo. A reinforcement learning approach to irrigation decision-making for rice using weather forecasts. *Agricultural Water Management*, 250:106838, 2021.

[7] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, June 2009.

[8] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.

[9] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. *arXiv preprint arXiv:2006.16241*, 2020.

[10] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *CoRR*, abs/1610.02136, 2016.

[11] Matt Krause (https://stats.stackexchange.com/users/7250/matt krause). What is the difference between dropout and drop connect? Cross Validated. URL:https://stats.stackexchange.com/q/201891 (version: 2016-03-15).

[12] Jemin Hwangbo, Joonho Lee, Alexey Dosovitskiy, Dario Bellicoso, Vassilios Tsounis, Vladlen Koltun, and Marco Hutter. Learning agile and dynamic motor skills for legged robots. *Science Robotics*, 4(26), 2019.

[13] Alex Krizhevsky. Learning multiple layers of features from tiny images. *University of Toronto*, 05 2012.

[14] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *arXiv preprint arXiv:1612.01474*, 2016.

[15] Ken Lang. Newsweeder: Learning to filter netnews. In Armand Prieditis and Stuart Russell, editors, *Machine Learning Proceedings 1995*, pages 331 – 339. Morgan Kaufmann, San Francisco (CA), 1995.

[16] W. Lawson, E. Bekele, and K. Sullivan. Finding anomalies with generative adversarial networks for a patrolbot. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 484–485, 2017.

[17] Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples, 2018.

[18] Shiyu Liang, Yixuan Li, and R. Srikant. Principled detection of out-of-distribution examples in neural networks. *CoRR*, abs/1706.02690, 2017.

[19] Si Liu, Risheek Garrepalli, Thomas G. Dietterich, Alan Fern, and Dan Hendrycks. Open category detection with PAC guarantees. *CoRR*, abs/1808.00529, 2018.

[20] 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, May 2019.

[21] Claudio Michaelis, Benjamin Mitzkus, Robert Geirhos, Evgenia Rusak, Oliver Bringmann, Alexander S. Ecker, Matthias Bethge, and Wieland Brendel. Benchmarking robustness in object detection: Autonomous driving when winter is coming. *arXiv preprint arXiv:1907.07484*, 2019.

[22] Aryan Mobiny, Pengyu Yuan, Supratik K Moulik, Naveen Garg, Carol C Wu, and Hien Van Nguyen. Dropconnect is effective in modeling uncertainty of bayesian deep networks. *Scientific reports*, 11(1):1–14, 2021.

[23] Philipp Oberdiek, Matthias Rottmann, and Hanno Gottschalk. Classification uncertainty of deep neural networks based on gradient information. *CoRR*, abs/1805.08440, 2018.

[24] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, D Sculley, Sebastian Nowozin, Joshua V. Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift, 2019.

[25] Jie Ren, Peter J. Liu, Emily Fertig, Jasper Snoek, Ryan Poplin, Mark A. DePristo, Joshua V. Dillon, and Balaji Lakshminarayanan. Likelihood ratios for out-of-distribution detection, 2019.

[26] Andreas Sedlmeier, Thomas Gabor, Thomy Phan, Lenz Belzner, and Claudia Linnhoff-Popien. Uncertainty-based out-of-distribution detection in deep reinforcement learning. *CoRR*, abs/1901.02219, 2019.

[27] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.

[28] Yeming Wen, Paul Vicol, Jimmy Ba, Dustin Tran, and Roger Grosse. Flipout: Efficient pseudo-independent weight perturbations on mini-batches. *arXiv preprint arXiv:1803.04386*, 2018.

[29] Qing Yu and Kiyoharu Aizawa. Unsupervised out-of-distribution detection by maximum classifier discrepancy, 2019.