# Uncertainty Estimation Using a Single Deep Deterministic Neural Network - ML Reproducibility Challenge 2020

**Anonymous Author(s)**
Affiliation
Address
email

## Reproducibility Summary

### Scope of Reproducibility

Investigated paper claims RBF network when trained with BCE loss along with two sided gradient penalty outperforms deep ensemble in the task of out of distribution(OoD) detection along with competitive accuracy to softmax based models. Paper claims to outperform AUROC on Fashion-MNIST vs MNIST and CIFAR-10 vs SVHN. Proposed algorithm is reported to detect both aleatoric and epistemic uncertainty as OoD. Authors mention the need of a formal way to distinguish between the two kinds of uncertainty and pose it as an interesting future research avenue.
Scope of this report is to reproduce the results related to OoD detection presented in paper. Along with reproduction of results, we propose an extension for explicit detection of aleatoric and epistemic uncertainty as intended by the authors.

### Methodology

Author's training code is available on GitHub. Additionally, we have made available all the experimentation codes in the form of notebooks. We provide all the results and analysis on the models described in the paper and trained on NVIDIA Tesla T4 GPU.

### Results

Our reproduction supports the claim of paper, we are able to replicate exact trends as described in the paper, all results are within 1% of the value reported. Also, results of our proposed extension and its analysis are quite encouraging.

### What was easy

The proposed approach is quite simple and elegant. Availability of author's code made implementation of various experiments easy.

### What was difficult

Understanding of proposed approach require advanced knowledge of calculus related to Lipschitz constant and its role to quantify upper bound of sensitivity of any function.

### Communication with original authors

We discussed our report with authors, they find our analysis on aleatoric uncertainty interesting and appreciate our proposed extension and its encouraging results.

Submitted to ML Reproducibility Challenge 2020. Do not distribute.

# 1 Introduction

Reliable uncertainty quantification has been a challenge in deep learning, models giving overconfident wrong predictions can not be deployed for practical purposes and if deployed can even be fatal. The investigated paper Uncertainty Estimation Using a Single Deep Deterministic Neural Network [1] presents a method of training deep neural networks to detect out of distribution(OoD) points in a single forward pass along with classification.

Authors define a set of feature vectors called centroids each corresponding to certain class. Distance between feature vectors predicted by model and centroids are used for class prediction. This architecture along with BCE loss employs two-sided-gradient penalty to make model sensitive to input and hence providing OoD detection ability, this approach is called deterministic uncertainty quantification, DUQ.

# 2 Scope of reproducibility

The proposed algorithm outperforms state of the art techniques for OoD detection on Fashion-MNIST vs MNIST , CIFAR-10 vs SVHN in terms of AUROC as well as computational time. DUQ, a RBF network based algorithm provides competitive accuracy to softmax models along with strong OoD detection ability. Scope of this report is as follows.

## 2.1 Addressed claims from the original paper

- OoD detection ability on various datasets as reported in paper.
- Role of different hyperparameters (Ablation Study).

## 2.2 Other experiments

- Noise sensitivity of proposed algorithm.
- Propose extension E-DUQ for explicit detection of aleatoric and epistemic uncertainty .

# 3 Methodology

## 3.1 Model descriptions

Figure 1 represents the complete algorithm, first the features are computed through standard feature extractor ($f_\theta$) after this extracted features are passed through class specific layers ($W_c$) to calculate feature vector for each class. Distance ($K_c$) of this vector from the centroid in a kernel space represents uncertainty in prediction. Centroid is calculated for each batch as described in equations below .

$n_{c,t}$ = $\gamma \times \mathrm{n}_{c,t-1}$ + (1-$\gamma$)$\times \mathrm{n}_{c,t}$

$m_{c,t}$ = $\gamma \times \mathrm{m}_{c,t-1}$ + (1-$\gamma$)$\times \sum \mathrm{W}_c f_\theta(x_{c,t,i})$

$e_{c,t}$ = $m_{c,t}$ / $n_{c,t}$

Where $n_{c,t}$ is number of images of class $c$ in a particular batch at time $t$ and $\gamma$ is the momentum for updating centroids.

Total cost comprise of two loss functions, one is the BCE loss which is the sum of cross entropy of a binary one-hot vector with actual label of that class as described in equation below.

$L_1(x, y)$= $\sum_c \mathrm{y}_c$ log($K_c$) + (1 -$y_c$) log(1-$K_c$)

Second part is the gradient penalty to enforce sensitivity in the model, a two-sided gradient penalty loss is defined as:

Two-sided GP loss = $\lambda \cdot \left[ \|\nabla_x \sum_c \mathrm{K}_c \|_2^2 \text{ - } 1 \right]^2$

Another version of gradient penalty can be a single sided loss as:

Single-sided GP loss = $\lambda \cdot max(0, \|\nabla_x \sum_c \mathrm{K}_c \|_F^2 - 1)$

Investigated paper claims two-sided gradient penalty is able to enforce sensitivity in model while single sided penalty is not able to do so. We will verify these claims empirically in sections to go.

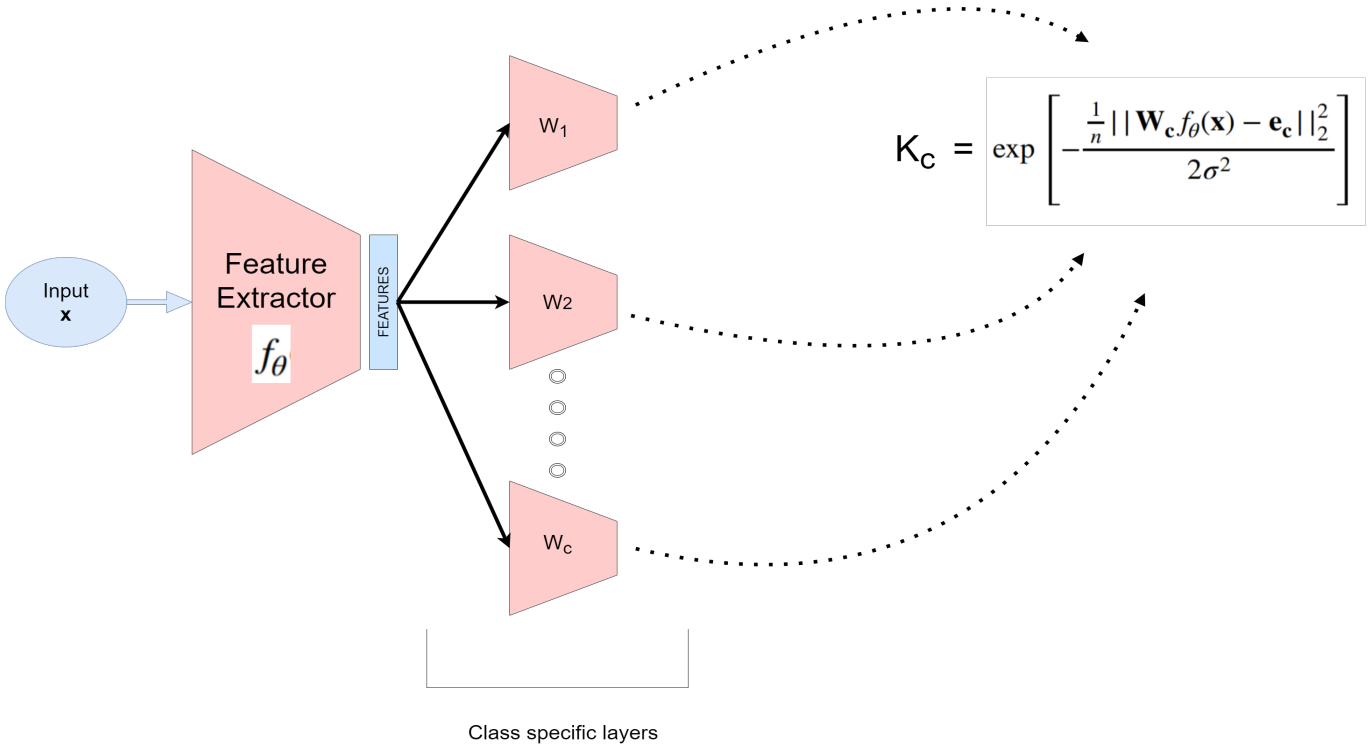


Figure 1: Figure describing complete algorithm, for each class it computes RBF score ($K_c$) which is the measure of OoD detection

## 3.2 Datasets

### 3.2.1 Two-moons dataset

It is a two dimensional dataset. As the name suggest, it has two intertwined classes on shape of the crescent. We obtain it using Sciket-learn implementation as described by authors.

### 3.2.2 Fashion-MNIST and MNIST

Fashion-MNIST is a dataset 28x28 grayscale images consisting of training set of 60,000 images and test set of 10,000 images each belonging to either of 10 classes. MNIST is a database of 28x28 gray scale handwritten digit images having training set of 60,000 and a test set of 10,000 images. We obtain both datasets using Pytorch's dataset module. As we are testing proposed algorithm on out of distribution detection Fashion-MNIST, MNIST have been notably difficult pair for this task [2].

### 3.2.3 CIFAR-10 and SVHN

CIFAR-10 is a dataset of 60,000 32x32 colour images in 10 classes, having 50,000 training images and 10,000 test images. SVHN is a real-world image dataset similar in flavor to MNIST, but incorporates an order of magnitude more labeled data with 604,388 training set images and 26,032 images for testing.
Similarly we will evaluate algorithm on CIFAR-10 vs SVHN OoD detection [2].

| Method | Accuracy(FM) | AUROC(M) | Train time(in sec) | Inference time |
|---|---|---|---|---|
| Deep Ensemble | 93.30 | 0.889 | 45 | 2.3 |
| DUQ | 92.13 | 0.947 | 23 | 1 |
| E-DUQ | 92.35 | 0.964 | 23 | 1 |

Table 1: Result comparison on model trained on Fashion MNIST by different methods, AUROC(M) is for separating Fashion-MNIST from MNIST. EDUQ is described in section 4.2.2

| Method | Accuracy | AUROC | Train time(in sec) | Inference time(in sec) |
|---|---|---|---|---|
| Deep Ensemble | 94.44 | 0.949 | 300 | 14 |
| DUQ ($\lambda$=0.5) | 93.45 | 0.931 | 210 | 4 |

Table 2: Result comparison on model trained on CIFAR-10 by deep ensemble and DUQ, AUROC reported is for separating CIFAR-10 from SVHN

### 3.3 Experimental setup

Author's training code is available on GitHub which was quite useful for us. We have added our codes for all analysis and experimentation on GitHub `https://github.com/dsgiitr/ML-Reproducibility-2020-DUQ`. We trained all models on NVIDIA Tesla T4 GPU.

We followed Appendix of the reported paper for architecture, optimization, number of epochs, batch size and generating data in case of two-moons.
Execution of the code is not computationally extensive. In the described experimental settings training on MNIST takes 15 minutes while training on CIFAR-10 takes 4 hours. Maximum 3 GB of GPU and 4 GB of RAM is sufficient to run the code.

## 4 Results

In this section we provide our empirical analysis on claims in Section 2, first on toy dataset, two-moons and then extend analysis for Fashion-MNIST and CIFAR-10. Our experiments verify the claims of paper with similar results on these datasets.

### 4.1 Two-moon

Our implementation on two-moons for deep ensemble, DUQ without gradient penalty, with single and with two-sided penalty are shown in Figs 2, 3, 4, 5. Yellow colour represents region of certainty while blue region shows uncertainty. It can be seen DUQ with two-sided penalty is certain only near the training data and uncertain elsewhere, hence showing best uncertainty estimation over the region as compared to other methods. These figures illustrate how important two-sided penalty is for DUQ.

### 4.2 Fashion-MNIST

DUQ is trained on Fashion-MNIST and its ability to detect MNIST datapoints as OoD is measured by AUROC, larger being the better. In order to select optimal $\lambda$ (gradient penalty factor) we evaluate AUROC for a third dataset, Not-MNIST as OoD and pick the $\lambda$ with highest AUROC (NM). We show results for different $\lambda$ in Table 5. These results follow similar trend as mentioned in paper. Further, we also do analysis over some of the other important hyperparameters in Section 4.2.3

In Table 1 we compare DUQ with deep ensemble, proposed approach clearly outperforms deep ensemble in AUROC as well as on training and inference time.
In an algorithm which is able to detect OoD points, another measure of performance could be rejection classification, in which we mix two-datasets and reject points on the basis of uncertainty predicted by model and expect increase in accuracy.
In Fig 8 we find that rejection classification performance on Fashion-MNIST vs MNIST for DUQ in comparison with deep ensemble gives slightly more accuracy increase per rejection. Thus, our analysis supports the claims made by the

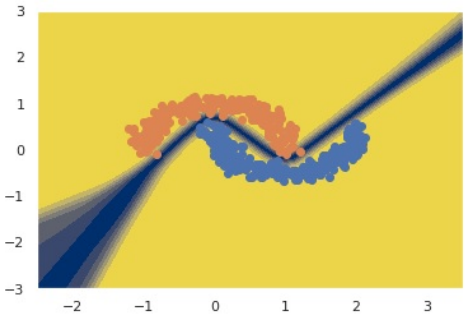


Figure 2: Uncertainty estimation using Deep Ensemble

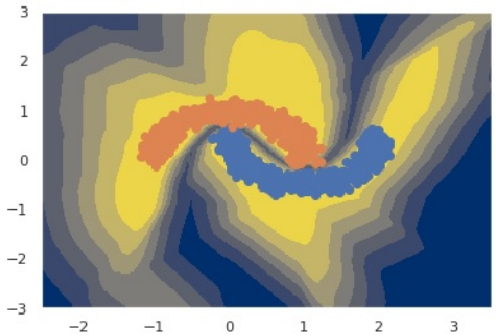


Figure 3: Uncertainty estimation using DUQ with $\lambda$=0

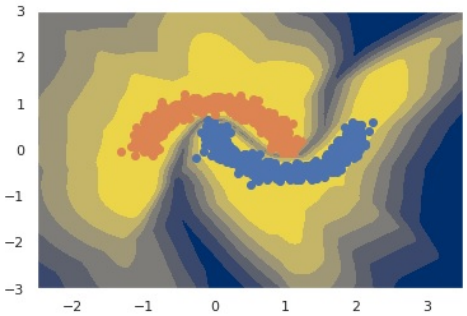


Figure 4: Uncertainty estimation using DUQ with single sided gradient penalty

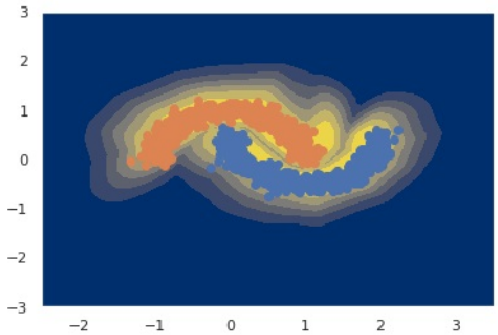


Figure 5: Uncertainty estimation using DUQ with two-sided penalty

authors.

#### 4.2.1 Aleatoric sensitivity

Authors provide OoD detection analysis on Fashion-MNIST vs MNIST datapoints which are fundamentally different datasets. Paper mentions that DUQ captures both aleatoric and epistemic uncertainty but doesn't provide any analysis on these datasets.

To evaluate how sensitively DUQ detects noisy data, we plot rejection classification performance of DUQ on Fashion-MNIST vs Noisy Fashion-MNIST as shown in Fig 6. It can be clearly seen that DUQ is able to even detect Noisy Fashion-MNIST datapoints with very small gaussian noise whereas deep ensemble is unable to do so. This clearly state that DUQ is very sensitive in aleatoric noise detection and deems even very low noise datapoints as OoD which can even be undesirable for practical purposes, as extreme sensitivity will harm model's decision making ability.

#### 4.2.2 Learnable $\sigma$ : Modelling noise through input

We propose to incorporate noise computation by predicting length scale($\sigma$) for each datapoint rather than keeping it constant as done by [3]. We name this extended approach as Explicit-DUQ or E-DUQ.

If noisy points are predicted with high $\sigma$, it will increase RBF score as shown in Fig 1 hence disabling model to predict noisy points as OoD. Hence E-DUQ should label only fundamentally different data distributions (i.e. epistemic uncertainty) as OoD.
In E-DUQ the feature extractor outputs features along with one extra logit for length scale, it will be very similar to model described in Fig 1, but now along with $f_\theta(x)$ length scale($\sigma$) will also be provided to compute $K_c$.

Fig 7 shows histogram of magnitude of $\sigma$ predicted by E-DUQ on Fashion-MNIST and Noisy Fashion-MNIST test set, it can be seen that model predicts higher values of $\sigma$ for noisy datapoints.
Rejection classification plot of E-DUQ (Fig 6) lies well below than that of DUQ which validates E-DUQ does not label noisy points as OoD.

Table 1 illustrates E-DUQ outperforms DUQ trained on Fashion-MNIST in AUROC(M), this is because MNIST datapoints are OoD with respect to Fashion-MNIST in terms of epistemic uncertainty, which E-DUQ models explicitly. With the above emperical analysis we can say it is possible to predict epistemic and aleatoric uncertainties independently using RBF score($K_c$) and length scale($\sigma$) respectively.

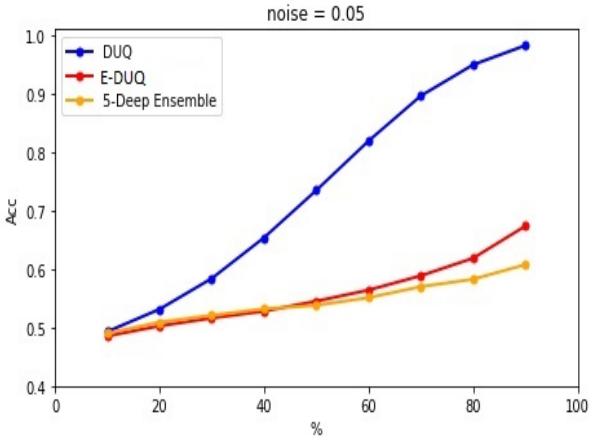


Figure 6: Rejection classification performance by different methods for Fashion-MNIST vs Noisy Fashion-MNIST. (X-axis is the % of data rejected using uncertainty estimates, Y-axis is the corresponding accuracy)

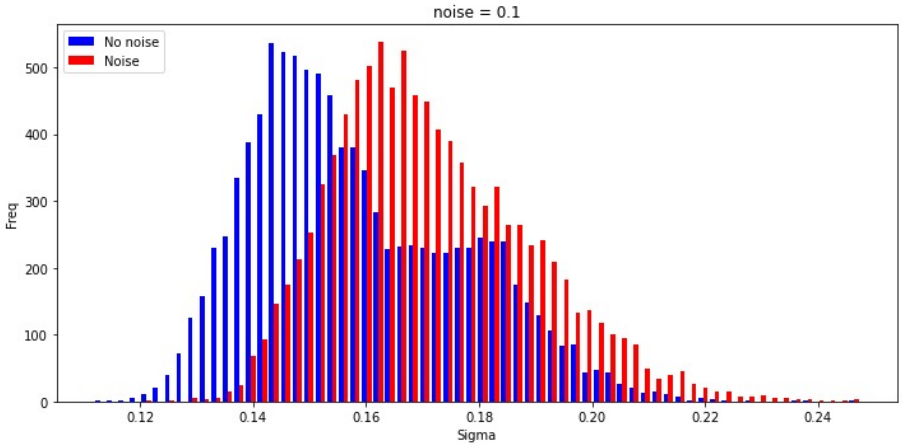


Figure 7: Histogram of values of length scale(sigma) predicted by E-DUQ for Fashion-MNIST and Noisy Fashion-MNIST

#### 4.2.3 Other Hyperparameters and Ablations

Table 3 shows how performance vary with dimension of centroid vector ($z$), we note AUROC first increase with $z$ and then saturates while accuracy remains unaffected. Authors used $z = 256$ which according to us is a descent choice.

We also experiment how varying the constant number($c$) in two-sided gradient penalty affect training, in Table 4 we fix $\lambda$=1 and experiment for different values for $c$, we find its performance to be better than model trained with optimal $\lambda$ in table 5 which shows $c$ can be a crucial hyperparameter.

| Centroid dimension | Accuracy | AUROC(M) |
|---|---|---|
| 10 | 92.15 | 0.887 |
| 256 | 92.13 | 0.947 |
| 500 | 92.32 | 0.959 |
| 1000 | 92.11 | 0.955 |

Table 3: Accuracy and AUROC(M) trend with centroid-vector dimension on Fashion MNIST.

| $c$ | Accuracy | AUROC(M) |
|---|---|---|
| 0.01 | 91.89 | 0.928 |
| 0.1 | 92.23 | 0.951 |
| 0.2 | 92.36 | 0.953 |
| 0.5 | 91.15 | 0.918 |

Table 4: Accuracy and AUROC(M) trend with $c$ for GP loss = $\left[\|\Delta_x \sum_c \mathrm{K}_c\|_2^2 - \mathrm{c}\right]^2$ on Fashion-MNIST

| $\lambda$ | Accuracy | AUROC(NM) | AUROC(M) |
|---|---|---|---|
| 0 | 92.39 | 0.941 | 0.941 |
| 0.05 | 92.13 | 0.962 | 0.947 |
| 0.1 | 92.11 | 0.953 | 0.942 |
| 0.2 | 92.03 | 0.929 | 0.954 |
| 0.3 | 92.11 | 0.941 | 0.946 |
| 0.5 | 91.94 | 0.939 | 0.931 |
| 1.0 | 91.12 | 0.895 | 0.901 |

Table 5: Accuracy and AUROC trend with $\lambda$ for model trained on Fashion-MNIST.

### 4.3 CIFAR-10

Till now we have validated the central claim of the paper on Fashion-MNIST dataset, in this subsection we provide few experiments to note how well DUQ scales to larger datasets.
In Table 2 we compare DUQ with deep ensemble trained on CIFAR-10, deep ensemble performs slightly better than DUQ in terms of accuracy and AUROC however DUQ is much better in terms of computational time.

We obtain similar rejection classification plot for CIFAR-10 vs SVHN as reported in paper, we see similar performance by both DUQ and deep ensemble as shown in Fig 9.

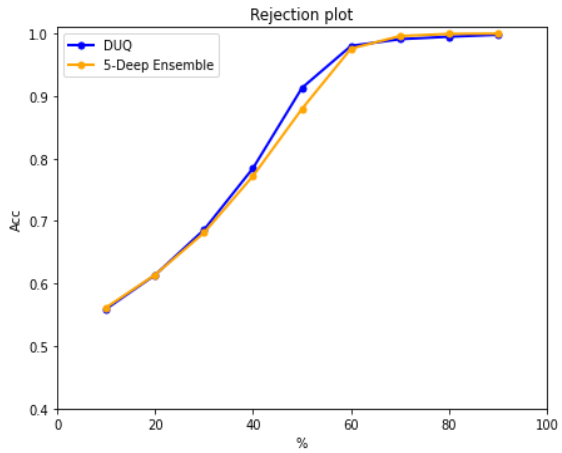


Figure 8: Rejection classification plot for deep ensemble and DUQ for Fashion-MNIST vs MNIST, x-axis is the % of data rejected using uncertainty estimates. Y-axis is the corresponding accuracy of prediction.

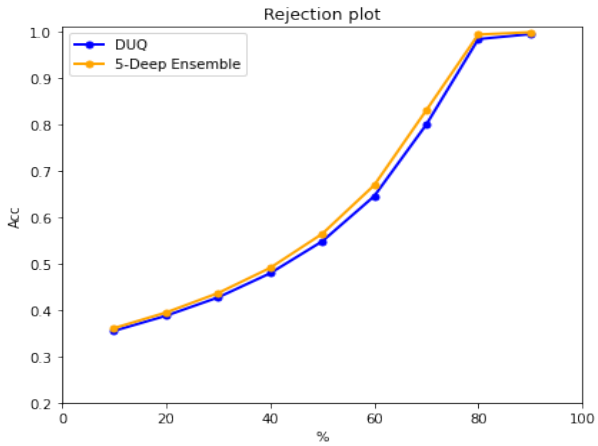


Figure 9: Rejection classification plot for deep ensemble and DUQ on CIFAR-10 vs SVHN. (X-axis is the % of data rejected using uncertainty estimates. Y-axis is the corresponding accuracy)

We provide prediction histogram of DUQ on CIFAR-10 vs SVHN as shown in Fig 10. Plot for CIFAR-10 is skewed towards the right extreme depicting higher certainty whereas a uniform plot for SVHN indicates uncertainty. These results support the claims from paper.

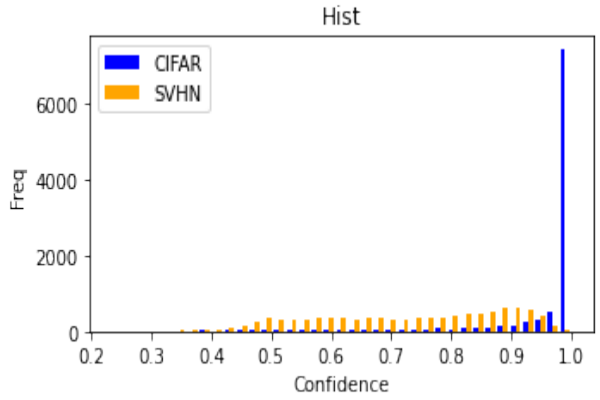


Figure 10: Uncertainty histogram of CIFAR-10 and SVHN for DUQ ($\lambda = 0.5$) trained on CIFAR-10

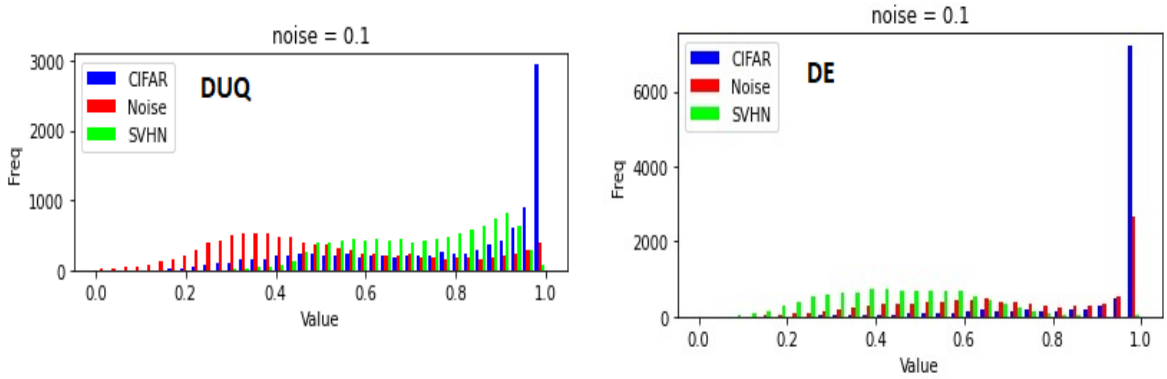


Figure 11: Uncertainty histogram of CIFAR-10 , SVHN and Noisy CIFAR-10 for DUQ ($\lambda = 0.5$) and deep-ensemble trained on CIFAR-10

We also provide similar histogram on Noisy CIFAR-10 as shown in Fig 11. We see that DUQ is unconfident with Noisy CIFAR-10 points whereas deep ensemble is able to classify them with certainty, this finding is in line with that of Section 4.2.1.

## 5 Discussion

Simple approach along with data and models involved being in our computational limits helped us in the reproduction of results. We are able to validate most of the results and trends as reported in the paper. Paper mentioned work on aleatoric and epistemic uncertainty detection is required, our naive extension of predicting $\sigma$ inspired by [3] is able to even outperform author's results on Fashion-MNIST along with explicit control over uncertainty detection which is required for practical purposes. These preliminary results are encouraging for future research to be done in this direction.

## References

[1] Joost van Amersfoort, Lewis Smith, Yee Whye Teh, and Yarin Gal. Uncertainty estimation using a single deep deterministic neural network. 2020.

[2] Eric T. Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Görür, and Balaji Lakshminarayanan. Hybrid models with deep and invertible features. In *ICML*, pages 4723–4732, 2019.

[3] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 5574–5584. Curran Associates, Inc., 2017.