Combating Label Noise in Selective Classifier using Abstention

Chuchen Deng chuchend@usc.edu
Haobo Xu haobox@usc.edu
Kai Wang kwang425@usc.edu
Ren Zhong rzhong@usc.edu

Abstract

Selective prediction is a powerful tool which enables machine learning models to make more confident predictions. In this project, we conducted experiments to examine the performance of DNN-based selective classifier on image classification tasks when training label noise exists. Then we combined the Data Abstaining Classifier (DAC) as a training data cleanser with the DNN model, and compared the new selective classifier with the original DNN classifier. Our results shows that the Post DAC DNN selective classifier significantly outperforms the baseline either when the training data is uniformly corrupted or has randomized single-class label noise. We also analyzed the accuracy and robustness of DNN selective classifier on Out-of-Distribution data.

1 Introduction

During the recent years, the deep neural network (DNN) has shown impressive performance in a variety of machine learning tasks, such as image classification and speech recognition (LeCun et al., 2015). The well-trained models can make accurate predictions on many difficult and time-consuming problems. However, such functionality of always returning a prediction of a given input is undesired for some high-risk systems (Asif et al., 2020). For instance, if the X-ray scanning image of a patient is blurry, a human doctor will not confirm the patient's prescription until further tests are made. The artificial intelligence is expected to behave like humans to reject making uncertain predictions.

One way to accomplish this is to apply selective prediction in the machine learning models. The selective prediction approach was firstly introduced by Chow in 1957, which focuses on the Bayesian solution on the data with fully known distribution. This method intends to train a selective classifier which consists of two models, one to perform classification and the other to decide to accept the prediction or not (El-Yaniv et al., 2010). In 2017, Geifman and El-Yaniv proposed a selective classification technique for optimizing the risk-vs-coverage curve based on the Softmax output of the trained model. Their results reveals that it is possible to perform high confident classifications (>98%) on the majority of the inputs (>60%).

The success of current DNN models, including the selective classifier is based on the large quantity of carefully labelled training data. However, data labeling is a very complex and time-consuming process, and the label noise can naturally occur when human experts are involved (McNicol, 2005). Although crowdsourcing marketplace like Amazon Mechanical Turk have enables the quick and low cost large-scale data annotation, possible label noise is the inevitable shortcoming of this platform (Zhang and Sabuncu, 2018). The reported ratio of corrupted labels in open-source datasets ranges from 8.0% to 38.5% (Song et al., 2020).

To cope with the challenge of noisy labels, many approaches have been proposed. Thongkam et al. mentioned the K-fold cross validation scheme, which involves in training multiple classifiers and voting out the likely-corrupted data. Ghosh et al. revealed that the loss function, Mean Absolute Brror (MAE), can be robust against mislabeled data. Another famous approach is adding a noise addptation layer on the top of DNN architecture to learn the label transition pattern (Chen and Gupta, 2015).

The main contribution of this project is in three aspects. First, we examined the performance of DNN-based selective classifier on *CIFAR-10* image classification tasks when the training set is corrupted with different fractions of label noise. Second, we implemented a Data Abstaining Classifier (DAC) as a data cleanser to filter the corrupted data before sending it to the DNN model, and conducted several experiments to compare the

DNN-based and Post DAC DNN-based selective classifiers. In addition, the DNN classifier's performance on Out-of-Distribution (OOD) test data was assessed.

The remainder of the report is structured as follows. In Section 2, we discussed the existing studies of this topic. Section 3 introduces the definition and notation of the selective classifier trained with noisy data problem. In Section 4 and Section 5, the description of the models and experimental settings are demonstrated. In Section 6, the results of our experiments are presented and thoroughly discussed. Section 7 discussed our findings and proposed some future directions. Section 8 is the conclusion of our paper.

2 Related Work

The reject option technique received extensive attentions in recent years. Cordella et al. proposed a method for improving classification reliability by setting a confidence score to get best trade-off between reject rate and misclassification rate. A more advanced confidence score method named Monte-Carlo dropout (Gal and Ghahramani, 2016) is published in 2016, which obtains the confidence score from the statistics of many forward passes in a dropout network. Geifman and El-Yaniv introduced the DNN-based selective classifier in 2017. In that paper, they proved that the selective classifier could make confident predictions with controlled risk. The SelectiveNet (Geifman and El-Yaniv, 2019), which integrated the rejection option into the neural network classifier, was reported. This new structure outperforms many known selective classification methods and may point to the future direction of the selective classifier.

Another related topic of this project is the learning with noisy labels. Han et al. presented the co-teaching method which trains a pair of networks in parallel and only do gradient updates on those examples which are classified with low loss in the other model. Zhang and Sabuncu introduced a generalized noisy-robust MAE loss function that can be employed with any existing DNN algorithm. The DAC method, which is applied in this project, is proposed by Thulasidasan et al.. The paper indicates that the downstream DNN models can be significantly improved when the data-cleaning with the DAC is utilized. More detailed theoretical and implementation will be discussed in the following sections.

3 Problem Statement

Selective classifiers have been analyzed on deep neural networks before, and many different sources of label noises in our real world would be a factor affecting the performance of neural networks. Plus, the effect of training label noise on selective classifiers and the functional approach to do data cleansing remains an unknown area. Therefore, we want to discuss the effect of training label noise on selective classifiers and any other methods to improve the performance of selective classifiers. Moreover, out-of-distribution (OOD) data are also considered as noise added in the testing data, which the corresponding performances are displayed in the Results part. Additionally, different strategies of adding label noise during training are illustrated in the later part of this paper.

In this project, we constructed a selective classifier for an image classification task and plotted the risk-vs-coverage (RC) graph from (Geifman and El-Yaniv, 2017) and added label noise in the training data and added OOD data as noise in the testing data to analyze the effect of label noise. Plus, we also constructed a deep abstaining classifier (Geifman and El-Yaniv, 2017) to cleanse the label noise in training data, in order to improve a better performance on this image classification task.

4 Methodology

4.1 Selective Classification

A selective classifier is a pair (f,g), where f is a classifier, and g is a selection function, which serves as a binary qualifier for f to map input to 0 or 1, which is defined that:

$$(f,g)(x) \triangleq \begin{cases} f(x), & g(x)=1 \\ don't\ know, & g(x)=0 \end{cases} \tag{3}$$

Thus, the selective classifier abstains from prediction at a point x if and only if g(x) equals 0. And the performance of a selective classifier is quantified using coverage and risk. According to our baseline (Geifman and El-Yaniv, 2017), we define g(x) as follows.

$$g_\theta(x) = g_\theta(x|f(x)) \triangleq \begin{cases} 1, & f(x) \geq \theta \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

Also, we define the risk R(f,g) of the selection classifier as follow:

$$R(f,g) = \frac{wrong\ predictions}{all\ predictions} \tag{5}$$

Using binary search (Geifman and El-Yaniv, 2017), we can find θ meeting the following risk requirement given confidence parameter δ, and a desired risk target r^*.

$$\boldsymbol{Pr}_{S_m}\{R(f,g) > r^*\} < \delta \qquad (6)$$

We want the prediction coverage to be as large as possible, as long as the risk requirement is met. So, we defined the optimum θ as the largest θ that meets the risk requirement. To build the classifier, we choose VGG-16 architecture with a few modifications by adding batching normalization and dropout layers. During training, the input to our ConvNets is a fixed-size 32 × 32 RGB image. The image is passed through a stack of convolutional(conv.) layers, where we use a kernel size of 3 × 3. The convolution stride and padding are fixed to 1 pixel. Each convolutional layer is followed by a ReLU and a batch normalization layer. Between every two convoluted layers, we added a dropout layer with a dropout rate from 0.3 to 0.5,which increases with the layer depth. Max-pooling is performed between every two convolutional layers over a 2 × 2 pixel window, with stride 2. Two Fully-Connected (FC) layers come after a stack of convolutional layers: the first one has 512 channels each, and the second one has 10 channels for classification. The ConvNet configurations used in this project are outlined in Appendix.

4.2 Deep Abstaining Classification

Deep Abstaining Classification is a k-class classifier with a deep neural network where the input is x and the output is y. For any given input x, we define its probabilities of the ith class as

$$p_i = p_w(y = i|x) \qquad (1)$$

where w is the weight of our deep neural network.

According to (Thulasidasan et al., 2019), in contrast to the standard cross-entropy deep neural network's training loss,$L_{standard} = -\sum_{i=1}^{k} t_i * \log p_i$, where t_i is the ground-truth label, DAC has an additional $(k+1)^{st}$ output probability indicating the "abstention".

We train the DAC with the modified k-class cross-entropy loss function:

$$L(x_j) = (1 - p_{k+1})(-\sum_{i=1}^{k} t_i * log\frac{p_i}{1 - p_{k+1}}) + \alpha log(\frac{1}{1 - p_{k+1}})$$

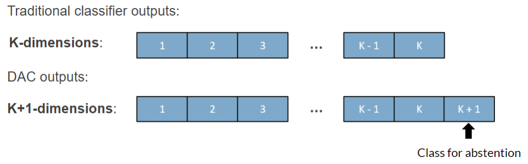


Figure 1: The DAC classifier uses an additional class to collect the data for abstention

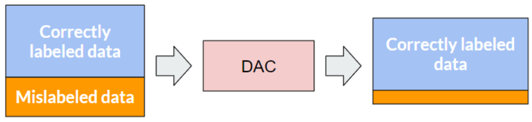


Figure 2: The DAC model detects and abstains the training data before sending it to the downstream DNN model

If $p_{k+1} = 0$, the above loss will be exactly the same with the standard cross-entropy loss. If $p_{k+1} > 0$, the first term would be a modified version of the standard k-class cross-entropy loss where the abstention mass has been normalized out. The second term represents the abstention penalty factored by α . Higher α means a higher degree of abstention penalty, which drives the model less likely to abstain. On the other hand,lower α drives the model more likely to abstain on unsure samples. We have also follows algorithm to auto-tuning , where we start with a small α value and gradually increase it. This will encourage, at the early stage, the model to abstain all examples except the easiest based on what it learned so far. As the training epochs increase, the α value will be increased as well, regularizing the model's behavior to abstain only samples it is really unsure. Deep Abstaining Classification (DAC) has been shown its capability as a data cleaner in both structured and unstructured label noises in the training phrase. Structured label noise occurs when there is a co-occurrence relationship between the noise and certain features, whereas unstructured or random label noise occurs randomly.

In our project, we intend to use DAC as an extra data cleaning procedure before training the deep neural network and applying softmax selective classification techniques on top of it. In other words, DAC serves as an additional layer between the original dataset which possibly contains noises and the neural network to reduce the noise level in the dataset before training.

4.3 Adding Label Noise

In this project, we analyzed the effect of label noise for selective classification, where we tested different noise rates with 0.1, 0.2 and 0.4 to randomly mislabel our training data before training, comparing to the training data without any noise as baseline experiment. Moreover, to make the training data with label noise more realistic to the real world data, we also chose one class as the "target" class which has the lowest accuracy among all classification labels, and test different noise rates with 0.1, 0.2, and 0.4 to mislabel the data in the "target" class before training. All noise data in the training dataset are assigned to a label randomly other than the one which it belongs to.

4.4 Adding Out-of-Distribution Data

Considering more resources of noise in our real world, the out-of-distribution data should also be experimented during tests. Thus, we also tested different rates with 0.1, 0.2, and 0.4 to randomly add out-of-distribution data from the OOD dataset we chose to the testing dataset. To make sure the size of noise testing data is the same as the size of testing data without any noise, we randomly sampled images from the OOD dataset with different rates proportional to the size of the original testing data. The original testing data and noise testing data are loaded before training because we want to make sure that the unpolluted part of data in the noise testing data is the same as the original testing data. In this way, we could train our selective classifier once, and test both on the original dataset, and the noise testing data with OOD data in it.

4.5 Training pipeline

To simulate real-world situations, we assume that the training and testing data set can both be noisy. The training set could be noisy because of annotation errors mentioned before, and the testing set could be noisy since it might contain OOD data that are out of domain. Thus, our ultimate goal is to learn a selective classifier to abstain from uncertain testing samples while making the coverage as much as possible. We expect a selective classifier trained with less noisy training data would outperform the one with more noisy training data. So before feeding the training samples into the training procedure of the selective classifier, we try to make use of the power of DAC, which serves as a data cleaner to filter out most of either the structured label noises or random label noises. So we firstly use the training data which could contain noisy labels to train a DAC and then apply the trained DAC on the same training data. As specified in 4.2, for each sample , if the predicted class is k+1, it would indicate it should be filtered out. Additionally, if the predicted class is different from the sample's label, we also consider it as noisy data. We expect the DAC will filter out most of the label noises and then feed the filtered training set into the deep neural network. Finally, we use its softmax score as a selective classifier to abstain all the OOD data in the test set. We would like to compare our post-DAC DNN with the baseline DNN.

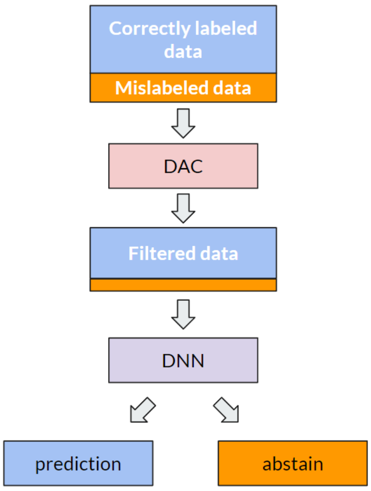


Figure 3: Training pipeline

5 Experiment

5.1 Datasets

In this study, we used the *CIFAR-10* dataset provided in library torchvision and downloaded before training as our training dataset, which consists of 60000 32 × 32 color images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images. For out-of-distribution (OOD) dataset, we chose SVHN dataset and FakeData dataset which include house numbers images and randomly generated images correspondingly. They are also provided in library torchvision and downloaded before training. In that color images in SVHN dataset and FakeData dataset are not related to the color images in the *CIFAR-10* dataset and classified into 10 classes of *CIFAR-10* dataset, they belong to out-of-distribution data in this situation.

Method	Parameter
Horizontal flip	Probability = 0.5
Rotation	Degrees ∈ (-15,15)
Image shift	Translate ∈ (0.1,0.1)

Table 1: Data Augmentation Methods

5.2 Data Preprocess

The preprocessing of our training dataset contains several parts to generate more various pictures because the work done in (Geifman and El-Yaniv, 2017) also used the data augmentation method to make the model more generalized. First, we normalize the training images for each channel with 0.5 mean and 0.5 standard deviation, which would facilitate the model to converge in less time. Second, we allowed that pictures with 50% probability could be flipped horizontally to generate more samples. Third, more training images are produced by rotating 15 degrees clockwise and counterclockwise. Moreover, we transformed the training dataset images to tensors and allowed that images could be translated 0.1 times horizontally and vertically. Overall, all methods about data augmentation are displayed in Table 1.

For the testing dataset, we also transformed the color images to tensors and normalized the testing images and testing images for each channel with 0.5 mean and 0.5 standard deviation. Plus, we had to resize the color images to 32 × 32. Because the size of testing images in *CIFAR-10* dataset is 32 × 32, to make sure the out-of-distribution images added to the testing dataset have the same size, we resize them to 32 × 32 pictures.

5.3 Adding Label Noise

As discussed in 4.3, we simulate two kinds of situations: normally distributed label noise over all classes and normally distributed label noise over a single class. Randomly noise distribution over all classes represents the unstructured noise. To simulate such a situation, we randomly pick portions of the training samples and flip their label to one of the other nine classes. Noise over a single class represents structured noises, a case where one of the classes might be easily recognized as other classes in the real-life environment. We pick the class “cat” and pick a portion of the sample, flip it to one of the other nine classes. For both cases, we consider picking 10%, 20% and 40% the number of samples of all data. For the first case, the total number of noisy data is 4000/8000/16000 and for the second case, the total number of noisy data is 400/800/1600.

noise rate	Accuracy
0.0	93.24%
0.1	86.59%
0.2	83.52%
0.4	69.56%

Table 2: Model accuracy on *CIFAR-10* under different uniform label noise rate

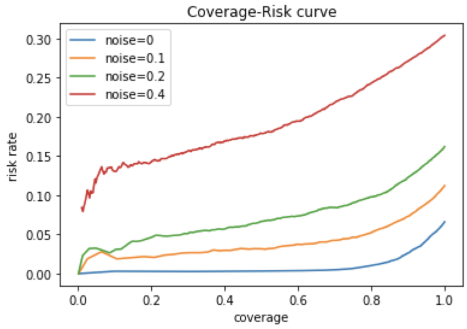


Figure 4: Risk coverage curves for selective classifier trained with noise rate = (0, 0.1, 0.2, 0.4)

5.4 Applying DAC and DNN

We trained a deep classification classifier on the noisy data set of different noise levels. After that, we applied the DAC on the same noisy dataset and filtered out any sample with a different label than the DAC’s output of it. In such a way, DAC serves as an effective data cleaner to reduce the noise level in the training dataset before feeding it to the deep neural network. We then trained a deep neural network specified in 4.1 based on the data filtered by DAC.

6 Results

6.1 DNN-based Selective Classifier on Uniform Label Noise

The first part of our experiment is to examine the performance of DNN-based selective classifier. Firstly, we reconstructed the DNN-based selective classifier as the baseline. For the *CIFAR-10* classification task, our VGG-16 model achieved similar accuracy (93.24%) as the baseline (93.54%).

we focus on **uniform label noise** in training set. In Table 2, we present the model accuracy under different label noise rate. The prediction accuracy slows drops when the noise begins to introduce.

noise rate	r^*	risk	coverage
0.0	0.05	0.0330	0.939
0.0	0.1	0.0681	0.999
0.0	0.2	0.0611	0.999
0.1	0.05	0.0291	0.688
0.1	0.1	0.0725	0.814
0.1	0.2	0.1011	0.999
0.2	0.05	N/A	0.0
0.2	0.1	0.0708	0.72
0.2	0.2	0.1542	0.999
0.4	0.05	N/A	0.0
0.4	0.1	N/A	0.0
0.4	0.2	0.1505	0.432

Table 3: DNN Risk-coverage under different confidence level (r^*) and different uniform label noise rate

And the dropping rate is accelerated as the noise fraction increases. In Figure 4, the risk-coverage curve for these models was shown and compared. It shows the obvious risk-coverage trade-off in selective classifiers, which means that selective classifiers have ability to decrease risk rate at the cost of coverage. Also, we can see label noise could have huge impact on the result of these classifiers. To cover the same amount of data. The risk of the prediction increases exponentially with respect to the noise rate. But even when the noise rate is 0.4 and the original classifier has a risk rate of 0.3, the selective classifier can still give relatively more confident predictions (around 0.2) by predicting half of the testing data.

In Table 3, we show the risk rate, coverage of both validation set and test set for different r* levels under different level of noise label rate, given confidence parameter δ=0.05. From these results, we can see the selective classifiers can effectively increase the performance of original classifier at the sacrifice of coverage. When we increase the label noise, which means that the accuracy of original classifier reduces, desired risk target r* must be increased in order to get reasonable coverage. For the original dataset, the accuracy of prediction can be increased to 98% when the coverage is around 80%. Also, for dataset with very huge label noise, the selective classifier can give more certain predictions for a portion of the test set.

noise rate	DNN	Post-DAC DNN
0	93.24%	93.37%
0.1	86.59%	90.54%
0.2	83.52%	87.54%
0.4	69.56%	83.53%

Table 4: Comparison of accuracy of Post-DAC DNN against DNN model

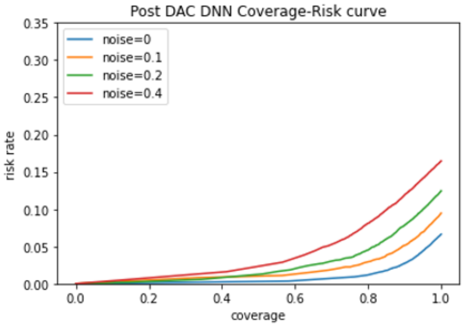


Figure 5: Post-DAC DNN Risk coverage curves for selective classifier trained with uniform noise

6.2 Post-DAC DNN-based Selective Classifier on Uniform Label Noise

In order to improve the robustness of selective classifiers to handle noisy labels, the DAC method is applied to detect and abstain the noise labels in training set. A DAC model was trained using the corrupted training set. Then, we train VGG-16 models with same structure on the training set filtered by the DAC model. In Figure 5, the risk-coverage curve for these models was shown and compared. In Table 4, we present the model accuracy of DNN and Post-DAC DNN under different label noise rate. By comparing the results with simple DNN model generation, we can see a huge improvement. For the original training set, DNN and Post-DAC DNN give similar results, which means DAC does not discard too much data when they are all labeled correctly. When it comes to corrupted training set, Post-DAC DNN shows strong ability to recognize the noise data and make correct prediction. Especially when noise rate is relatively high, e.g. 40%, Post-DAC DNN improved the accuracy of 14% than the DNN model with same structure.

Based on the fact that the amount of correct data received by the model will be reduced when adding noise data to the training set, this means that the difference in accuracy under different noise levels may be due to the influence of this factor. In Figure

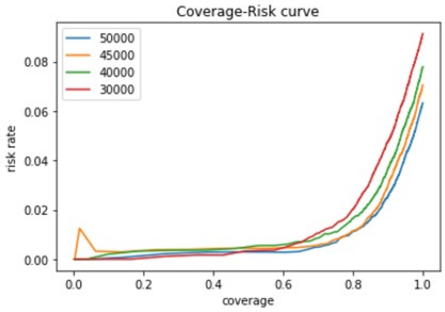


Figure 6: DNN Risk coverage curves for selective classifier trained with different amount of data

noise rate	r^*	risk	coverage
0.0	0.05	0.0419	0.9490
0.0	0.1	0.0670	0.9998
0.0	0.2	0.0674	0.9998
0.1	0.05	0.0331	0.8524
0.1	0.1	0.0901	0.9924
0.1	0.2	0.0940	0.9996
0.2	0.05	0.0340	0.7654
0.2	0.1	0.0973	0.9370
0.2	0.2	0.1226	0.9998
0.4	0.05	0.0410	0.6434
0.4	0.1	0.0865	0.8206
0.4	0.2	0.1696	0.9998

Table 5: Post-DAC DNN Risk-coverage under different confidence level (r^*) and different uniform label noise rate

6, the risk-coverage curve for the same structure DNN models trained with different size of training set was shown and compared. Here, there are 50000 data in the original training set. So 45000, 40000 and 30000 are the number of correct labeled data actually received by the model when the noise level is 0.1, 0.2 and 0.4 respectively. We can see that accuracy under different noise levels is not only influenced by model performance, but also by the decrease of correct labeled data as label noise increases.

And in table 5, we show the risk rate, coverage of test set for different r* levels under different level of noise label rate, given confidence parameter δ=0.05. From the result, we can see the potential of combining DAC and selective classifier to make more accurate prediction. Now, with Post-DAC DNN model, the risk can be limited to 5% by covering 64% the test set even at the noise rate of 0.4.

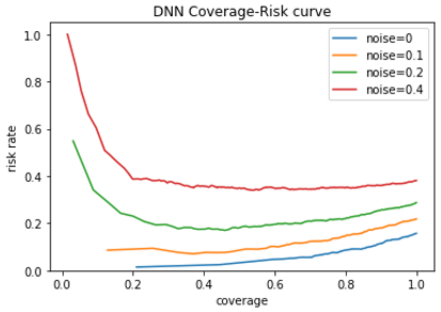


Figure 7: the "Cat" class Risk-Coverage curve of DNN for data corrupted with single class label noise

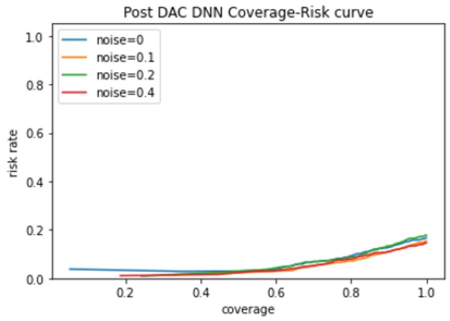
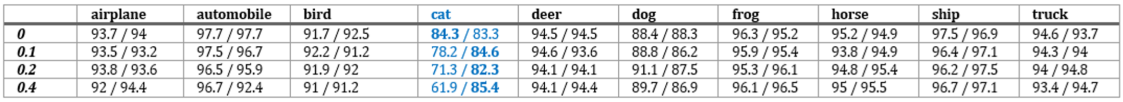


Figure 8: the "Cat" class Risk-Coverage curve of Post-DAC DNN for data corrupted with single class label noise

6.3 Single-class Training Set Noise

From above, the Post-DNN DNN model shows a good filtering ability for uniform noise data. The simultaneous use of selective classifier can improve the prediction accuracy of the model at the expense of coverage. Figure 7 and Figure 8 shows comparison of class-level accuracy of DNN against Post-DAC DNN model and the risk-coverage curve when only one class ("cat" in this case) has label noise. Because the principle of DAC is to divide the incorrectly labeled data into a single class, we expect it to have an excellent ability to identify single class noise, which is indeed the case. From Figure 9, for the class with noise, DAC can well identify the wrong labeled data, and maintain the prediction accuracy similar to that without noise under different noise levels. However, for simple DNN, they can't recognize the noise of single class, and the prediction accuracy decreases greatly with the increase of noise level. Also, from the graph, when the noise level is high, DNN coverage-risk curve is not monotonous, which means the MaxProb scores do not represent the confidence of the result and selective classifiers cannot be applied. For Post-DAC DNN models, the coverage-



	airplane	automobile	bird	cat	deer	dog	frog	horse	ship	truck
0	93.7 / 94	97.7 / 97.7	91.7 / 92.5	**84.3** / 83.3	94.5 / 94.5	88.4 / 88.3	96.3 / 95.2	95.2 / 94.9	97.5 / 96.9	94.6 / 93.7
0.1	93.5 / 93.2	97.5 / 96.7	92.2 / 91.2	78.2 / **84.6**	94.6 / 93.6	88.8 / 86.2	95.9 / 95.4	93.8 / 94.9	96.4 / 97.1	94.3 / 94
0.2	93.8 / 93.6	96.5 / 95.9	91.9 / 92	71.3 / **82.3**	94.1 / 94.1	91.1 / 87.5	95.3 / 96.1	94.8 / 95.4	96.2 / 97.5	94 / 94.8
0.4	92 / 94.4	96.7 / 92.4	91 / 91.2	61.9 / **85.4**	94.1 / 94.4	89.7 / 86.9	96.1 / 96.5	95 / 95.5	96.7 / 97.1	93.4 / 94.7

Figure 9: Comparison of class-level accuracy of DNN against PostDAC DNN model (DNN acc. / PostDAC DNN acc.

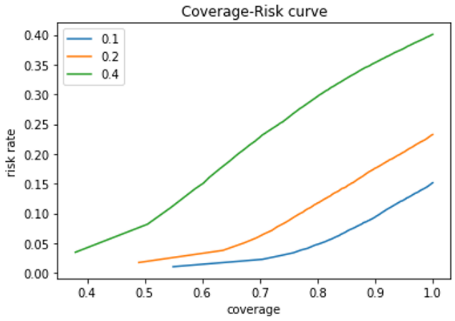


Figure 10: Risk-Coverage curve of DNN selective classifier on *CIFAR-10* testset polluted with different fraction of OOD data (FakeData)

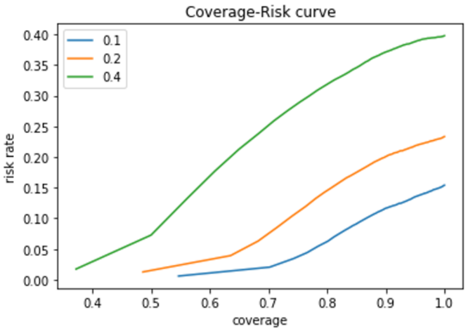


Figure 11: Risk-Coverage curve of DNN selective classifier on *CIFAR-10* testset polluted with different fraction of OOD data (SVHN)

risk curve remains monotonous, so selective classifiers can be applied to further improve the accuracy of prediction.

6.4 OOD Test Set Noise

We discussed about different types of label noise in training set, however, when OOD noise is added to test set, DNN selective classifier is not robust enough to address them. From Figure 10, Figure 11, and Table 6, we added different types of OOD data on test set. There is a corresponding drop on accuracy for different level of OOD data. Since the coverage-risk curve is still monotonous, selective classifier still has potential to increase the prediction accuracy at the cost of coverage.

noise rate	FakeData	SVHN
0.1	84.59%	84.86%
0.2	76.67%	76.75%
0.4	60.23%	59.94%

Table 6: Accuracy of DNN selective classifier on *CIFAR-10* testset polluted with different fraction of OOD data

7 Discussion and Future works

Some of the future works could be interesting as well. Firstly, we currently only considered random data noise, uniformly distributed in all classes or in a single class. It would be helpful if we can gather real-world training data with annotation of possible noisy data. And then apply the same method. In such way, a realistic simulation and result can be gained. Other than that, we are also interested in even adding some OOD data into the training set as well. The intuition is adding OOD data perhaps will be helpful for our model to learn the mapping between noisy labels and OOD features. Finally, we currently only test out VGG-16 model on CIFAR-10 dataset. We can also test the same method on other datasets or even other fields than image recognition to see if the pattern we discovered here is universal. We leave these tasks as future works.

8 Conclusion

In this project, we proposed a new noisy-robust selective classifier, Post DAC DNN, and proved its potential utilization on noisy training data. Our results shows that Post DAC DNN selective classifier significantly outperforms traditional DNN selective classifier either when the training data is uniformly corrupted or has randomized single-class label noise. In addition, Post DAC DNN selective classifier can predict most samples with low risk (¡ 0.02) even when the training data is highly corrupted. Furthermore, there is a corresponding drop on accuracy for different level of OOD data and DNN selective classifier is not robust enough to address OOD data.

9 Teamwork Partition

Each team member has equal contribution to the project. To be more specific, Kai Wang and Ren Zhong are responsible for DAC implementation, corrupted dataset construction, and noisy data experiment. Chunchen Deng is responsible for the construction and analysis of the selective classifiers. Haobo Xu is responsible for building DNN model and performing OOD noise test. All members contributed in writing the final reports.

References

Amina Asif et al. 2020. Generalized neural framework for learning with rejection. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Xinlei Chen and Abhinav Gupta. 2015. Webly supervised learning of convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 1431–1439.

Chi-Keung Chow. 1957. An optimum character recognition system using decision functions. *IRE Transactions on Electronic Computers*, (4):247–254.

Luigi Pietro Cordella, Claudio De Stefano, Francesco Tortorella, and Mario Vento. 1995. A method for improving classification reliability of multilayer perceptrons. *IEEE Transactions on Neural Networks*, 6(5):1140–1147.

Ran El-Yaniv et al. 2010. On the foundations of noise-free selective classification. *Journal of Machine Learning Research*, 11(5).

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

Yonatan Geifman and Ran El-Yaniv. 2017. Selective classification for deep neural networks. *Advances in neural information processing systems*, 30.

Yonatan Geifman and Ran El-Yaniv. 2019. Selectivenet: A deep neural network with an integrated reject option. In *International Conference on Machine Learning*, pages 2151–2159. PMLR.

Aritra Ghosh, Himanshu Kumar, and PS Sastry. 2017. Robust loss functions under label noise for deep neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. 2018. Co-teaching: Robust training of deep neural networks with extremely noisy labels. *Advances in neural information processing systems*, 31.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature*, 521(7553):436–444.

Don McNicol. 2005. *A primer of signal detection theory*. Psychology Press.

Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. 2020. Learning from noisy labels with deep neural networks: A survey. *arXiv preprint arXiv:2007.08199*.

Jaree Thongkam, Guandong Xu, Yanchun Zhang, and Fuchun Huang. 2008. Support vector machine for outlier detection in breast cancer survivability prediction. In *Asia-Pacific Web Conference*, pages 99–109. Springer.

Sunil Thulasidasan, Tanmoy Bhattacharya, Jeff Bilmes, Gopinath Chennupati, and Jamal Mohd-Yusof. 2019. Combating label noise in deep learning using abstention. *arXiv preprint arXiv:1905.10964*.

Zhilu Zhang and Mert Sabuncu. 2018. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems*, 31.

A Appendix

<table>
<tr><th>VGG-16</th><th>Our Configuration</th></tr>
<tr><td>16 weight layers</td><td>16 weight layers</td></tr>
<tr><td>Conv3-64
Conv3-64</td><td>Conv3-64
Dropout-0.3
Conv3-64</td></tr>
<tr><td colspan="2">maxpool</td></tr>
<tr><td>Conv3-128
Conv3-128</td><td>Conv3-128
Dropout-0.4
Conv3-128</td></tr>
<tr><td colspan="2">maxpool</td></tr>
<tr><td>Conv3-256
Conv3-256
Conv3-256</td><td>Conv3-256
Dropout-0.4
Conv3-256
Dropout-0.4
Conv3-256</td></tr>
<tr><td colspan="2">maxpool</td></tr>
<tr><td>Conv3-512
Conv3-512
Conv3-512</td><td>Conv3-512
Dropout-0.4
Conv3-512
Dropout-0.4
Conv3-512</td></tr>
<tr><td colspan="2">maxpool</td></tr>
<tr><td>Conv3-512
Conv3-512
Conv3-512</td><td>Conv3-512
Dropout-0.4
Conv3-512
Dropout-0.4
Conv3-512</td></tr>
<tr><td colspan="2">maxpool</td></tr>
<tr><td>FC
FC
FC
Softmax</td><td>Dropout-0.5
FC-512
ReLU
Batchnorm
Dropout(0.5)
FC-10</td></tr>
</table>

Table 7: Model Architecture