A SIMPLE TRAINING-FREE METHOD FOR REJECTION OPTION

Anonymous authors
Paper under double-blind review

ABSTRACT

We present a simple yet effective method to implement the rejection option for a pre-trained classifier. Our method is based on a sound mathematical framework, enjoys good properties, and is hyperparameter free. It is *lightweight*, since it does not require any re-training of the network, and it is *flexible*, since it can be used with any model that outputs soft-probabilities. We compare our solution to state-of-the-art methods considering popular benchmarks (CIFAR-10, CIFAR-100, SVHN), and various models (VGG-16, DenseNet-121, ResNet-34). At evaluation time, our method, which is applied post-training to any classification model, achieves similar or better results with respect to its competitors that usually require further training and/or tuning of the models.

1 INTRODUCTION

Deep Neural Networks (DNN) have gained a position of preference in many applications related to automated decision making. Consequently, extensive efforts are being made in several fields to make these systems more reliable as they prove to be error-prone (e.g., in computer vision (Gao et al., 2022; Cobb & Looveren, 2022), in autonomous driving (Amodei et al., 2016; Bicer et al., 2020), in NLP (Jin et al., 2022; Carlini et al., 2021), and in medical analysis (Subbaswamy & Saria, 2020; Bernhardt et al., 2022)). Due to the dramatic impact wrong decisions have in various applications, detecting them and avoiding them is of the essence.

Abstention as a way to avoid wrong decisions has been considered since the dawn of artificial intelligence (cf. Chow (1957)). The idea of a detector-based rejection strategy capable of distinguishing between 'secure' and 'non-secure' decision has been applied to multiple tasks. For instance, *novelty detection* and *out-of-distribution* (OOD) *detection* (Pimentel et al., 2014) focus on identifying sample which belongs to a novel class or are far from the training distribution, i.e. samples on which the decision should not be trusted; *misclassification detection* (Granese et al., 2021) focuses on detecting whether a prediction of a classifier is likely to be correct or not at test time, and therefore should be trusted or rejected respectively; *adversarial attack detection* (Szegedy et al., 2014) focuses on detecting whether a given input sample is a natural sample or a malicious one (i.e., if it has been perturbed with the purpose of fooling the target model), opting for rejection in the latter case.

In this paper we consider the problem of increasing the reliability of a model by equipping it with a *rejection option*. While standard models always give an answer related to the task they have learned when they are presented with an input samples, the rejection option enables them to reject the decision and abstain. Clearly, abstention raises the question of the trade-off between reducing the risk of making wrong decisions while keeping the number of abstentions as low as possible, therefore maintaining data coverage. *Selective classification* (Geifman & El-Yaniv, 2017) is one of the main areas of research invested in finding the aforementioned trade-off, by enabling basic models to express a confidence for their decision. Many works have been focusing on this problem and they have improved on the seminal paper (Geifman & El-Yaniv, 2019; Liu et al., 2019; Corbière et al., 2019). Learning how to achieve a given target coverage while maintaining a good classification accuracy is indeed a natural and powerful way to tackle the problem of finding the aforementioned trade-off. However, most of the times this means that standard classifiers cannot be used, since they need to be re-trained or tuned. This requires computational resources, time to fit the new parameters of the models and a (usually large) amount of samples.

To address this problem, we make the following contributions:

1. **Training-free rejection option.** We propose a new method to implement training-free rejection option for a given classifier neural network. We base our model on the Gini Impurity Score (Gini, 1912) which can been intented as an approximation of true probability of error classification. Up to our knowledge, we are the first to propose such a function in the selective context.
2. **Mathematically grounded lightweight and flexible selector.** Our proposed method is based on a state-of-the-art method for misclassification detection Granese et al. (2021)), which has been shown to improve on Geifman & El-Yaniv (2017). We revisit Granese et al. (2021)), underlining a connection between the score proposed in it and the Rényi divergence, and showing how such score can be used to implement the rejection option for a given pre-trained classifier. Our method is *lightweight* since it does not require any expensive re-training/fine-tuning of the network, and so *flexible* that any architecture can be used out of the shelf as long as it outputs a soft-distribution;
3. **Extensive experimental benchmark.** We evaluate the proposed method and we compare it with state-of-the art rejection option methods and selective classification frameworks. Our evaluation includes popular models (VGG-16, DenseNet-121, ResNet-34) and benchmark datasets (CIFAR-10, CIFAR-100, SVHN). Overall, we show that with our post-training method we achieve a performance which is comparable to state-of-the-art methods that require further training and/or tuning of the model and sometimes outperform them without requiring expensive and ad-hoc tuning of the network. We also showcase performance on the ImageNet dataset, demonstrating great scalability and ease of use out of the shelf.

The paper is organized as follows. In Sec. 2 we review the literature on the rejection option. In Sec. 3 we introduce our proposed selective model and we focus on the selector function on which it relies. In Sec. 4 we describe the evaluation setting with a particular attention to the metrics used to assess the performances of the proposed selective model. In Sec. 5 we present the results of the numerical experiments. The discussion to the limitation of this work and the final remarks are relegated to Sec. 5.3 and Sec. 6, respectively.

2 Related works

Rejection option has been studied since the dawn of artificial intelligence (cf. Flores (1958); Chow (1970)) as a way to avoid low-confidence decisions (Pudil et al., 1992), and miclassifications. Rejections can be split into two main groups (Hendrickx et al., 2021): ambiguity, i.e., when the learned model is not able to replicate the optimal decision in some areas of the input space (Hellman, 1970; Fukunaga & Kessell, 1972), and novelty, i.e., when inputs at test time are too dissimilar from those at train time (Vasconcelos et al., 1995; Seo et al., 2000; Vailaya & Jain, 2000)).

More recently, interest in the rejection option has increased again due to the popularity of deep learning. Among the most influential works, Hendrycks & Gimpel (2017) established a standard baseline for deep neural networks which relies on considering the maximum of the softmax distribution output by a model. Jiang et al. (2018) introduced a new confidence measure, which measures the agreement between the considered classifier and a modified nearest-neighbor classifier on an evaluation dataset. Gal & Ghahramani (2016) proposed using Monte Carlo Dropout (MCDropout) to estimate the posterior predictive network distribution by sampling several stochastic network predictions.

Corbière et al. (2019) and Geifman & El-Yaniv (2019) improved on the previous works suggesting to combine new architectures and new ad-hoc loss functions in the training process. In particular, the former introduced the training of an additional network to predict the confidence of a pre-trained model. Such additional network can observe the input representation produced by the pre-trained model before its decision layer. Geifman & El-Yaniv (2019) proposed to achieve selective classification by training a neural network with three heads: a classification head, and auxiliary head, and one head to estimate the confidence of the classification decision. Moreover a new loss in introduced in order to control coverage and risk at training time. Liu et al. (2019) presented another method for implementing selective classification by introducing the gambler's loss derived from general portfolio theory in the training. Such a loss is aimed at maximizing the double rate of a gambler that gambles the payoffs of previous bets. It is important to notice that these methods require data and resources to train/tune complex models.

Huang et al. (2020) proposed a method to improve the generalization empirical risk minimization of deep models, focusing on the task of learning from corrupted data and showing how calibrating the model at training time improves the models' performance for selective classification. A similar results is also reaffirmed in Fisch et al. (2022). Recently, Feng et al. (2022) have noticed that complex training techniques involving ad-hoc loss functions do not necessarily imply dramatic improvements for decision making with rejection option. They have revisited Hendrycks & Gimpel (2017), and they have proposed to further regularize popular objective functions with entropy-minimization at training time. Rabanser et al. (2022) introduced a framework that, for a given test input, monitors the disagreement with the final predicted label over intermediate models obtained during training. Although no active training is required, this frameworks need all the side information contained in the training dynamics, which are used after being discretized. Einbinder et al. (2022) show how conformal predictors can enjoy smaller conformal prediction sets with higher conditional coverage, after exact calibration with hold-out data. Gangrade et al. (2021a) studied the problem of selective classification in a game-like context where a family of selective classifiers is available. In such context, an adversary produces features and labels, but the labels are only visible to the classifiers in case of abstention creating a trade-off between numbers of abstention and classification accuracy. Lin et al. (2022); Schreuder & Chzhen (2021); Gangrade et al. (2021b) focus on the effect of selective classification on each individual class, introducing an interesting fairness related perspective in the field.

More recently, Granese et al. (2021) have introduced a new simple state-of-the-art framework for misclassification detection which builds and improves on Hendrycks & Gimpel (2017). They applied a modified version of the Rényi Entropy to obtain a score for each input sample which is then used to decide whether to accept or reject the decision relative to the sample itself. This method does not require any training and only looks at the soft-probabilities output by the model, making it very appealing w.r.t. popular but more computationally heavy methods such as Geifman & El-Yaniv (2019), Corbière et al. (2019), and Liu et al. (2019).

For completeness, we mention important theoretical results Herbei & Wegkamp (2006); Franc et al. (2021); Fischer et al. (2016), and we observe how the interesting topic of rejection option crossed the boundaries of other well-established research areas such as certified robustness (cf.Cohen et al. (2019); Tramèr (2022)) and adversarial examples detection (cf.Aldahdooh et al. (2021)). Lastly, for a comprehensive look at the topic of machine learning with reject option we reference Hendrickx et al. (2021).

3 Mathematical background

We start by introducing the mathematical definition of selective model in Sec. 3.1; we provide the justification of the rejection function we base the proposed method in Sec. 3.2 which is then introduced in Sec. 3.3.

3.1 The selective model

Assume that $\mathcal{X} \subseteq \mathbb{R}$ is the feature space and $\mathcal{Y} = \{1, \dots, C\}$, is the label space related to some relevant task. The training set $\mathcal{D}_n = \{(\mathbf{x}_i, y_i)\}_{i=1}^n \sim p_{XY}$ is a random realization of n i.i.d. samples according to p_{XY}, the underlying and unknown probability density function over $\mathcal{X} \times \mathcal{Y}$.

Throughout the paper we call *selective model* the pair $(f_{\mathcal{D}_n}, \mathrm{S})$ where $f_{\mathcal{D}_n} : \mathcal{X} \to \mathcal{Y}$ is the *predictor* (e.g. the classifier) defined as $f_{\mathcal{D}_n}(\mathbf{x}) = \arg\max_{y \in \mathcal{Y}} P_{\widehat{Y}|X}(y|\mathbf{x}; \mathcal{D}_n)$ where $P_{\widehat{Y}|X}$ is the soft-prediction of the class posterior probability given a sample; $\mathrm{S} : \mathcal{X} \to \{0, 1\}$ is the *selector* (i.e. the rejection condition). The selective model is then generally defined as:

$$(f_{\mathcal{D}_n}, \mathrm{S})(\mathbf{x}) \doteq \begin{cases} f_{\mathcal{D}_n}(\mathbf{x}) & \text{if } \mathrm{S}(\mathbf{x}) \\ \emptyset & \text{otherwise,} \end{cases} \tag{1}$$

where $\emptyset$ indicates that $f_{\mathcal{D}_n}$ abstains from the prediction. Geifman & El-Yaniv (2017) base the selector on a *confidence-rate* function. The interpretation is that given any $(\mathbf{x}_1, y_1) \sim p_{XY}$ and $(\mathbf{x}_2, y_2) \sim p_{XY}$, and denoting the *ideal* confidence-rate function as $\kappa_{f_{\mathcal{D}_n}} : \mathcal{X} \to \mathbb{R}^+$, $\kappa_{f_{\mathcal{D}_n}}(\mathbf{x}_1) \geq \kappa_{f_{\mathcal{D}_n}}(\mathbf{x}_2)$ if and only if $\ell(f_{\mathcal{D}_n}(\mathbf{x}_1), y_1) \leq \ell(f_{\mathcal{D}_n}(\mathbf{x}_2), y_2)$, where $\ell : Y \times Y \to \mathbb{R}^+$ is a given loss function (e.g., the 0-1 loss).

3.2 Gini Impurity Score based Selector

We recall that, the *probability of classification error* for a given $\mathbf{x} \in \mathcal{X}$ w.r.t. the true posterior probability $P_{Y|X}$ is defined as $\text{Pe}(\mathbf{x}) \doteq 1 - P_{Y|X}(f_{\mathcal{D}_n}(\mathbf{x})|\mathbf{x})$. Normally, we do not know $P_{Y|X}$, and all we can observe are samples drawn from the joint distribution p_{XY} from which we can learn an empirical soft-distribution $P_{\widehat{Y}|X}$. Interestingly, it is possible to derive a bound to the unknown function $\mathbf{x} \mapsto \text{Pe}(\mathbf{x})$ through the *Gini Impurity Score* (Gini, 1912; Granese et al., 2021):

$$\sqrt{\text{Gini}(\mathbf{x})} - \Delta(\mathbf{x}) \leq \text{Pe}(\mathbf{x}) \leq \text{Gini}(\mathbf{x}) + \Delta(\mathbf{x}),$$

where

$$\Delta(\mathbf{x}) \doteq 2\sqrt{2\, D_{\text{KL}}\left(P_{Y|X}(\cdot|\mathbf{x})||(P_{\widehat{Y}|X}(\cdot|\mathbf{x})\right)},$$

$$\text{Gini}(\mathbf{x}) \doteq \sum_{y\in\mathcal{Y}} P_{\widehat{Y}|X}(y|\mathbf{x}) \Pr(\widehat{Y} \neq y|\mathbf{x}) = 1 - \sum_{y\in\mathcal{Y}} P^2_{\widehat{Y}|X}(y|\mathbf{x}), \tag{2}$$

and $D_{\text{KL}}(\cdot||\cdot)$ denotes the Kullback-Leibler divergence between two distributions. The Gini Impurity represents the probability that the input sample would be classified incorrectly if randomly labeled according to the distribution $P_{\widehat{Y}|X}$.

Interestingly, Eq. (2) can be linked to a much popular information theoretic measure, i.e. the Rényi divergence, which in our scenario, is defined as:

$$D_\alpha\left(P_{\widehat{Y}|X}(\cdot,\mathbf{x})||Q_Y\right) \doteq \frac{1}{\alpha-1}\log\left(\sum_{y\in\mathcal{Y}}\left(P^\alpha_{\widehat{Y}|X}(y|\mathbf{x})Q_Y^{(1-\alpha)}(y)\right)\right), \tag{3}$$

where $P_{\widehat{Y}|X}(\cdot,\mathbf{x})$ is the model soft-distribution for a fixed input sample $\mathbf{x}$, and Q_Y is a distribution over the labels set $\mathcal{Y}$. Indeed, by fixing $\alpha = 2$, and Q_Y as a uniform over the labels, we can derive a strong connection to $\text{Gini}(\mathbf{x})$. We reference Appendix A.1 for a more in-depth analysis.

3.3 Our proposed selective model

We leverage the results presented in Section 3.2 to build our Gini-based selective model.

Definition 3.1 (Implementing the rejection option with Gini$(\cdot)$)

$$(f_{\mathcal{D}_n}, Gini, \gamma)(\mathbf{x}) \doteq \begin{cases} f_{\mathcal{D}_n}(\mathbf{x}) & \textit{if Gini}(\mathbf{x}) \leq \gamma \\ \emptyset & \textit{if Gini}(\mathbf{x}) > \gamma, \end{cases} \tag{4}$$

where $\gamma \in [0,1]$ is the threshold parameter and $\emptyset$ indicates that $f_{\mathcal{D}_n}$ abstains from the prediction.

Clearly, in our scenario, and w.r.t. Eq. (1), it holds true that

$$\text{S}(\mathbf{x}) = \text{Gini}(\mathbf{x}) \leq \gamma.$$

Further details on the calibration of the parameter γ can be found in Section 4.4.

4 Experimental Setting

This section describes the experiments we run and the evaluation settings. Specifically, in Sec. 4.1 we begin by listing the datasets and the classifiers involved in our evaluation; in Sec. 4.2 we move on with the explanation of the *risk* and *coverage* metrics; in Sec. 4.3 we give more details on the methods to which we compare, and we conclude with the analysis of the coverage calibration in Sec. 4.4.

4.1 Datasets and classifiers

We run our experiments on CIFAR-10 (Krizhevsky, 2009), CIFAR-100 (Krizhevsky, 2009) and SVHN (Netzer et al., 2011) image datasets. For all of them, we have considered as classifiers

VGG-16 (Simonyan & Zisserman, 2015), ResNet-34 (He et al., 2016) and DenseNet-121 (Huang et al., 2017) architectures. They have been trained with the cross entropy loss for 300 epochs, using as optimizer the stochastic gradient descent (SGD) with a learning rate of 0.1, a cosine annealing learning rate scheduler, weight decay of 0.0005, and momentum of 0.9. The accuracy achieved by the classifiers on the original testing data for VGG-16 are $94.01 \pm 0.14\%$ on CIFAR-10, $74.75 \pm 0.31\%$ on CIFAR-100, and $95.67 \pm 0.12\%$ on SVHN; for ResNet-34 are $95.62 \pm 0.13\%$ on CIFAR-10, $79.41 \pm 0.43\%$ on CIFAR-100, and $96.14 \pm 0.1\%$ on SVHN; and for DenseNet-121 are $94.10 \pm 0.23\%$ on CIFAR-10, $74.02 \pm 0.27\%$ on CIFAR-100, and $95.80 \pm 0.22\%$ on SVHN.

4.2 Evaluation metrics

We measure the performances of the selective model post *coverage calibration* in terms of empirical coverage (Geifman & El-Yaniv, 2017; 2019) (the higher the better):

$$\hat{\phi}(\mathrm{S};\mathcal{D}_m) \doteq \frac{1}{m}\sum_{i=1}^{m} \mathbb{1}_{[\mathrm{S}(\mathbf{x}_i)]}, \tag{5}$$

and in terms of empirical selective risk (Geifman & El-Yaniv, 2017; 2019) (the lower the better):

$$\hat{r}\left(f_{\mathcal{D}_n},\mathrm{S};\mathcal{D}_m\right) \doteq \frac{\sum_{i=1}^{m} \mathbb{1}_{[f_{\mathcal{D}_n}(\mathbf{x}_i)\neq y_i]} \cdot \mathbb{1}_{[\mathrm{S}(\mathbf{x}_i)]}}{\sum_{i=1}^{m} \mathbb{1}_{[\mathrm{S}(\mathbf{x}_i)]}}, \tag{6}$$

where $\mathcal{D}_m = \{(\mathbf{x}_i, y_i)\}_{i=1}^{m}$ is the test or evaluation set and $\mathbb{1}_{[\cdot]}$ denote the indicator function. In particular, the effectiveness of a selective model are expressed by drawing the *risk-coverage curve* (El-Yaniv & Wiener, 2010) of its induced rejection function.

Note that, the empirical selective risk value is usually multiplied by 100.

4.3 Benchmark details

The method we propose relies on a state-of-the-art misclassification detection method to implement the rejection option. As a consequence it can be used jointly with many popular out of the shelf pre-trained models which are available nowadays. Compared to SelectiveNet (cf. Geifman & El-Yaniv (2019)), ConfidNet (cf. Corbière et al. (2019)), and DeepGambler (cf. Liu et al. (2019)), our method provides a solution to the selective classification problem that is less demanding in terms of resources and that requires less parameters to be tuned. This is a clear advantage, especially for scenarios in which it is hard to collect extra samples for parameter optimization.

For the three datasets, we used the entire training set for training the methods. We follow the corresponding training recipe for each method as proposed in the original works[1]. For SelectiveNet, a model is fit for every target coverege, and the practitioner should implement a custom loss that might not be so easy to optimize depending on the task in hand. For ConfidNet, the auxiliary confidence network adds a considerable overhead, specially for inference. The proposed architecture has 1M extra parameters, which is already bigger that a DenseNet-121 model, which could limit some applications, e.g., ML applications on the edge Murshed et al. (2022). For DeepGamblers, there is a considerable amount of optimization needed on top of the basic cross entropy training, which would require an extensive search of hyperparameters. In contrast, we propose a selector with zeros hyperparameters and zero extra training or architectural changes.

From the methodology point of view, for every model, dataset, and method, we ran experiments with 5 different random seeds and we report error bars for all of our results. Also, the methods share the same backbone architecture for the neural network, except for VGG in SelectiveNet, where we added dropout layers at each block as in Geifman & El-Yaniv (2019). Compared to previous works, we propose two extra neural network architectures, the ResNet and DenseNet, and a more challenging benchmark with CIFAR-100. We uniformed the procedure of coverage calibration across methods, which is unclear in some of the previous works.

[1] SelectiveNet: `https://github.com/geifmany/selectivenet`; ConfidNet: `https://github.com/valeoai/ConfidNet`; DeepGamblers: `https://github.com/Z-T-WANG/NIPS2019DeepGamblers`.

4.4 Coverage Calibration Details

For the three datasets, we partition the test dataset into two sets, one for coverage calibration only and another for evaluation only. We refer to the first as calibration set and the second one as test set. The calibration set corresponds to 10% of the original partition, note that we select the data randomly for each of the five seeds. The coverage calibration algorithm is given in Algorithm 1 where $\mathcal{D}_{m'}$ denotes the calibration dataset of size m'. Following previous works, we set as target coverages $\tau \in \{0.70, 0.75, 0.80, 0.85, 0.90, 0.95, 1.00\}$. Intuitively, in order to guarantee the target coverage, we compute the Gini Impurity Score for all the samples in the calibration set and we order these values in ascending order. Then, we select as final threshold the score value at position $\lceil \tau \cdot m' \rceil$ as this guarantee that at least $\lceil \tau \cdot m' \rceil$ samples will be classified.

We calibrated all of the methods with this same procedure to achieve fair and comparable results.

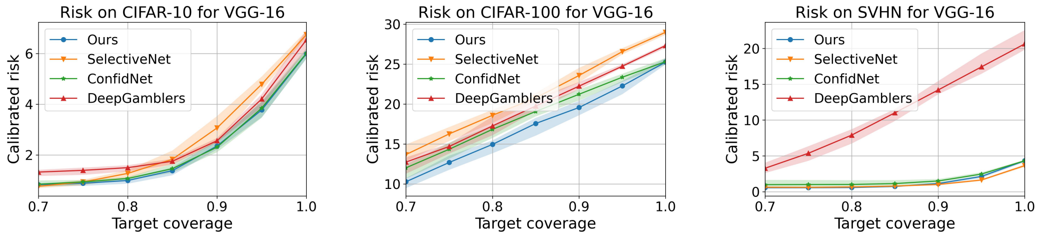
Algorithm 1: Coverage calibration algorithm for our proposed method

```
Data: $\mathcal{D}_{m'} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{m'}$, with $m' < m$ and $m' < n$; target coverage $\tau \in [0, 1]$
Result: $\gamma^\star$ threshold value that guarantees the target coverage on $\mathcal{D}_{m'}$

SList ← [ ]                                   ▷ Initialize an empty list of Scores
for $i \leftarrow 1$ to $m'$ do
    SList.append(Gini($\mathbf{x}_i$))
sort(SList, ascend=True)                      ▷ Sort SList in ascending order
return $\gamma^* =$ SList[$\lceil \tau \cdot m' \rceil$]
```

5 Main Results and Discussion

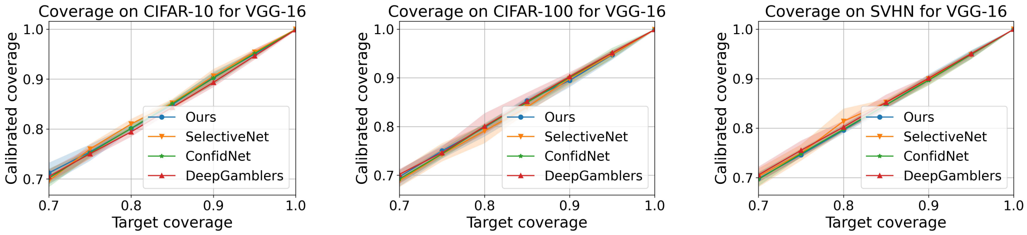
We discuss the empirical results obtained with our methods, and we compare them to the results obtained with SelectiveNet, ConfidNet, and DeepGambler. Tab. 1 and Tab. 2 report the achieved calibrated risk and calibrated coverage respectively, on CIFAR-10, across the different architectures which we included in our evaluation. Fig. 1 and Fig. 2 report the performance achieved by the VGG-16 architectures over all the considered benchmark datasets for each considered method. Extended results for CIFAR-100 and SVHN are available in Appendix A.2, reported in Tab. 4 and Tab. 5 respectively. Moreover, additional plots are reported in Figs. 4 and 5, and Figs. 6 and 7, for DenseNet and ResNet respectively.



(a) Calibrated risk on CIFAR-10. (b) Calibrated risk on CIFAR-100. (c) Calibrated risk on SVHN.

Figure 1: Calibrated risk versus target coverage for the VGG-16 model. Our post-training method has superior performance on CIFAR-100 and comparable performance on CIFAR-10 and SVHN.

Crucially, our simple method, whose performance is consistently comparable to that of its competitors, is able to outperform them in more challenging task such as classification on CIFAR-100. For the sake of fairness, it is important to notice that this benchmark was not considered in Liu et al. (2019) and in Geifman & El-Yaniv (2019). As a consequence, the optimized training procedures for CIFAR-100 are not published alongside the aforementioned papers, and we resorted to using the same training settings released for CIFAR-10. Clearly, it can be claimed that this would lead to



(a) Calibrated coverage on CIFAR-10.

(b) Calibrated coverage on CIFAR-100.

(c) Calibrated risk on SVHN

Figure 2: Calibrated coverage for the VGG-16 model, highlighting that every method tested can be calibrated with a few extra data post-training training.

Table 1: Empirical selective risk in percentage for the classification benchmark with three models for various target coverages on the CIFAR-10 benchmark. Bold font indicates best result for the line.

	Target Coverage	Calibrated Risk			
		SelectiveNet	ConfidNet	DeepGamblers	Ours
VGG-16 (CIFAR-10)	1.00	6.74±0.11	6.01±0.15	6.54±0.29	**5.99±0.14**
	0.95	4.78±0.22	3.85±0.25	4.20±0.20	**3.77±0.27**
	0.90	3.06±0.37	**2.31±0.17**	2.56±0.08	2.35±0.14
	0.85	1.83±0.22	1.46±0.14	1.75±0.09	**1.38±0.12**
	0.80	1.27±0.13	1.07±0.05	1.50±0.09	**0.99±0.09**
	0.75	0.94±0.09	0.94±0.07	1.39±0.11	**0.88±0.07**
	0.70	**0.76±0.06**	0.84±0.13	1.32±0.09	0.83±0.06
ResNet-34 (CIFAR-10)	1.00	7.42±0.36	4.37±0.15	4.79±0.10	**4.38±0.13**
	0.95	5.21±0.51	**2.24±0.22**	2.66±0.20	**2.24±0.25**
	0.90	3.69±0.29	1.28±0.09	1.58±0.09	**1.26±0.11**
	0.85	2.17±0.24	0.82±0.09	0.99±0.10	**0.79±0.11**
	0.80	1.37±0.26	0.62±0.07	0.76±0.18	**0.57±0.09**
	0.75	0.87±0.22	0.5±0.06	0.67±0.18	**0.48±0.06**
	0.70	0.68±0.07	0.46±0.06	0.62±0.17	**0.39±0.06**
DenseNet-121 (CIFAR-10)	1.00	7.12±0.26	5.92±0.19	**5.74±0.15**	5.90±0.23
	0.95	5.10±0.46	3.71±0.29	**3.67±0.24**	3.72±0.27
	0.90	3.41±0.38	2.42±0.24	**2.24±0.17**	2.43±0.24
	0.85	2.27±0.17	1.51±0.12	1.58±0.23	**1.50±0.14**
	0.80	1.67±0.25	1.01±0.13	1.38±0.25	**0.98±0.09**
	0.75	1.28±0.14	0.75±0.15	1.24±0.19	**0.72±0.17**
	0.70	1.11±0.18	0.59±0.13	1.12±0.19	**0.55±0.10**

sub-optimal results, and we believe that further optimization would improve the competitors' performance. **However, the main takeaway is that, our simple method, can achieve high performance with off the shelf models requiring no further optimization. This means that our framework is extensible to any new classification task without adding any optimization or hyper-parameters search burden on top of the base model's training.** This is clearly a considerable advantage for practitioners willing to integrate a rejection option to their classification systems.

Tab. 2 and Fig. 2 show that all the methods are consistently good in terms of achieved empirical coverage after the coverage calibration on the held out samples. The main takeaway is therefore the fact that our proposed method which does not enjoy any ad-hoc training and/or tuning achieves the same performance as the competitors. Coverage calibration is important, because in critical applications we want to guarantee that the target coverage will be achievable in practice. Our procedure should satisfy this condition as long as there is not a large drift between the test and the calibration data distributions.

Table 2: Experiment with the VGG16 (CIFAR-10) model. The same behavior is observed for all the models and datasets studied in this work.

Target Coverage	Calibrated Coverage			
	SelectiveNet	ConfidNet	DeepGamblers	Ours
1.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00
0.95	0.95±0.01	0.95±0.01	0.95±0.01	0.95±0.00
0.90	0.91±0.01	0.90±0.01	0.89±0.01	0.90±0.01
0.85	0.85±0.01	0.85±0.01	0.84±0.01	0.85±0.01
0.80	0.81±0.01	0.80±0.01	0.79±0.01	0.80±0.01
0.75	0.76±0.01	0.75±0.01	0.75±0.01	0.75±0.01
0.70	0.70±0.01	0.70±0.01	0.71±0.01	0.71±0.02

5.1 Results on ImageNet

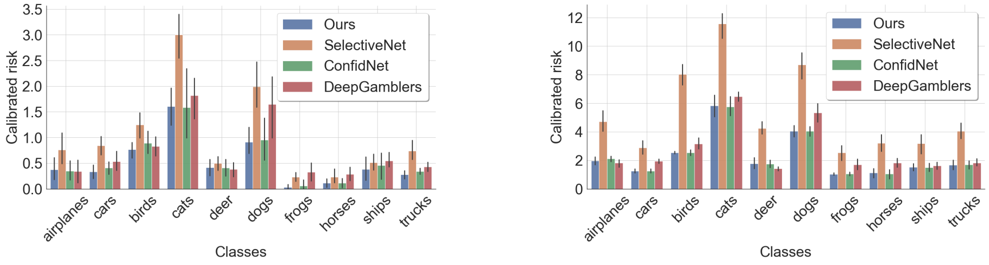
In order to study how accuracy impacts the risk of a rejection option on a large scale problem, we draw the following experiment. We took five off-the-shelf pre-trained ResNet models with different number of parameters and increasing accuracy (ResNet-18, ResNet-34, ResNet-50, ResNet-101, and ResNet-152) from Paszke et al. (2019). We evaluated the performance of our Gini selector on the ILSVRC2012, or ImageNet-1K dataset (Deng et al., 2009) validation partition for each of the target coverages. We used 10% of this partition to coverage calibration and 90% to evaluation purposes. Tab. 3 shows that the empirical risk our method decreases with the accuracy of the base model on the same task. Of course this has the cost of increasing the number of parameters. These results reassures a very deterministic and consistent performance for our Gini selector, which scales perfectly well to tasks of any size.

Table 3: Calibrated risk for our Gini selector for residual deep neural networks of increasing accuracy on the ImageNet dataset.

Target Coverage	Calibrated Risk				
	ResNet-18	ResNet-34	ResNet-50	ResNet-101	ResNet-152
1.00	30.09	26.58	19.57	18.19	17.56
0.95	27.35	23.80	17.32	15.53	14.90
0.90	24.81	21.37	15.34	13.35	12.70
0.85	22.35	18.47	13.69	11.78	10.84
0.80	20.02	16.02	12.27	9.79	9.03
0.75	17.77	14.07	11.10	8.08	7.38
0.70	15.18	12.02	10.05	6.52	6.08
Accuracy	69.76	73.30	80.35	81.67	82.35

5.2 Per-class analysis of the calibrated risk

In Fig. 3 we report the per-class calibrated risk results of the proposed method together with those of competitors on CIFAR10 and ResNet-34. The fixed target coverage is $\tau = 0.75$ in Fig. 3a and $\tau = 0.95$ in Fig. 3b. As can be seen, in both figures, the distribution of calibrated risk is not uniform among the classes, regardless of the method used. The classes 'cats' and 'dogs' are the most exposed to risk, while 'frogs' and 'horses' are the least exposed. For all classes, the proposed method outperforms its competitors (or performs equally well). SelectiveNet, on the other hand, stands out as the worst method. The best performance between ConfidNet and DeepGamblers depends on the class chosen. However, the gap between the results of the two methods on the classes 'cats' and 'dogs' suggests ConfidNet as the more stable of the two.



(a) $\tau = 0.75$

(b) $\tau = 0.95$

Figure 3: Per-class analysis of the calibrated risk on CIFAR10 and ResNet-34.

5.3 Limitations of this work

All the methods empirically evaluated in this work have been validated using an held out calibration set. Our method makes no difference and it relies on this extra data as well to fix a threshold (the γ in Eq. (4)) to be used at evaluation time. At validation time, we select the threshold which most closely achieves the target coverage over the held out data, and then we test the performance in terms of risk using the selected threshold. Lack of extra samples to implement this validation step clearly represents a limitation for our proposed method.

Granese et al. (2021) presents a version of their framework which includes scaling of the logits and input perturbation. Although we did not try these techniques in this work[2], the fact that $\text{Gini}(\cdot)$ relies on the self-confidence of a model means that the extreme case in which the considered model is equally confident when it correctly classifies a sample and when it incorrectly classifies a sample, represents yet another limitation. In this case further calibration techniques could be implemented to avoid such an unfavorable scenario (cf. Guo et al. (2017)).

At this time, the proposed selective model can only be applied to classification tasks since the selector is based on soft-probability. We leave as future work its extension to other tasks.

6 Final remarks

We have presented a new method that can be used to implement the rejection option for pre-trained models, based on a state-of-the-art misclassification detection method. We have empirically shown that the simple and inexpensive method we propose achieves comparable, and sometimes better performance w.r.t. more complex methods for selective classification when tested on popular datasets and popular deep learning models.

The appeal of our proposed solution rests in the fact that it does not require further training or expensive tuning of the models and it can be used with any standard model, provided that its output can be interpreted as a soft-distribution. Although ad-hoc training and loss functions seem like a fairly smart way to implement rejection options, they come at the cost of tuning more hyperparameters and adding more parameters to popular neural network models. We are convinced that these methods are powerful and deserve further analysis. At the same time we claim that for many popular classification models and benchmark dataset our simple solution provides good performance while enjoying a much lighter framework.

That said, we would like to invite the community to reflect on the question: should we use additional data and computational resource to make the soft-distribution output by the model more reliable before resorting to more complex training procedures, at least for many popular applications? Techniques such as calibration (cf. Guo et al. (2017)) are particularly useful to correct the overconfidence of modern deep learning models and can be used to improve the quality of the soft distribution outputs, which in turn are used for post-training methods like the one proposed in this paper.

[2] One of the paper's goals is to show that the proposed method, even if much less complex than the competitors, is not inferior. Since the performances are already comparable and, in most cases, superior, we leave further fine-tuning as future work.

REFERENCES

Ahmed Aldahdooh, Wassim Hamidouche, and Olivier Déforges. Revisiting model's uncertainty and confidences for adversarial example detection, 2021. URL https://arxiv.org/abs/2103.05354.

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.

Melanie Bernhardt, Fabio De Sousa Ribeiro, and Ben Glocker. Failure detection in medical image classification: A reality check and benchmarking testbed. *CoRR*, abs/2205.14094, 2022. doi: 10.48550/arXiv.2205.14094. URL https://doi.org/10.48550/arXiv.2205.14094.

Yunus Bicer, Ali Alizadeh, Nazim Kemal Ure, Ahmetcan Erdogan, and Orkun Kizilirmak. Sample efficient interactive end-to-end deep learning for self-driving cars with selective multi-class safe dataset aggregation. *CoRR*, abs/2007.14671, 2020. URL https://arxiv.org/abs/2007.14671.

Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom B. Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. Extracting training data from large language models. In Michael Bailey and Rachel Greenstadt (eds.), *30th USENIX Security Symposium, USENIX Security 2021, August 11-13, 2021*, pp. 2633–2650. USENIX Association, 2021. URL https://www.usenix.org/conference/usenixsecurity21/presentation/carlini-extracting.

C. K. Chow. An optimum character recognition system using decision functions. *IRE Transactions on Electronic Computers*, EC-6(4):247–254, 1957. doi: 10.1109/TEC.1957.5222035.

C. K. Chow. On optimum recognition error and reject tradeoff. *IEEE Trans. Inf. Theory*, 16(1): 41–46, 1970. doi: 10.1109/TIT.1970.1054406. URL https://doi.org/10.1109/TIT.1970.1054406.

Oliver Cobb and Arnaud Van Looveren. Context-aware drift detection. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 4087–4111. PMLR, 2022. URL https://proceedings.mlr.press/v162/cobb22a.html.

Jeremy M. Cohen, Elan Rosenfeld, and J. Zico Kolter. Certified adversarial robustness via randomized smoothing. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 1310–1320. PMLR, 2019. URL http://proceedings.mlr.press/v97/cohen19c.html.

Charles Corbière, Nicolas Thome, Avner Bar-Hen, Matthieu Cord, and Patrick Pérez. Addressing failure prediction by learning model confidence. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 2898–2909, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/757f843a169cc678064d9530d12a1881-Abstract.html.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Bat-Sheva Einbinder, Yaniv Romano, Matteo Sesia, and Yanfei Zhou. Training uncertainty-aware classifiers with conformalized deep learning. *CoRR*, abs/2205.05878, 2022. doi: 10.48550/arXiv.2205.05878. URL https://doi.org/10.48550/arXiv.2205.05878.

Ran El-Yaniv and Yair Wiener. On the foundations of noise-free selective classification. *J. Mach. Learn. Res.*, 11:1605–1641, 2010. doi: 10.5555/1756006.1859904. URL https://dl.acm.org/doi/10.5555/1756006.1859904.

Leo Feng, Mohamed Osama Ahmed, Hossein Hajimirsadeghi, and Amir H. Abdi. Stop overcomplicating selective classification: Use max-logit. *CoRR*, abs/2206.09034, 2022. doi: 10.48550/arXiv.2206.09034. URL https://doi.org/10.48550/arXiv.2206.09034.

Adam Fisch, Tommi Jaakkola, and Regina Barzilay. Calibrated selective classification. *CoRR*, abs/2208.12084, 2022. doi: 10.48550/arXiv.2208.12084. URL https://doi.org/10.48550/arXiv.2208.12084.

Lydia Fischer, Barbara Hammer, and Heiko Wersing. Optimal local rejection for classifiers. *Neurocomputing*, 214:445–457, 2016. doi: 10.1016/j.neucom.2016.06.038. URL https://doi.org/10.1016/j.neucom.2016.06.038.

Ivan Flores. An optimum character recognition system using decision functions. *IRE Trans. Electron. Comput.*, 7(2):180, 1958. doi: 10.1109/TEC.1958.5222530. URL https://doi.org/10.1109/TEC.1958.5222530.

Vojtech Franc, Daniel Průša, and V. Voracek. Optimal strategies for reject option classifiers. *CoRR*, abs/2101.12523, 2021. URL https://arxiv.org/abs/2101.12523.

Keinosuke Fukunaga and David L. Kessell. Application of optimum error-reject functions (corresp.). *IEEE Trans. Inf. Theory*, 18(6):814–817, 1972. doi: 10.1109/TIT.1972.1054919. URL https://doi.org/10.1109/TIT.1972.1054919.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In Maria-Florina Balcan and Kilian Q. Weinberger (eds.), *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pp. 1050–1059. JMLR.org, 2016. URL http://proceedings.mlr.press/v48/gal16.html.

Aditya Gangrade, Anil Kag, Ashok Cutkosky, and Venkatesh Saligrama. Online selective classification with limited feedback. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 14529–14541, 2021a. URL https://proceedings.neurips.cc/paper/2021/hash/79b6245ff93841eb8c120cec9bf8be14-Abstract.html.

Aditya Gangrade, Anil Kag, and Venkatesh Saligrama. Selective classification via one-sided prediction. In Arindam Banerjee and Kenji Fukumizu (eds.), *The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, April 13-15, 2021, Virtual Event*, volume 130 of *Proceedings of Machine Learning Research*, pp. 2179–2187. PMLR, 2021b. URL http://proceedings.mlr.press/v130/gangrade21a.html.

Yue Gao, Ilia Shumailov, and Kassem Fawaz. Rethinking image-scaling attacks: The interplay between vulnerabilities in machine learning systems. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 7102–7121. PMLR, 2022. URL https://proceedings.mlr.press/v162/gao22g.html.

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 4878–4887, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/4a8423d5e91fda00bb7e46540e2b0cf1-Abstract.html.

Yonatan Geifman and Ran El-Yaniv. Selectivenet: A deep neural network with an integrated reject option. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2151–2159. PMLR, 2019. URL http://proceedings.mlr.press/v97/geifman19a.html.

Corrado Gini. Variabilità e mutabilità; contributo allo studio delle distribuzioni e delle relazioni statistiche. In *[Fasc. I.]. Tipogr. Di P. Cuppini 1912.*, 1912.

Federica Granese, Marco Romanelli, Daniele Gorla, Catuscia Palamidessi, and Pablo Piantanida. DOCTOR: A simple method for detecting misclassification errors. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 5669–5681, 2021. URL `https://proceedings.neurips.cc/paper/2021/hash/2cb6b10338a7fc4117a80da24b582060-Abstract.html`.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1321–1330. PMLR, 2017. URL `http://proceedings.mlr.press/v70/guo17a.html`.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 770–778. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.90. URL `https://doi.org/10.1109/CVPR.2016.90`.

Martin E. Hellman. The nearest neighbor classification rule with a reject option. *IEEE Trans. Syst. Sci. Cybern.*, 6(3):179–185, 1970. doi: 10.1109/TSSC.1970.300339. URL `https://doi.org/10.1109/TSSC.1970.300339`.

Kilian Hendrickx, Lorenzo Perini, Dries Van der Plas, Wannes Meert, and Jesse Davis. Machine learning with a reject option: A survey. *CoRR*, abs/2107.11277, 2021. URL `https://arxiv.org/abs/2107.11277`.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL `https://openreview.net/forum?id=Hkg4TI9xl`.

Radu Herbei and Marten H. Wegkamp. Classification with reject option. *The Canadian Journal of Statistics / La Revue Canadienne de Statistique*, 34(4):709–721, 2006. ISSN 03195724. URL `http://www.jstor.org/stable/20445230`.

Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pp. 2261–2269. IEEE Computer Society, 2017. doi: 10.1109/CVPR.2017.243. URL `https://doi.org/10.1109/CVPR.2017.243`.

Lang Huang, Chao Zhang, and Hongyang Zhang. Self-adaptive training: beyond empirical risk minimization. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL `https://proceedings.neurips.cc/paper/2020/hash/e0ab531ec312161511493b002f9be2ee-Abstract.html`.

Heinrich Jiang, Been Kim, Melody Y. Guan, and Maya R. Gupta. To trust or not to trust A classifier. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 5546–5557, 2018. URL `https://proceedings.neurips.cc/paper/2018/hash/7180cffd6a8e829dacfc2a31b3f72ece-Abstract.html`.

Di Jin, Shuyang Gao, Seokhwan Kim, Yang Liu, and Dilek Hakkani-Tür. Towards textual out-of-domain detection without in-domain labels. *IEEE ACM Trans. Audio Speech Lang. Process.*, 30:1386–1395, 2022. doi: 10.1109/TASLP.2022.3162081. URL `https://doi.org/10.1109/TASLP.2022.3162081`.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

Zhen Lin, Lucas Glass, M. Brandon Westover, Cao Xiao, and Jimeng Sun. SCRIB: set-classifier with class-specific risk bounds for blackbox models. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pp. 7497–7505. AAAI Press, 2022. URL https://ojs.aaai.org/index.php/AAAI/article/view/20714.

Ziyin Liu, Zhikang Wang, Paul Pu Liang, Ruslan Salakhutdinov, Louis-Philippe Morency, and Masahito Ueda. Deep gamblers: Learning to abstain with portfolio theory. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 10622–10632, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/0c4b1eeb45c90b52bfb9d07943d855ab-Abstract.html.

M. G. Sarwar Murshed, Christopher Murphy, Daqing Hou, Nazar Khan, Ganesh Ananthanarayanan, and Faraz Hussain. Machine learning at the network edge: A survey. *ACM Computing Surveys*, 54(8):1–37, nov 2022. doi: 10.1145/3469029. URL https://doi.org/10.1145%2F3469029.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019.

Marco AF Pimentel, David A Clifton, Lei Clifton, and Lionel Tarassenko. A review of novelty detection. *Signal processing*, 99:215–249, 2014.

Pavel Pudil, Jana Novovicová, Svatopluk Bláha, and Josef Kittler. Multistage pattern recognition with reject option. In *11th IAPR International Conference on Pattern Recognition, ICPR 1992. Conference B: Pattern Recognition Methodology and Systems, The Hague, Netherlands, August 30-September 3, 1992*, pp. 92–95. IEEE, 1992. doi: 10.1109/ICPR.1992.201729. URL https://doi.org/10.1109/ICPR.1992.201729.

Stephan Rabanser, Anvith Thudi, Kimia Hamidieh, Adam Dziedzic, and Nicolas Papernot. Selective classification via neural network training dynamics. *CoRR*, abs/2205.13532, 2022. doi: 10.48550/arXiv.2205.13532. URL https://doi.org/10.48550/arXiv.2205.13532.

Nicolas Schreuder and Evgenii Chzhen. Classification with abstention but without disparities. In Cassio P. de Campos, Marloes H. Maathuis, and Erik Quaeghebeur (eds.), *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence, UAI 2021, Virtual Event, 27-30 July 2021*, volume 161 of *Proceedings of Machine Learning Research*, pp. 1227–1236. AUAI Press, 2021. URL https://proceedings.mlr.press/v161/schreuder21a.html.

Sambu Seo, Marko Wallat, Thore Graepel, and Klaus Obermayer. Gaussian process regression: Active data selection and test point rejection. In *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks, IJCNN 2000, Neural Computing: New Challenges and Perspectives for the New Millennium, Como, Italy, July 24-27, 2000, Volume 3*, pp. 241–246. IEEE Computer Society, 2000. doi: 10.1109/IJCNN.2000.861310. URL https://doi.org/10.1109/IJCNN.2000.861310.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1409.1556.

Adarsh Subbaswamy and Suchi Saria. From development to deployment: dataset shift, causality, and shift-stable models in health ai. *Biostatistics*, 21(2):345–352, April 2020. ISSN 1465-4644. doi: 10.1093/biostatistics/kxz041.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In Yoshua Bengio and Yann LeCun (eds.), *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6199.

Florian Tramèr. Detecting adversarial examples is (nearly) as hard as classifying them. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 21692–21702. PMLR, 2022. URL https://proceedings.mlr.press/v162/tramer22a.html.

Aditya Vailaya and Anil K. Jain. Reject option for vq-based bayesian classification. In *15th International Conference on Pattern Recognition, ICPR'00, Barcelona, Spain, September 3-8, 2000*, pp. 2048–2051. IEEE Computer Society, 2000. doi: 10.1109/ICPR.2000.906016. URL https://doi.org/10.1109/ICPR.2000.906016.

Germano C. Vasconcelos, Michael C. Fairhurst, and David L. Bisset. Investigating feedforward neural networks with respect to the rejection of spurious patterns. *Pattern Recognit. Lett*, 16(2): 207–212, 1995.

A Appendix

A.1 On the derivation of Gini$(\cdot)$ from the Rényi divergence

Let us define the Rényi divergence, as:

$$D_\alpha \left(P_{\widehat{Y}|X}(\cdot,\mathbf{x})||Q_Y\right) \doteq \frac{1}{\alpha-1}\log\left(\sum_{y\in\mathcal{Y}}\left(P^{\alpha}_{\widehat{Y}|X}(y|\mathbf{x})Q_Y^{(1-\alpha)}(y)\right)\right), \tag{7}$$

where $P_{\widehat{Y}|X}(\cdot,\mathbf{x})$ is the model soft-distribution for a fixed input sample $\mathbf{x}$, and Q_Y is a distribution over the labels set $\mathcal{Y}$. By fixing $\alpha = 2$, Eq. (7) becomes

$$D_2 \left(P_{\widehat{Y}|X}(\cdot,\mathbf{x})||Q_Y\right) = \log\left(\sum_{y\in\mathcal{Y}}\left(\frac{P^{2}_{\widehat{Y}|X}(y|\mathbf{x})}{Q_Y(y)}\right)\right). \tag{8}$$

Let us now take a closer look at the argument of the logarithm. Let us fix the reference distribution Q_Y as a uniform distribution over the classes, i.e. $Q_Y = q,\ \forall y \in \mathcal{Y}$. Then, the argument of the logarithm writes

$$\frac{1}{q}\sum_{y\in\mathcal{Y}}\left(P^{2}_{\widehat{Y}|X}(y|\mathbf{x})\right), \tag{9}$$

which corresponds to $1 - \text{Gini}(\mathbf{x})$ multiplied by a constant.

Table 4: Empirical selective risk in percentage for the classification benchmark with three models for various target coverages on the CIFAR-100 benchmark. Bold font indicates best result for the line.

	Target Coverages	Calibrated Risk			
		SelectiveNet	ConfidNet	DeepGamblers	Ours
VGG-16 (CIFAR-100)	1.00	28.98±0.23	25.3±0.29	27.31±0.18	**25.25±0.31**
	0.95	26.55±0.28	23.35±0.47	24.73±0.16	**22.26±0.72**
	0.90	23.57±0.74	21.22±0.55	22.24±0.35	**19.56±0.75**
	0.85	20.77±0.43	19.07±0.54	19.75±0.59	**17.56±0.89**
	0.80	18.58±0.65	16.83±0.63	17.26±0.85	**14.94±0.65**
	0.75	16.26±0.70	14.34±0.75	14.69±0.51	**12.68±0.63**
	0.70	13.67±0.88	12.01±0.76	12.73±0.84	**10.27±0.47**
ResNet-34 (CIFAR-100)	1.00	30.82±0.40	20.68±0.38	23.72±0.83	**20.59±0.43**
	0.95	28.37±1.02	18.51±0.51	21.27±0.83	**17.84±0.45**
	0.90	27.45±0.91	16.63±0.58	18.59±1.04	**15.19±0.36**
	0.85	25.52±0.68	14.45±0.39	16.29±0.65	**13.05±0.47**
	0.80	22.99±1.76	12.57±0.36	14.20±0.64	**10.78±0.30**
	0.75	21.59±1.12	10.58±0.46	12.18±0.61	**8.77±0.36**
	0.70	18.86±1.11	8.94±0.62	10.45±0.65	**6.91±0.27**
DenseNet-121 (CIFAR-100)	1.00	30.63±0.57	25.92±0.26	**25.65±0.49**	25.98±0.27
	0.95	28.31±0.57	23.36±0.49	23.27±0.19	**23.10±0.52**
	0.90	25.76±0.21	20.90±0.68	21.21±0.19	**20.79±0.42**
	0.85	23.57±0.75	18.51±0.68	19.06±0.19	**18.26±0.59**
	0.80	22.06±0.32	15.95±0.70	17.05±0.44	**15.62±0.74**
	0.75	19.42±0.82	13.66±0.69	15.35±0.50	**13.24±0.54**
	0.70	17.20±0.88	11.33±0.52	13.60±0.31	**11.01±0.50**

A.2 Extended results

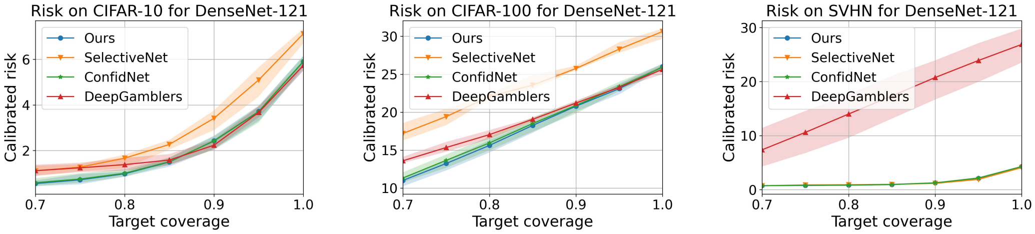
In this section we report extended results which complement the analysis of Tab. 1 and Tab. 2. In particular, Tab. 4 and Tab. 5 report the achieved results in terms of calibrated risk when DenseNet-121 and ResNet-34 are considered.

Table 5: Empirical selective risk in percentage for the classification benchmark with three models for various target coverages on the SVHN benchmark. Bold font indicates best result for the line.

	Target Coverages	Calibrated Risk			
		SelectiveNet	ConfidNet	DeepGamblers	Ours
VGG-16 (SVHN)	1.00	**3.64±0.15**	4.32±0.12	20.63±1.16	4.33±0.12
	0.95	**1.64±0.10**	2.46±0.18	17.42±1.10	2.13±0.14
	0.90	**1.01±0.08**	1.50±0.22	14.20±0.73	1.14±0.13
	0.85	0.78±0.06	1.13±0.34	11.02±0.71	**0.73±0.06**
	0.80	0.67±0.05	1.01±0.43	7.90±0.84	**0.58±0.05**
	0.75	0.60±0.03	1.00±0.45	5.36±0.86	**0.56±0.05**
	0.70	0.62±0.03	0.99±0.45	3.28±0.58	**0.55±0.05**
ResNet-34 (SVHN)	1.00	3.93±0.20	3.87±0.10	24.07±1.71	**3.86±0.10**
	0.95	2.32±0.27	1.78±0.09	21.19±1.80	**1.74±0.17**
	0.90	0.95±0.10	0.96±0.08	18.23±2.17	**0.92±0.06**
	0.85	**0.66±0.04**	0.70±0.05	14.85±2.27	**0.66±0.03**
	0.80	**0.56±0.05**	0.68±0.05	11.13±1.95	0.63±0.02
	0.75	**0.55±0.04**	0.66±0.04	7.92±1.77	0.62±0.03
	0.70	**0.54±0.06**	0.65±0.03	5.08±1.41	0.62±0.02
DenseNet-121 (SVHN)	1.00	**4.09±0.06**	4.29±0.08	26.89±2.38	4.28±0.09
	0.95	**1.92±0.12**	2.16±0.16	23.91±2.65	2.11±0.13
	0.90	**1.18±0.09**	1.27±0.07	20.77±2.68	1.27±0.05
	0.85	1.00±0.04	0.97±0.04	17.36±2.86	**0.95±0.05**
	0.80	0.92±0.07	0.86±0.05	14.01±2.93	**0.84±0.05**
	0.75	0.88±0.14	**0.80±0.05**	10.61±2.84	**0.80±0.05**
	0.70	0.75±0.04	0.76±0.05	7.39±2.72	**0.75±0.06**

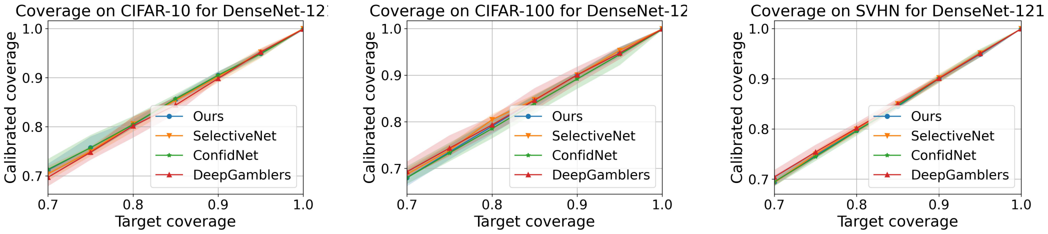
A.3 Additional plots

In this section we report additional plots which complement the analysis of Fig. 1 and Fig. 2 over the other considered architectures, namely DenseNet-121 and ResNet-34.



(a) Calibrated risk on CIFAR-10 (b) Calibrated risk on CIFAR-100 (c) Calibrated risk on SVHN

Figure 4: Calibrated risk for DenseNet

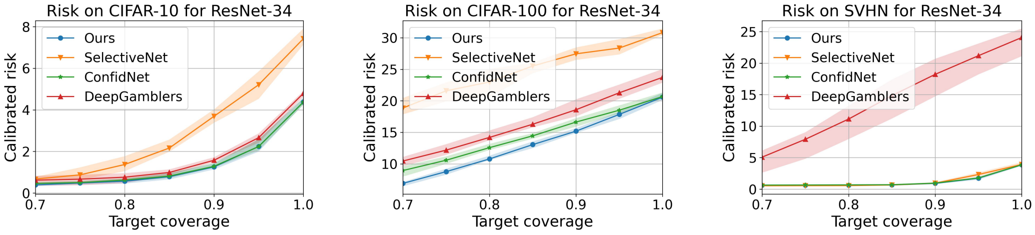


(a) Calibrated coverage on CIFAR-10

(b) Calibrated coverage on CIFAR-100

(c) Calibrated risk on SVHN

Figure 5: Calibrated coverage for DenseNet

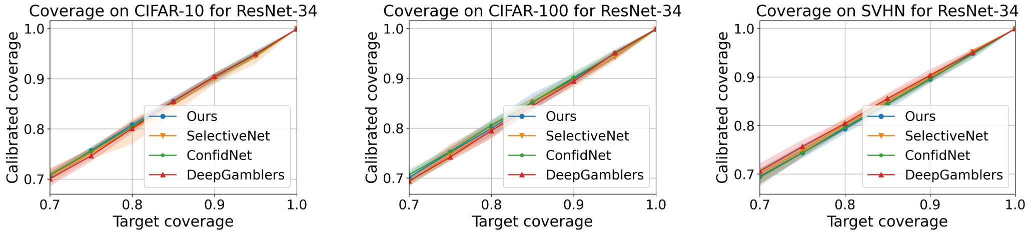


(a) Calibrated risk on CIFAR-10

(b) Calibrated risk on CIFAR-100

(c) Calibrated risk on SVHN

Figure 6: Calibrated risk for ResNet



(a) Calibrated coverage on CIFAR-10

(b) Calibrated coverage on CIFAR-100

(c) Calibrated risk on SVHN

Figure 7: Calibrated coverage for ResNet