Trusting the Untrustworthy: A Cautionary Tale on the Pitfalls of Training-based Rejection Option

Anonymous authors
Paper under double-blind review

Abstract

We consider the problem of selective classification, also known as rejection option. We first analyze state-of-the-art methods that involve a training phase to produce a selective classifier capable of determining when it should abstain from making a decision. Although only some of these frameworks require changes to the basic architecture of the classifier, by adding a module for selection, all methods necessitate implementing modifications to the standard training procedure and loss function for classification. However, our observations reveal that, due to the nature of these modifications, the outcome of the considered frameworks varies significantly in terms of coverage and risk across the classes. Additionally, some of these methods also decrease the accuracy of the final classification. We discuss the limitations of each framework, demonstrating that these shortcomings occur for a wide range of models and datasets. We establish a mathematical connection between the problem of detecting misclassification errors and the risk minimization for selective classification, proposing a statistical test that does not require training and can be applied to pre-trained standard classifiers to enable them with a rejection option.

1 Introduction

In many applications, incorrect decisions can have severe consequences. Therefore, detecting and preventing them is crucial. Consequently, significant efforts are being made in various areas of artificial intelligence to enhance the reliability of automatic systems, as they are known to be prone to errors (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)).

Avoiding wrong decisions by abstaining has been investigated in the field of artificial intelligence since its early stages Chow (1957). In this paper, we consider the problem of enhancing the reliability of a model by incorporating a rejection option. Standard models are designed to provide answers related to the task they have learned when presented with input samples. By incorporating a rejection option, these models have the ability to abstain from providing a decision when deemed too risky. 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. While current state-of-the-art train-based rejection option methods achieve great performance in controlling the global risk based on target coverage, a non negligible imbalance is observed when looking at their performance class by class in Figure 1. This work aims to shed a light on the causes of this unwanted behavior and to identify alternative principled approaches that mitigate disparities across classes.

Main contributions. Our contribution is threefold:

1. We show that train-based models for selective classification, although they provide consistent performance across the entire dataset, exhibit a large variance in terms of risk and coverage across the classes. This means that over multiple independent runs of the same model and datasets, these methods attain extremely different coverage and risk for certain classes.

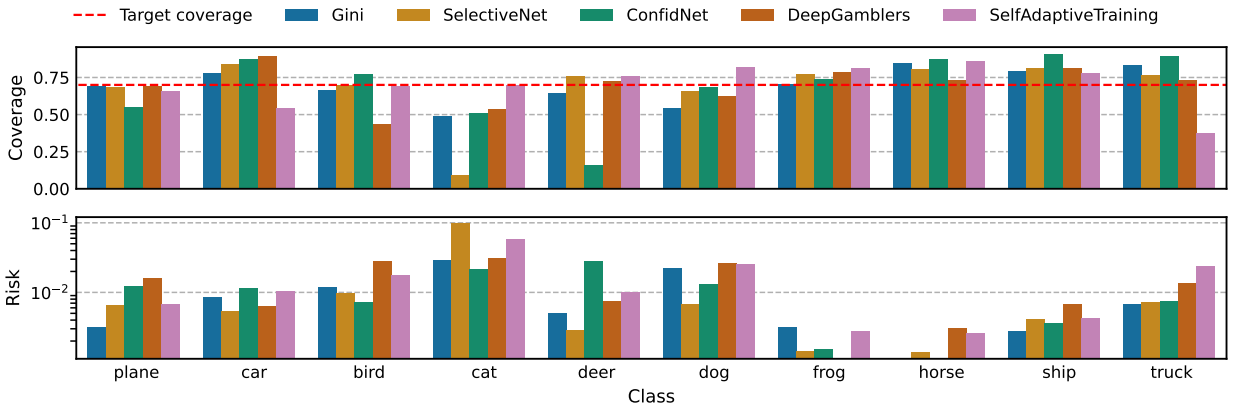


Figure 1: The performance of training-based methods often falls short, resulting in cases where the coverage (top) is low for at least one class, but the risk (bottom) is high. This undermines the reliability of these methods.

2. We investigate and propose insights on the reason for this variation in risk and coverage across classes. Additionally, we note that some of the train-based methods also decrease performance in terms of classification accuracy. We support these observations by presenting results that involve several models (ResNet-34, DenseNet-121, and VGG-16) and benchmarks (CIFAR-10, CIFAR-100, and SVHN).
3. We make a mathematical connection between the problem of risk minimization in selective classification and the problem of error minimization in the context of misclassification detection. We formally establish that addressing the latter problem with a state-of-the-art solution provides a method to implement the rejection option for standard pre-trained classifiers without training selective models and attaining favorable results.

2 Related works

Strategies to reject wrong or low-confidence decisions have been studied in the context of artificial intelligence since its early days Flores (1958); Chow (1970); Pudil et al. (1992). Rejections can be broadly categorized into two groups Hendrickx et al. (2021): ambiguity, where the model is unable to replicate the optimal decision for certain inputs Hellman (1970); Fukunaga & Kessell (1972), and novelty, where inputs at test time are significantly different from those encountered during training (Vasconcelos et al., 1995; Seo et al., 2000; Vailaya & Jain, 2000).

The growing body of work in deep learning has led to a renewed interest in the problem of decision rejection. Initially, researchers have been focusing on new metrics to assess the confidence of a model. Hendrycks & Gimpel (2017); Geifman & El-Yaniv (2017) propose to use the maximum of the softmax distribution output by a model as a confidence score on its decisions; Jiang et al. (2018) introduce a trust score which is proportional to the agreement between the considered model and a nearest-neighbor algorithm modified to only account for confident decisions; Gal & Ghahramani (2016) model the uncertainty using Monte Carlo Dropout to estimate the posterior predictive network distribution by sampling many stochastic network predictions.

More recent works in the field have focused on embedding the concept of rejection option directly at training time. These approaches propose either updated loss functions to account for the measure of risk related to accepting or rejecting a decision Liu et al. (2019) or a 'selection architecture' that returns an abstention output alongside the classification output Corbière et al. (2019), or both Geifman & El-Yaniv (2019). Huang et al. (2020) introduce a method to enhance the generalization of deep models through empirical risk minimization, specifically when dealing with corrupted data. By contributing to the calibration of the model's output during training, the technique is reported to increase the final model's rejection decision. This result is also supported by Fisch et al. (2022).

Two recent studies related to the problem of selective classification are Schreuder & Chzhen (2021) and Gangrade et al. (2021b). The former focuses on fairness and adds the concept of demographic parity to the risk and coverage loss functions as an additional requirement. This results in a more balanced rejection rate for underrepresented groups in datasets such as Adults and Germans, where minority groups are present, at the cost of accuracy loss. As to the latter, to the best of our efforts due to the lack of official published code, we have not been able to reproduce the comparison with Geifman & El-Yaniv (2019); Liu et al. (2019) and add the comparison with Corbière et al. (2019) which is missing in Gangrade et al. (2021b). Feng et al. (2022) revisits Hendrycks & Gimpel (2017), and they have proposed to further regularize popular objective functions with entropy-minimization at training time. Rabanser et al. (2022) introduce a framework that, for a given test input, monitors the disagreement with the final predicted label over the intermediate models obtained during training. Although no active training is required, these frameworks need all the side information contained in the training dynamics. For both works, the code release is still pending. More recently, Granese et al. (2021) introduce a new simple state-of-the-art framework for misclassification detection which builds and improves on Hendrycks & Gimpel (2017). They apply 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 since it only uses the soft-probabilities output by the model, and will be formally linked to selective classification.

For completeness, we mention important theoretical results Herbei & Wegkamp (2006); Franc et al. (2021); Fischer et al. (2016), and Cao et al. (2022) where the authors show the intrinsic equality between standard classification and selective classification by addition of a class representing abstention. Moreover, we observe how the interesting topic of rejection option crossed the boundaries of other well-established research areas such as certified robustness Cohen et al. (2019); Tramèr (2022), adversarial examples detection Aldahdooh et al. (2021), distribution shift Snoek et al. (2019), conformal prediction Einbinder et al. (2022), extension of cost-sensitive selective classification (Liu et al. (2019)) to a larger family of loss functions Charoenphakdee et al. (2021), and selective classification in a more relaxed scenario, where extra side information in the shape of classification feedback is provided in case of abstention Gangrade et al. (2021a). Finally, we reference Zhang et al. (2023) as the most up to date survey paper on the topic.

3 Background on rejection option

Our study shows that training-based frameworks for rejection options often have inconsistent performance when evaluated on multiple instances of the same experiment. These methods may have adequate risk and coverage rates overall but may reject samples from specific classes excessively. Although some classes are naturally harder to classify than others, the unwanted behavior mentioned above sheds light on the sensitivity of the frameworks to the initialization of the underlying model. We propose a brief analysis of popular train-based frameworks to understand these limitations.

Let us consider a standard classification task, where $\mathcal{X} \subseteq \mathbb{R}^d$ is the feature space and $\mathcal{Y} = \{1, \dots, C\}$ is the label space. Let $\mathcal{D}_n = \{(\mathbf{x}_i, y_i)\}_{i=1}^n \sim p_{XY}$ denote the training set as a random realization of n i.i.d. samples according to p_{XY}, the underlying and unknown probability density function over $\mathcal{X} \times \mathcal{Y}$. Let us define the *predictor* (i.e., the classifier) 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. For clarity, we define a soft-probability model that is associated with the predictor $f_{\mathcal{D}_n}$, s.t. $h_{\mathcal{D}_n} : \mathcal{X} \to \mathbb{R}^C$, s.t. $h_{\mathcal{D}_n,y}(\mathbf{x}) \in (0,1)$ and $\sum_{y=1}^C h_{\mathcal{D}_n,y}(\mathbf{x}) = 1$, i.e. $h_{\mathcal{D}_n}(\cdot)$ outputs $P_{\widehat{Y}|X}$[1]. The predictor is usually trained with the cross-entropy (CE) loss. Let $\mathrm{S} : \mathcal{X} \to \{0, 1\}$ be the *selector* which is responsible for rejecting/accepting the decision made by the *predictor*.

The <u>selective model</u> for a sample $\mathbf{x} \in \mathcal{X}$ is defined as:

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

where $\emptyset$ indicates that $f_{\mathcal{D}_n}$ abstains from the prediction.

[1]To simplify the notation, in the remaining of the work we will use f and h interchangeably with $f_{\mathcal{D}_n}$ and $h_{\mathcal{D}_n}$, respectively.

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

$$\hat{\phi}(\mathrm{S};\mathcal{D}_m) \triangleq \frac{1}{m}\sum_{i=1}^{m}\mathrm{S}(\mathbf{x}_i), \tag{2}$$

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

$$\hat{r}\left(f_{\mathcal{D}_n},\mathrm{S};\mathcal{D}_m\right) \triangleq \frac{\sum_{i=1}^{m}\mathbb{1}_{[f_{\mathcal{D}_n}(\mathbf{x}_i)\neq y_i]}\mathrm{S}(\mathbf{x}_i)}{\sum_{i=1}^{m}\mathrm{S}(\mathbf{x}_i)}, \tag{3}$$

where $\mathcal{D}_m = \{(\mathbf{x}_i, y_i)\}_{i=1}^{m}$ is the test or evaluation set and $\mathbb{1}_{[\cdot]}$ denotes the indicator function.

Instead of relying on the observation of the post-training performance of the model to decide whether a decision for an input sample should be accepted or rejected, train-based methods embed the rejection option within the training phase imposing the use of continuous and derivable functions for the selection, combined with architecture restructuring Geifman & El-Yaniv (2019); Corbière et al. (2019) and convex combinations of multiple loss functions, or upgraded versions of the CE loss function that take into account a $|C|+1$-th class that corresponds to the rejection option Liu et al. (2019); Huang et al. (2020). Our main observation is that, while these methods appear promising from a theoretical perspective, in practice they require significant tuning to achieve near-optimal solutions on heterogeneous datasets. This tuning requires additional samples to optimize the hyper-parameters involved in the training process. Without this consideration, the performance of these methods may exhibit unpredictable and undesirable behavior.

SelectiveNet (SN): Geifman & El-Yaniv (2019) propose to train their selective classifier by minimizing the loss function $\mathcal{L}_{SN}(h, \mathrm{S}_{SN}, h', c; \mathcal{D}_n) = \alpha \times A + (1-\alpha)\times B$, where $A = r\left(h, \mathrm{S}_{SN}; \mathcal{D}_n\right) + \lambda \times \Psi\left(\mathrm{S}_{SN}, c; \mathcal{D}_n\right)$, $B = \frac{1}{m}\sum_{i=1}^{m}\ell\left(h'\left(\mathbf{x}_i\right), y_i\right)$, $\mathrm{S}_{SN}:\mathcal{X}\rightarrow[0,1]$ represents a soft selector, $\phi(\mathrm{S}_{SN};\mathcal{D}_n) \triangleq \frac{1}{n}\sum_{i=1}^{n}\mathrm{S}_{SN}(\mathbf{x}_i)$, $r(h, \mathrm{S}_{SN}; \mathcal{D}_n) \triangleq \frac{\frac{1}{n}\sum_{i=1}^{n}\ell(h(\mathbf{x}_i), y_i)\mathrm{S}_{SN}(\mathbf{x}_i)}{\phi(\mathrm{S}_{SN};\mathcal{D}_n)}$ is the soft risk, $\Psi(\mathrm{S}_{SN}, c; \mathcal{D}_n) = \max\{0, (c-\phi\left(\mathrm{S}_{SN};\mathcal{D}_n\right))^2\}$ denotes the constraint for the target coverage c, $\ell: Y\times Y\rightarrow\mathbb{R}^+$ is the loss function related to the classification task (e.g., cross-entropy), and $h':\mathcal{X}\rightarrow\mathbb{R}^C$ is implemented by an auxiliary model that shares the same body of the model implementing h but has an independent prediction head. Finally, λ and α, are fixed to 32 and 0.5, respectively. While the convex loss function proposed by the authors has a global minimum, the gradient descent training algorithm does not guarantee the absence of multiple local minima that, while not far from the optimal solution, may arrange risk and coverage over the classes in different ways that provide similar global values for the loss function. This issue is exacerbated by the lack of validation for the parameters λ and α. In particular, the case presented in Figure 1 shows how SelectiveNet exhibits a much more conservative coverage than the other methods for the class "cat" in CIFAR-10, without containing the risk. Figure 4 shows that this is not uncommon, as both coverage and risk have many outlier results when considering the same experiment with 10 different random initializations.

ConfidNet (CN): given a pre-trained predictor f, Corbière et al. (2019) minimize the loss $\mathcal{L}_{\mathrm{CN}}(\mathrm{S}_{CN}; h, \mathcal{D}_n) = \frac{1}{n}\sum_{i=1}^{n}\left(\mathrm{S}_{CN}\left(\mathbf{x}_i\right) - h_{y_i}\left(\mathbf{x}_i\right)\right)^2$, where, during training, $\mathrm{S}_{CN}:\mathcal{X}\rightarrow[0,1]$ is the learned selector, and h_{y_i} represents the confidence of h for the prediction f over the ground truth class. In a nutshell, the training algorithm learns to predict the confidence of the model by analyzing the true posterior class probability of supervised samples. According to the theoretical framework introduced in Corbière et al. (2019), inputs with a low true class probability (smaller than $\frac{1}{C}$) are misclassified, while inputs with a true class probability greater than $\frac{1}{2}$ are correctly classified. While this method yields impressive results, there is no guarantee that the confidence levels for correctly and incorrectly classified samples will not overlap in the interval $[\frac{1}{C}, \frac{1}{2}]$. As a result, the algorithm may assign different confidences to the same samples across multiple iterations, leading to a good overall performance at the cost of one or more classes being rejected more than necessary in some iterations if their samples are assigned too low a confidence. This can be observed in Figure 1, where these methods exhibit very low coverage with high associated risk for the class deer in CIFAR-10. Such a phenomenon is repeated over different classes in the set of experiments with 10 different initialization seeds considered in Figure 4, where ConfidNet exhibits many outliers in both coverage and risk.

DeepGamblers (DG): Liu et al. (2019) implements the rejection option for a predictor f by optimizing $\mathcal{L}_{DG}(h;\mathcal{D}_n) = -\frac{1}{n}\sum_{i=1}^{n}\log\left(h_{y_i}(\mathbf{x}_i)o_i + h_{C+1}(\mathbf{x}_i)\right)$, where h_{y_i} represents the confidence associated to the

correct class $y_i \in \mathcal{Y}$ for the input sample $\mathbf{x}_i \in \mathcal{X}$, $h_{C+1}(\cdot)$ is the model's confidence for the extra class that represents the selection and the coefficient o is the payoff associated to accepting a decision. In a nutshell, the function above corresponds to the classical cross-entropy for $o_i = 1$ and $h_{C+1}(\cdot) = 0$. The training algorithm requires a warm-up period during which the weights are updated by optimizing the classic cross-entropy loss for about one-third of the total training iterations. After that, the loss above replaces the standard cross-entropy and the model is trained to learn when to accept/reject decisions. As by admission of the authors themselves, the cost term o_i, which is a hyper-parameter that requires tuning through the information conveyed by extra samples, is a key component of the algorithm. In particular, a lower o causes the model to learn to reject better, but with a larger variance. Due to the lack of any indication on how to choose the right value for o, and since we did not use validation to be fair to the other frameworks, in our experiments, we maintained a fixed value $o = 2.2$ for CIFAR-10, $o = 2.6$ for SVHN reported as indicated in the original paper Liu et al. (2019), and $o = 2.0$ for CIFAR-100 which is the suggested default value. Figure 1 exhibits subpar coverage and high risk for this method when considering the class bird in CIFAR-10. Non-negligible variance, although lower than w.r.t. other competitors, is consistently reported across the experiments with 10 different initialization seeds reported in the box-plots in Figure 4.

Self-Adaptive Training (SAT): although Huang et al. (2020) primarily deals with the problem of generalization improvement under the assumption of potentially corrupted training data, the authors show how their method can be adapted to the problem of selective classification by adding an extra class to represent the rejection (as in Liu et al. (2019)) directly using the model prediction as a signal for learning abstention. The model is initially trained with the standard cross-entropy loss function during a warm-up set of epochs (60 in the basic algorithm reported in the paper). Then the cross-entropy loss is replaced by $\mathcal{L}_{SAT}(h; \mathcal{D}_n) = -\frac{1}{n}\sum_{i=1}^{n}[t_{i,y_i}\log(h_{y_i}(\mathbf{x}_i)) + (1 - t_{i,y_i})\log(h_{C+1}(\mathbf{x}_i))]$. In this equation, the subscript y_i is the index of the correct class for the input $\mathbf{x}_i$ and $\mathbf{t}_i$ is a convex combination defined as $\mathbf{t}_i = \alpha \times \mathbf{y}_i + (1-\alpha) \times h(\mathbf{x}_i)$, where $\mathbf{y}_i$ is the one-hot encoded version of the true label for $\mathbf{x}_i$. According to the basic algorithm reported in the paper, $\alpha = 0.9$. Clearly a small t_{i,y_i} reveals low confidence and enforces the selector to reject the decision. If t_{i,y_i} is close to one, abstention becomes unlikely, and the function above recovers the standard cross-entropy. This framework achieves favorable results across the independent experiments summarized by the plot in Figure 4. However, the role played by a non-optimized number of warm-up epochs or value of α necessarily affects the value of $\mathbf{t}_i$, which, in turn, may cause poor performance as reported in Figure 1 for the class "truck" in CIFAR-10.

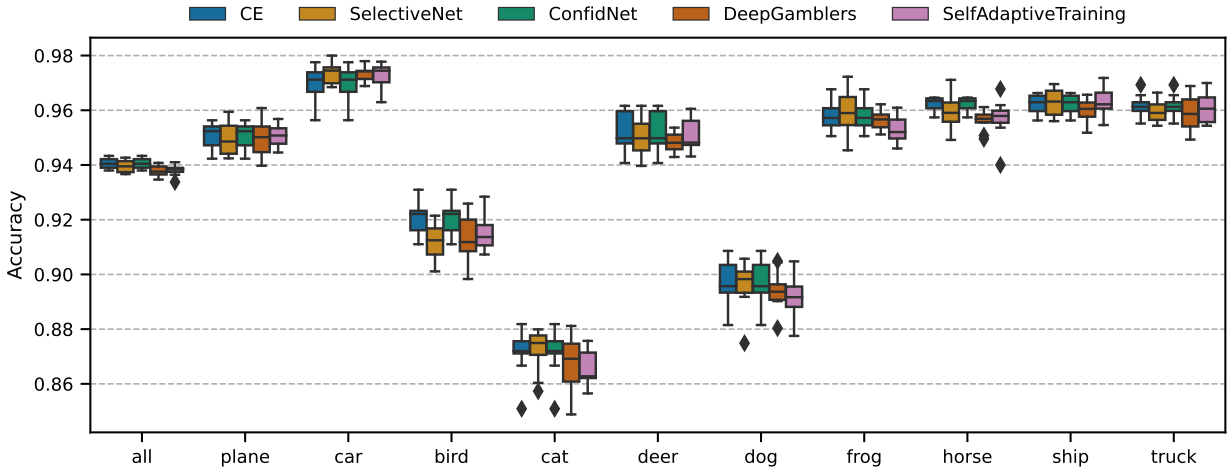


Figure 2: Global and class-wise accuracy for a VGG-16 model trained on CIFAR-10 with a cross-entropy (CE) loss and with the loss defined on state-of-the-art training-based rejection option methods over 10 different runs.

4 Rejection option as a special case of optimal misclassification detection

In this section, we establish the **relationship between the rejection option and misclassification detection** by mathematically deriving the optimal risk of the oracle error detector. In doing so, we prove that the optimization of this problem leads to the minimization of risk as defined in Equation (3). A similar derivation was introduced in Chow (1970). We then consider a recently developed misclassification detection framework Granese et al. (2021) that has been shown to outperform previous methods Geifman & El-Yaniv (2017), and we argue for its use as a rejection/acceptance selector. One of the key advantages of this method is that it relies on a state-of-the-art classifier trained using classical cross-entropy loss optimization, resulting in models that converge to similar final performance despite being randomly initialized. We show that because of this reason, **the acceptance/rejection selection for these models within our proposed framework has lower variance**.

4.1 Preliminaries

Let us consider a discrete r.v. $E = \mathbb{1}[f(X) \neq Y]$. The *misclassification* event is defined as $E = 1$. We can express the probability density function p_{XY} as a mixture:

$$p_{XY}(\mathbf{x}, y) = p_{XY|E}(\mathbf{x}, y|E = 1)P_E(1) + p_{XY|E}(\mathbf{x}, y|E = 0)P_E(0). \tag{4}$$

By taking the marginal of Equation (4) over Y, we obtain:

$$p_X(\mathbf{x}) = p_{X|E}(\mathbf{x}|1)P_E(1) + p_{X|E}(\mathbf{x}|0)P_E(0) \tag{5}$$

where $p_{X|E}(\mathbf{x}|1)$ denotes the pdf truncated to the error event and $p_{X|E}(\mathbf{x}|0)$ the pdf truncated to the event of correct classification. Let us also define $\mathcal{X}_0 = \{\mathbf{x} \in \mathcal{X} : E(\mathbf{x}) = 0\}$ the set of correctly classified samples and $\mathcal{X}_1 = \{\mathbf{x} \in \mathcal{X} : E(\mathbf{x}) = 1\}$ the set of incorrectly classified samples. It is simple to verify that $\mathcal{X} = \mathcal{X}_0 \cup \mathcal{X}_1$, $\mathcal{X}_0 \cap \mathcal{X}_1 = \varnothing$, $p_{X|E}(\mathbf{x}|1) = 0$ if $\mathbf{x} \in \mathcal{X}_0$, and, inversely, $p_{X|E}(\mathbf{x}|0) = 0$ if $\mathbf{x} \in \mathcal{X}_1$. Then, the *optimal selector* is given by $S^\star(\mathbf{x}) = 1$ whenever $\mathbf{x} \in \mathcal{X}_0$ and $S^\star(\mathbf{x}) = 0$ otherwise.

The *Probability of classification error* is finally defined as:

$$\mathrm{Pe}(\mathbf{x}) \triangleq P_{E|X}(1|\mathbf{x}) = 1 - P_{Y|X}\left(f_{\mathcal{D}_n}(\mathbf{x}) \mid \mathbf{x}\right) \tag{6}$$

4.2 From selective risk to probability of error

Let us recall the risk definition from El-Yaniv & Wiener (2010) with the standard 0/1 loss. We will expand the selective model framework outlined in Section 3 by rewriting the risk in terms of the underlying probability distribution p_X and the mixture model proposed in Equation (5):

$$r(f, \mathrm{S}) \triangleq \frac{\mathbb{E}_{XY}[\mathbb{1}_{[f_{\mathcal{D}_n}(\mathbf{x}) \neq y]}\mathrm{S}(\mathbf{x})]}{\mathbb{E}_X[\mathrm{S}(\mathbf{x})]} = \frac{\mathbb{E}_{X|1}[\mathrm{S}(\mathbf{x})]}{\mathbb{E}_{X|1}[\mathrm{S}(\mathbf{x})] + \frac{P_E(0)}{P_E(1)}\mathbb{E}_{X|0}[\mathrm{S}(\mathbf{x})]} = \frac{P_{\mathrm{II}}}{P_{\mathrm{II}} + \beta\,(1 - P_{\mathrm{I}})}, \tag{7}$$

where $\beta = \frac{P_E(0)}{P_E(1)} = \frac{\text{Accuracy}}{1-\text{Accuracy}}$ is a constant[2]. We relegate the full derivation of Equation (7) to the Appendix 8.1.

In Equation (7), we show that the selective risk can be expressed in terms of the Error of Type-I (i.e., rejection of a sample that would be correctly classified) and the Error of Type-II (i.e., accepting a sample that would be misclassified). However, the error event is unobservable. Fortunately, according to Granese et al. (2021) (cf. Proposition 3.1 therein), an oracle, who knows all the involved probability distributions, can control the trade-off between both types of errors with a threshold, thus, the same oracle can control the selection risk which is also a function of P_{I} and P_{II}. By minimizing Equation (7) for all admissible P_{I} and P_{II}, we can derive the optimal selector and obtain Equation (8) where $\gamma^\star \in \mathbb{R}^+$ is a threshold:

$$\mathrm{S}^*(\mathbf{x}; \gamma^\star) = \mathbb{1}\left[\frac{\mathrm{Pe}(\mathbf{x})}{1 - \mathrm{Pe}(\mathbf{x})} \leq \gamma^\star\right]. \tag{8}$$

[2]We only consider the derivation for classification models that are not perfectly accurate, as otherwise, the necessity for a rejection option would not be adequately justified.

4.3 Gini selector

Since the true probability of error Pe is unknown and cannot be learned from samples, we will rely on an approximation of the optimal selector based on the state-of-the-art method for misclassification detection developed in Granese et al. (2021) and introduced in Equation (9):

$$\text{Gini}(\mathbf{x}) \triangleq \sum_{y \in \mathcal{Y}} P_{\widehat{Y}|X}(y|\mathbf{x}) \Pr(\widehat{Y} \neq y|\mathbf{x}) = 1 - \sum_{y \in \mathcal{Y}} P^2_{\widehat{Y}|X}(y|\mathbf{x}). \tag{9}$$

Interestingly, we found out that Equation (9) can be linked to a popular information theoretic measure which is the Rényi divergence. We reference Section 8.2 for a more in-depth analysis.

Definition 4.1 (Rejection option with Gini(·))**.**

$$(f_{\mathcal{D}_n}, \text{Gini}, \gamma)(\mathbf{x}) \triangleq \begin{cases} f_{\mathcal{D}_n}(\mathbf{x}) & \text{if Gini}(\mathbf{x}) \leq \gamma \\ \emptyset & \text{if Gini}(\mathbf{x}) > \gamma, \end{cases} \tag{10}$$

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

With Equations (7) and (9) and Definition 4.1 we have established a strong link between Type-I and Type-II errors from the point of view of misclassification detection and risk in the context of selective classification. This justifies the use of Equation (10) for post-training methods to implement the rejection option. In addition, Figure 2 shows that classifiers based on standard cross-entropy loss at training time (e.g. the base classified in ConfidNet) show consistently high accuracy with low variance, suggesting that this should also be reflected in the selection mechanism that accepts/rejects decisions. Inspired by this observation, and aware of the limitations exposed in Section 3, we plot the empirical cumulative density function of Gini and ConfidNet over 10 runs in Figure 3. We can see that the Gini selection score has less distributional variability compared to ConfidNet. This observation supports our suggestion that the use of a pure cross-entropy training approach could lead to more consistent results and may be beneficial when seeking robust and reliable results in the context of selective classification, which is confirmed experimentally in Figure 4.

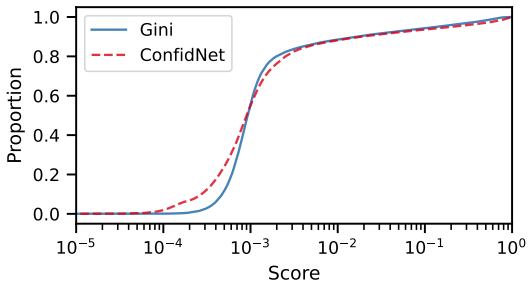


Figure 3: Empirical cumulative distribution of scores with 95% confidence intervals for Gini and ConfidNet for 10 different runs with CIFAR-10 and VGG-16, showing an important gap in variance difference between a post-hoc and a training-based soft-selection score.

5 Experimental Design

We experimentally analyze SelectiveNet (cf. Geifman & El-Yaniv (2019)), ConfidNet (cf. Corbière et al. (2019)), DeepGambler (cf. Liu et al. (2019)), and SelfAdaptiveTraining (cf. Huang et al. (2020)) compared to our post-hoc method denoted Gini.

Models and benchmark. In our experiments, we consider three datasets: CIFAR-10, CIFAR-100 Krizhevsky (2009), and SVHN Netzer et al. (2011). For each of the train based methods, we train the underlying models using the entire training sets and following the training guidelines as reported in the corresponding papers. We train the standard classifier model, which we apply our Gini selector to, using the cross-entropy loss function and a stochastic gradient descent optimizer with momentum and learning rate scheduling for 300 epochs and batch size of 64. To expand on previous research, we consider three neural network architectures: VGG-16, ResNet, and DenseNet, as well as the more challenging CIFAR-100 dataset. We conducted experiments with 10 different random seeds for each model, dataset, and method, and include error bars in our main results. Additionally, we also perform experiments on the larger ImageNet Deng et al. (2009) dataset to

further analyze the proposed post-hoc solution, which can be easily implemented using publicly available state-of-the-art model checkpoints.

Coverage calibration. To ensure fair and comparable results, we standardize the procedure for coverage calibration across all methods, which was not always clear in previous works. We divide the test partition for each of the three considered datasets into two subsets: one for calibration and another for evaluation. The calibration set corresponds to 10% of the original partition, chosen randomly for each of the ten seeds. The coverage calibration algorithm is outlined in Algorithm 1, where $\mathcal{D}_{m'}$ denotes the calibration dataset of size m'. We set target coverages τ from 0.50 to 1.00 with increments of 0.05. Intuitively, in order to guarantee the target coverage, we calculate scores for all samples in the calibration set and order them in ascending order. Then, we select the score value at index $\lceil \tau \cdot m' \rceil$ as threshold.

Algorithm 1 Coverage calibration algorithm.

Input: Calibration set $\mathcal{D}_{m'} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{m'}$, selector S, and target coverage $\tau \in [0, 1]$
`SList` $= [\,]$
for $i = 1$ **to** m' **do**
  `SList.append`$(\mathrm{S}(\mathbf{x}_i))$
end for
`sort(SList, ascend=True)`
Return: $\gamma^* =$ `SList`$[\lceil \tau \cdot m' \rceil]$

Implementation details. From an implementation point of view, post-hoc rejection option methods are the more resource-efficient and cost-effective solutions as they require fewer or no hyperparameters to be tuned and no architectural changes to the models. This is particularly beneficial in scenarios where it is difficult to collect additional samples for parameter optimization. For instance, SelectiveNet requires fitting a model for each target coverage. ConfidNet adds a significant overhead especially during inference, with an auxiliary confidence network with 1 million additional parameters. This overhead may limit some applications, e.g., ML applications on the edge Murshed et al. (2022). DeepGamblers requires hyperparameter validation and warm-up with only CE loss. SelfAdaptiveTraining also requires warm-up with cross-entropy and may require tuning the momentum parameter, which is globally set to 0.9 in the original paper.

Source code and computational resources. In this study, we utilize a cluster of GPUs to train and evaluate the deep learning models, allowing for efficient parallelization of our experiments. Adhering to the principles of open science, we have made our code and trained models publicly available[3] to facilitate the reproducibility of our research. We hope that this benchmark will be useful to the research community and inspire further studies on rejection option.

6 Discussion

We compare the performance of the Gini selector with the performance of the train-based methods discussed in Section 3 across the datasets listed in Section 5 and report the results in Table 1.

6.1 Main results

Gini, despite its simplicity, exhibits comparable or superior performance across various target coverages for all three datasets. Specifically, for CIFAR-10, we see in Table 1 that the proposed method achieves better performance on average from 80 to 100% coverage. The results on CIFAR-100, shown in Figure 5, demonstrate that the Gini selector outperforms the other methods and can achieve at most 5% risk with a 50% coverage, while the other methods fail to achieve so with a large gap. We believe this to be due, at least in part, to the limited availability of additional data to validate hyper-parameters for the training-based methods, which results in sub-optimal performance. This trend is also observed for the other models in the benchmark, for which we report extended results in Appendix 8.3. the proposed method outperforms, on

[3]Anonymized source code: `https://github.com/giniselector/giniselector`

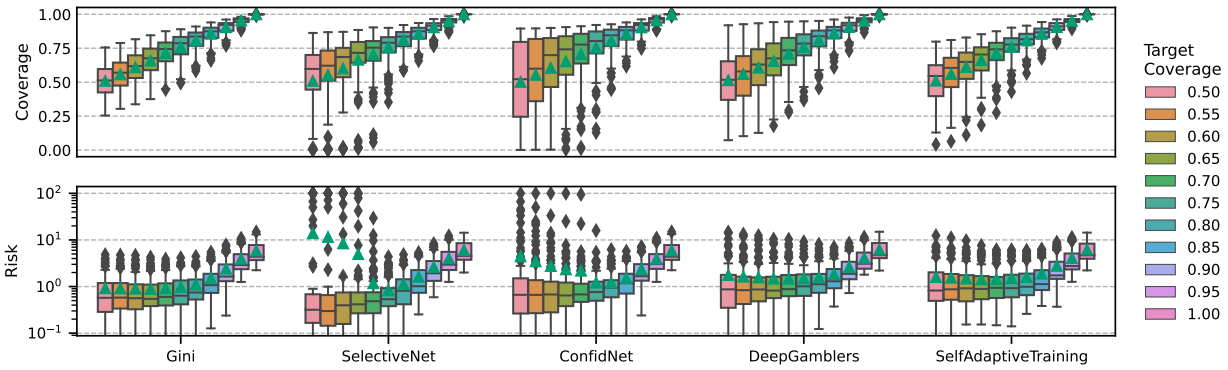


Figure 4: Box plots for the VGG-16 (CIFAR-10) benchmark comparing the coverage and risk of the proposed method to that of competing methods over 10 seeds and classes individually. The proposed method shows fewer outliers, indicating a more reliable performance. The green triangle marker indicates the average metric and the whiskers show the median.

average, all the state-of-the-art methods. It On the SVHN dataset, the proposed method attains comparable performance for lower coverage levels and slightly worse performance on higher coverage rates. The ConfidNet and the SelfAdaptiveTraining frameworks perform poorly in this last benchmark.

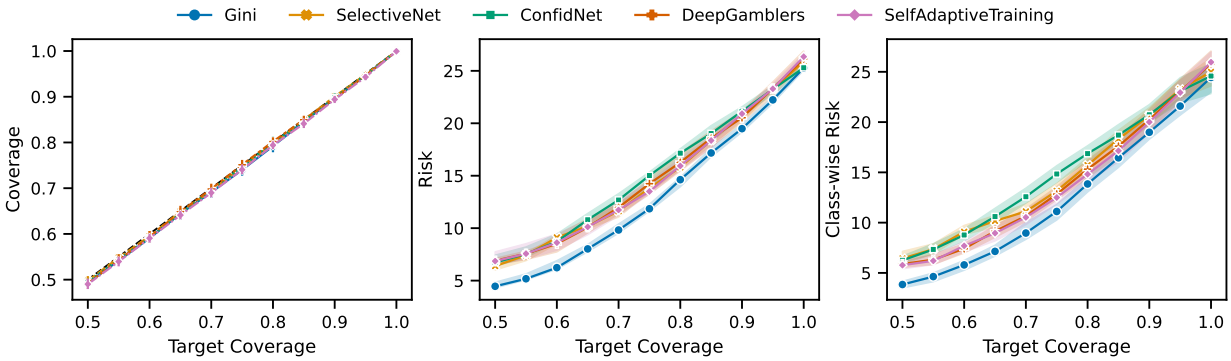


Figure 5: Coverage and risk comparing methods on a VGG-16 (CIFAR-100) model.

In Figure 4, we compare the performance of the proposed method to that of competing methods over 10 seeds and classes individually. The image contains multiple box plots that depict the calibrated coverage and risk of the methods for each class for specific target coverage. The box plots clearly show that the proposed method has a more consistent performance across different classes and seeds, with fewer outliers. In contrast, the competing methods exhibit a higher degree of variability, with many outliers. The image highlights the advantage of the proposed method in providing a more reliable and consistent performance across different classes and random initializations. These results demonstrate the effectiveness of the Gini selector in addressing the selective classification problem and its potential as a practical alternative to the training-based state-of-the-art. We provide similar plots in Section 8.3 for DenseNet and ResNet where similar trends are also observed.

6.2 Results on ImageNet

To study the relation between the accuracy and the risk linked to the rejection option on a large scale problem, we setup the following experiment. We consider five off-the-shelf pre-trained ResNet models with different numbers of parameters and increasing accuracy (ResNet-18, ResNet-34, ResNet-50, ResNet-101,

Table 1: Empirical selective risk (the lower the better given a fixed coverage) in percentage for the classification benchmark with VGG-16 for various target coverages over 10 runs. "Cov." stands for the target coverage.

	Cov.	Gini	SelectiveNet	ConfidNet	DeepGamblers	SAT
VGG-16 (CIFAR-10)	0.50	0.78 ± 0.1	0.36 ± 0.1	0.73 ± 0.2	1.07 ± 0.1	1.29 ± 0.2
	0.55	0.78 ± 0.1	0.39 ± 0.1	0.73 ± 0.1	1.08 ± 0.1	1.27 ± 0.2
	0.60	0.77 ± 0.1	0.49 ± 0.1	0.75 ± 0.1	1.12 ± 0.1	1.28 ± 0.1
	0.65	0.78 ± 0.1	0.58 ± 0.1	0.77 ± 0.1	1.14 ± 0.2	1.27 ± 0.1
	0.70	0.82 ± 0.1	0.63 ± 0.1	0.84 ± 0.1	1.20 ± 0.2	1.33 ± 0.1
	0.75	0.89 ± 0.1	0.73 ± 0.1	0.94 ± 0.1	1.28 ± 0.2	1.40 ± 0.1
	0.80	1.02 ± 0.1	1.03 ± 0.1	1.09 ± 0.2	1.43 ± 0.1	1.51 ± 0.1
	0.85	1.43 ± 0.2	1.51 ± 0.1	1.43 ± 0.2	1.72 ± 0.2	1.80 ± 0.1
	0.90	2.31 ± 0.3	2.51 ± 0.3	2.28 ± 0.2	2.53 ± 0.2	2.68 ± 0.3
	0.95	3.83 ± 0.4	3.92 ± 0.3	3.85 ± 0.3	4.01 ± 0.2	3.99 ± 0.3
	1.00	5.87 ± 0.2	6.01 ± 0.2	5.88 ± 0.2	6.14 ± 0.2	6.16 ± 0.2
VGG-16 (CIFAR-100)	0.50	4.53 ± 0.3	6.41 ± 0.5	6.86 ± 0.6	6.80 ± 0.5	7.04 ± 0.7
	0.55	5.32 ± 0.3	7.25 ± 0.4	7.73 ± 0.6	7.44 ± 0.5	7.70 ± 0.7
	0.60	6.47 ± 0.5	8.92 ± 0.5	8.97 ± 0.5	8.43 ± 0.6	8.69 ± 0.7
	0.65	8.14 ± 0.6	10.34 ± 0.5	10.76 ± 0.6	10.08 ± 0.7	10.15 ± 0.7
	0.70	9.92 ± 0.8	11.76 ± 0.9	12.75 ± 0.7	11.91 ± 0.7	11.84 ± 0.8
	0.75	12.10 ± 1.0	13.67 ± 0.5	14.74 ± 0.8	14.05 ± 0.8	13.72 ± 0.7
	0.80	14.55 ± 1.0	15.83 ± 0.4	17.01 ± 0.8	16.30 ± 0.9	16.06 ± 0.6
	0.85	17.17 ± 0.8	18.38 ± 0.6	18.99 ± 0.7	18.50 ± 0.9	18.30 ± 0.7
	0.90	19.58 ± 0.5	20.66 ± 0.5	21.15 ± 0.5	20.74 ± 0.8	20.89 ± 0.7
	0.95	22.23 ± 0.4	23.21 ± 0.4	23.24 ± 0.4	23.38 ± 0.8	23.42 ± 0.5
	1.00	25.27 ± 0.2	25.84 ± 0.3	25.31 ± 0.2	26.30 ± 0.5	26.43 ± 0.3
VGG-16 (SVHN)	0.50	0.39 ± 0.1	0.76 ± 0.3	1.96 ± 0.4	0.82 ± 0.1	8.74 ± 25.2
	0.55	0.42 ± 0.1	0.72 ± 0.1	2.13 ± 0.4	0.81 ± 0.1	8.74 ± 25.2
	0.60	0.47 ± 0.1	0.78 ± 0.2	2.34 ± 0.3	0.81 ± 0.1	8.74 ± 25.2
	0.65	0.52 ± 0.1	0.68 ± 0.1	2.54 ± 0.4	0.80 ± 0.1	8.74 ± 25.2
	0.70	0.60 ± 0.1	0.67 ± 0.1	2.76 ± 0.4	0.80 ± 0.1	8.74 ± 25.2
	0.75	0.70 ± 0.1	0.69 ± 0.1	2.98 ± 0.4	0.82 ± 0.1	8.75 ± 25.2
	0.80	0.83 ± 0.1	0.69 ± 0.1	3.20 ± 0.4	0.85 ± 0.1	8.76 ± 25.2
	0.85	1.11 ± 0.1	0.94 ± 0.1	3.52 ± 0.3	0.91 ± 0.1	8.82 ± 25.2
	0.90	1.68 ± 0.1	1.30 ± 0.1	3.93 ± 0.2	1.20 ± 0.1	9.11 ± 25.1
	0.95	2.97 ± 0.2	2.55 ± 0.2	4.40 ± 0.1	2.16 ± 0.2	10.02 ± 24.8
	1.00	5.31 ± 0.1	4.35 ± 0.1	5.31 ± 0.1	4.47 ± 0.2	11.96 ± 24.1

and ResNet-152) from Paszke et al. (2019); we evaluate the performance of the proposed Gini selector on the ILSVRC2012, or ImageNet-1K dataset (Deng et al., 2009) validation partition for each of the target coverages. We use 10% of this partition for coverage calibration and 90% for evaluation purposes. Figure 6 shows that the empirical risk of 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. Generally, increasing the accuracy by scaling up the base model will decrease the risk for fixed coverage, especially in the high coverage regime. These results reassure the practicality of the proposed post-hoc selection method.

7 Conclusion

This paper investigates the issue of large variance in risk and coverage across classes for train-based models for selective classification. We uncover the causes of this variation and present evidence to support our findings through results obtained from multiple models and benchmarks. Furthermore, we establish a mathematical

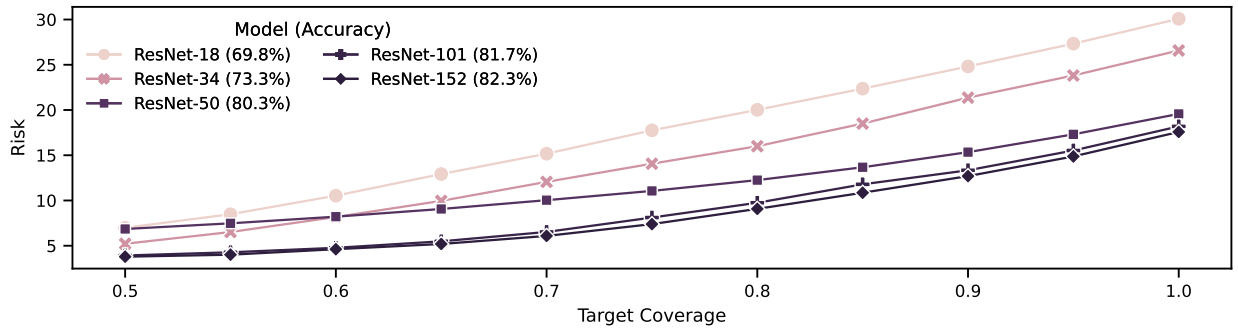


Figure 6: Risk-coverage curves for the Gini Selector on five ResNet models of different sizes and accuracies trained on ImageNet.

link between minimizing risk in selective classification and minimizing errors in misclassification detection. Our proposed solution offers a practical way to incorporate the rejection option for standard pre-trained classifiers. Perhaps, the main takeaway is that there is no free lunch when training confidence ranking functions, as their impressive global performance might come at a cost of unwanted behaviors across subgroups. Thus, we believe that this problem is open and that our results will encourage further research in the area.

References

Ahmed Aldahdooh, Wassim Hamidouche, and Olivier Déforges. Revisiting model's uncertainty and confidences for adversarial example detection, 2021.

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.

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.

Yuzhou Cao, Tianchi Cai, Lei Feng, Lihong Gu, Jinjie Gu, Bo An, Gang Niu, and Masashi Sugiyama. Generalizing consistent multi-class classification with rejection to be compatible with arbitrary losses. *Advances in Neural Information Processing Systems*, 35:521–534, 2022.

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.

Nontawat Charoenphakdee, Zhenghang Cui, Yivan Zhang, and Masashi Sugiyama. Classification with rejection based on cost-sensitive classification. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 1507–1517. PMLR, 2021. URL `http://proceedings.mlr.press/v139/charoenphakdee21a.html`.

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.

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.

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.

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.

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.

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.

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.

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

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

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

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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

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.

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.

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.

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.

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.

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.

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.

Jasper Snoek, Yaniv Ovadia, Emily Fertig, Balaji Lakshminarayanan, Sebastian Nowozin, D. Sculley, Joshua V. Dillon, Jie Ren, and Zachary Nado. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. 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. 13969–13980, 2019.

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.

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.

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.

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.

Xu-Yao Zhang, Guo-Sen Xie, Xiuli Li, Tao Mei, and Cheng-Lin Liu. A survey on learning to reject. *Proc. IEEE*, 111(2):185–215, 2023. doi: 10.1109/JPROC.2023.3238024. URL `https://doi.org/10.1109/JPROC.2023.3238024`.

8 Appendix

8.1 Full derivation of Equation (7)

In this section we include a more thorough derivation of the link between the selective risk and the probability of errors of Type-I and Type-II which are connected to the true probability of error.

$$
\begin{aligned}
r(f, \mathrm{S}) &\triangleq \frac{\mathbb{E}_{XY}[\mathbb{1}_{[f_{\mathcal{D}_n}(\mathbf{x})\neq y]}\mathrm{S}(\mathbf{x})]}{\mathbb{E}_X[\mathrm{S}(\mathbf{x})]} && (11)\\
&= \frac{\sum_{\mathbf{x}\in\mathcal{X}} \mathrm{S}(\mathbf{x}) \sum_{y\in\mathcal{Y}} p_{XY}(\mathbf{x}, y)\mathbb{1}_{[f_{\mathcal{D}_n}(\mathbf{x})\neq y]}}{\sum_{\mathbf{x}\in\mathcal{X}} p_X(\mathbf{x})\mathrm{S}(\mathbf{x})} && (12)\\
&= \frac{\sum_{\mathbf{x}\in\mathcal{X}} \mathrm{S}(\mathbf{x}) \sum_{y\in\mathcal{Y}} \left[p_{XY|E}(\mathbf{x}, y|1)P_E(1) + p_{XY|E}(\mathbf{x}, y|0)P_E(0)\right] \mathbb{1}_{[f_{\mathcal{D}_n}(\mathbf{x})\neq y]}}{\sum_{\mathbf{x}\in\mathcal{X}} \left[p_{X|E}(\mathbf{x}|1)P_E(1) + p_{X|E}(\mathbf{x}|0)P_E(0)\right] \mathrm{S}(\mathbf{x})} && (13)\\
&= \frac{\sum_{\mathbf{x}\in\mathcal{X}_1} \mathrm{S}(\mathbf{x}) \sum_{y\in\mathcal{Y}} p_{XY|E}(\mathbf{x}, y|1)P_E(1)}{P_E(1)\sum_{\mathbf{x}\in\mathcal{X}_1} p_{X|E}(\mathbf{x}|1)\mathrm{S}(\mathbf{x}) + P_E(0)\sum_{\mathbf{x}\in\mathcal{X}_0} p_{X|E}(\mathbf{x}|0)\mathrm{S}(\mathbf{x})} && (14)\\
&= \frac{P_E(1)\sum_{\mathbf{x}\in\mathcal{X}_1} p_{X|E}(\mathbf{x}|1)\mathrm{S}(\mathbf{x})}{P_E(1)\sum_{\mathbf{x}\in\mathcal{X}_1} p_{X|E}(\mathbf{x}|1)\mathrm{S}(\mathbf{x}) + P_E(0)\sum_{\mathbf{x}\in\mathcal{X}_0} p_{X|E}(\mathbf{x}|0)\mathrm{S}(\mathbf{x})} && (15)\\
&= \frac{\mathbb{E}_{X|1}[\mathrm{S}(\mathbf{x})]}{\mathbb{E}_{X|1}[\mathrm{S}(\mathbf{x})] + \frac{P_E(0)}{P_E(1)}\mathbb{E}_{X|0}[\mathrm{S}(\mathbf{x})]} && (16)\\
&= \frac{P_{\mathrm{II}}}{P_{\mathrm{II}} + \beta\,(1 - P_{\mathrm{I}})}. && (17)
\end{aligned}
$$

8.2 On the derivation of Gini$(\cdot)$ from the Rényi divergence

Let us define the Rényi divergence, as:

$$
D_\alpha\left(P_{\widehat{Y}|X}(\cdot, \mathbf{x})||Q_Y\right) \doteq \frac{1}{\alpha - 1}\log\left(\sum_{y\in\mathcal{Y}}\left(P^{\alpha}_{\widehat{Y}|X}(y|\mathbf{x})Q_Y^{(1-\alpha)}(y)\right)\right), \tag{18}
$$

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 $\alpha = 2$, Equation (18) becomes

$$
D_2\left(P_{\widehat{Y}|X}(\cdot, \mathbf{x})||Q_Y\right) = \log\left(\sum_{y\in\mathcal{Y}}\left(\frac{P^2_{\widehat{Y}|X}(y|\mathbf{x})}{Q_Y(y)}\right)\right). \tag{19}
$$

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

$$
\frac{1}{q}\sum_{y\in\mathcal{Y}}\left(P^2_{\widehat{Y}|X}(y|\mathbf{x})\right) \tag{20}
$$

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

8.3 Additional Results

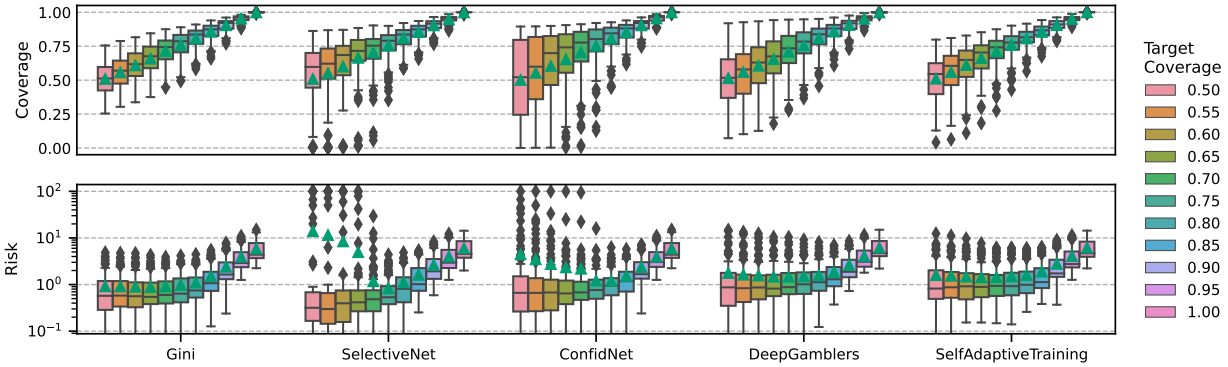
In this section, we present additional tables and plots that supplement the analysis we performed in the main manuscript.

Table 2: Empirical selective risk in percentage for the classification benchmark with DenseNet-121 for various target coverages over 10 runs.

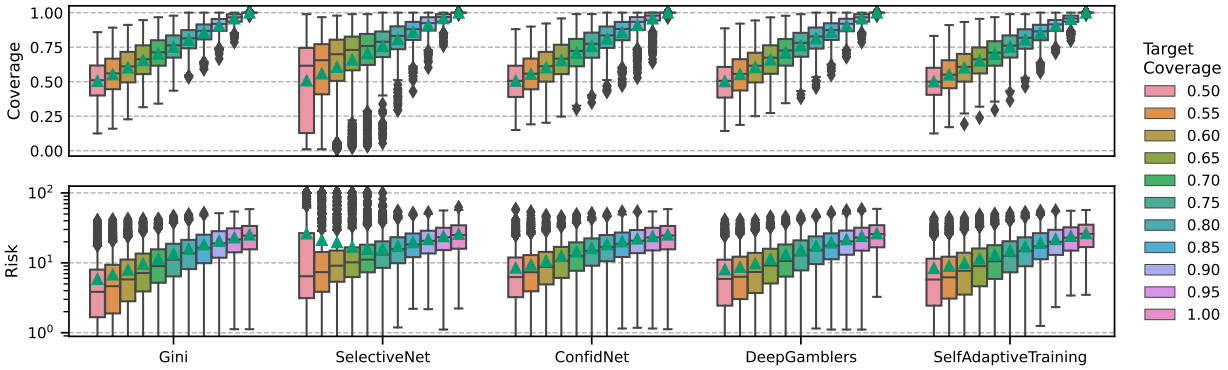
	Cov.	Gini	SelectiveNet	ConfidNet	DeepGamblers	SelfAdaptiveTraining
DenseNet-121 (CIFAR-10)	0.50	0.35 ± 0.1	0.34 ± 0.1	0.39 ± 0.1	0.89 ± 0.1	0.99 ± 0.3
	0.55	0.37 ± 0.1	0.39 ± 0.1	0.42 ± 0.1	0.92 ± 0.1	1.03 ± 0.3
	0.60	0.41 ± 0.1	0.40 ± 0.1	0.46 ± 0.1	0.98 ± 0.1	1.05 ± 0.3
	0.65	0.49 ± 0.1	0.54 ± 0.1	0.51 ± 0.1	1.05 ± 0.1	1.08 ± 0.3
	0.70	0.58 ± 0.1	0.62 ± 0.1	0.63 ± 0.1	1.09 ± 0.1	1.14 ± 0.2
	0.75	0.73 ± 0.1	0.73 ± 0.1	0.75 ± 0.1	1.17 ± 0.1	1.25 ± 0.2
	0.80	0.98 ± 0.1	0.88 ± 0.1	0.99 ± 0.1	1.32 ± 0.1	1.36 ± 0.2
	0.85	1.39 ± 0.1	1.31 ± 0.2	1.36 ± 0.1	1.57 ± 0.1	1.57 ± 0.2
	0.90	2.25 ± 0.2	1.95 ± 0.2	2.25 ± 0.2	2.09 ± 0.2	2.14 ± 0.2
	0.95	3.70 ± 0.3	3.43 ± 0.2	3.72 ± 0.3	3.23 ± 0.2	3.34 ± 0.3
	1.00	5.89 ± 0.1	5.28 ± 0.1	5.89 ± 0.2	5.29 ± 0.2	5.38 ± 0.2
DenseNet-121 (CIFAR-100)	0.50	4.62 ± 0.4	7.90 ± 0.9	4.85 ± 0.6	7.95 ± 0.6	7.02 ± 1.3
	0.55	5.92 ± 0.5	9.28 ± 1.0	6.44 ± 0.7	8.94 ± 0.4	8.19 ± 1.3
	0.60	7.57 ± 0.5	10.17 ± 1.1	8.19 ± 0.9	10.14 ± 0.5	9.36 ± 1.2
	0.65	9.35 ± 0.6	10.90 ± 0.5	10.14 ± 0.9	11.38 ± 0.5	10.77 ± 1.1
	0.70	11.43 ± 0.6	12.62 ± 1.0	12.31 ± 0.8	12.79 ± 0.4	12.37 ± 1.0
	0.75	13.38 ± 0.6	13.97 ± 0.8	14.54 ± 0.8	14.31 ± 0.4	14.00 ± 0.8
	0.80	15.58 ± 0.6	15.99 ± 0.7	16.86 ± 0.8	15.93 ± 0.4	15.81 ± 0.6
	0.85	18.03 ± 0.6	17.78 ± 0.7	18.99 ± 0.6	17.72 ± 0.3	17.76 ± 0.5
	0.90	20.46 ± 0.4	20.16 ± 0.6	21.19 ± 0.4	19.66 ± 0.4	19.71 ± 0.5
	0.95	23.03 ± 0.4	22.36 ± 0.4	23.54 ± 0.3	21.94 ± 0.4	22.09 ± 0.4
	1.00	25.92 ± 0.4	24.52 ± 0.7	25.94 ± 0.4	24.28 ± 0.3	24.60 ± 0.3
DenseNet-121 (SVHN)	0.50	0.67 ± 0.1	0.67 ± 0.1	3.88 ± 1.3	0.77 ± 0.1	0.80 ± 0.1
	0.55	0.68 ± 0.1	0.62 ± 0.1	3.92 ± 1.2	0.79 ± 0.1	0.81 ± 0.1
	0.60	0.70 ± 0.1	0.68 ± 0.1	3.98 ± 1.1	0.83 ± 0.1	0.82 ± 0.1
	0.65	0.72 ± 0.1	0.68 ± 0.1	4.04 ± 0.9	0.86 ± 0.1	0.83 ± 0.1
	0.70	0.74 ± 0.1	0.73 ± 0.1	4.09 ± 0.8	0.90 ± 0.1	0.86 ± 0.1
	0.75	0.77 ± 0.1	0.74 ± 0.1	4.18 ± 0.7	0.95 ± 0.1	0.90 ± 0.1
	0.80	0.81 ± 0.1	0.87 ± 0.1	4.25 ± 0.5	1.03 ± 0.1	0.95 ± 0.1
	0.85	0.90 ± 0.1	0.91 ± 0.1	4.33 ± 0.4	1.15 ± 0.1	1.06 ± 0.1
	0.90	1.13 ± 0.1	1.21 ± 0.1	4.46 ± 0.3	1.40 ± 0.2	1.29 ± 0.1
	0.95	1.89 ± 0.1	1.96 ± 0.1	4.64 ± 0.2	2.06 ± 0.1	2.02 ± 0.1
	1.00	3.98 ± 0.1	3.99 ± 0.1	4.94 ± 0.1	3.97 ± 0.1	3.94 ± 0.1

Table 3: Empirical selective risk in percentage for the classification benchmark with ResNet-34 for various target coverages over 10 runs.

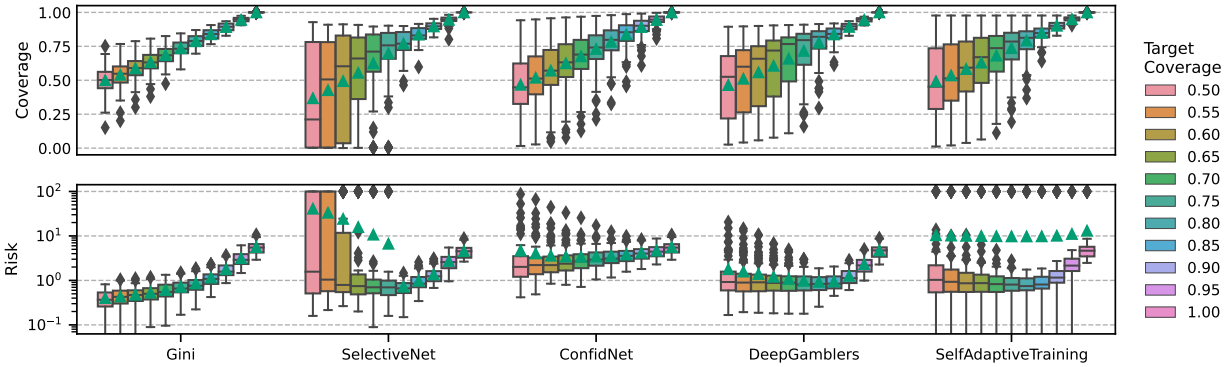
	Cov.	Gini	SelectiveNet	ConfidNet	DeepGamblers	SelfAdaptiveTraining
ResNet-34 (CIFAR-10)	0.50	0.27 ± 0.1	0.14 ± 0.1	0.39 ± 0.1	0.38 ± 0.1	0.60 ± 0.2
	0.55	0.29 ± 0.1	0.15 ± 0.0	0.40 ± 0.1	0.42 ± 0.1	0.61 ± 0.2
	0.60	0.31 ± 0.1	0.19 ± 0.1	0.42 ± 0.1	0.45 ± 0.1	0.62 ± 0.2
	0.65	0.34 ± 0.1	0.23 ± 0.1	0.45 ± 0.1	0.48 ± 0.1	0.63 ± 0.2
	0.70	0.37 ± 0.1	0.28 ± 0.1	0.47 ± 0.1	0.51 ± 0.1	0.65 ± 0.2
	0.75	0.45 ± 0.1	0.34 ± 0.0	0.54 ± 0.1	0.57 ± 0.1	0.69 ± 0.2
	0.80	0.53 ± 0.1	0.51 ± 0.1	0.62 ± 0.1	0.67 ± 0.1	0.82 ± 0.2
	0.85	0.74 ± 0.1	0.82 ± 0.2	0.81 ± 0.2	0.87 ± 0.1	1.04 ± 0.2
	0.90	1.25 ± 0.2	1.22 ± 0.3	1.28 ± 0.2	1.34 ± 0.2	1.50 ± 0.2
	0.95	2.35 ± 0.2	2.56 ± 0.4	2.35 ± 0.2	2.47 ± 0.2	2.53 ± 0.3
	1.00	4.41 ± 0.2	4.46 ± 0.2	4.42 ± 0.2	4.50 ± 0.3	4.49 ± 0.2
ResNet-34 (CIFAR-100)	0.50	2.62 ± 0.2	4.42 ± 0.8	3.58 ± 0.3	3.57 ± 0.2	4.05 ± 0.4
	0.55	3.22 ± 0.3	5.32 ± 0.8	4.56 ± 0.4	4.31 ± 0.2	4.86 ± 0.4
	0.60	4.10 ± 0.2	6.18 ± 0.8	5.80 ± 0.5	5.40 ± 0.3	5.86 ± 0.5
	0.65	5.27 ± 0.2	7.61 ± 0.4	7.11 ± 0.5	6.87 ± 0.4	7.05 ± 0.6
	0.70	6.75 ± 0.3	9.28 ± 0.5	8.80 ± 0.5	8.40 ± 0.6	8.73 ± 0.7
	0.75	8.53 ± 0.5	11.22 ± 0.6	10.63 ± 0.5	10.25 ± 0.5	10.54 ± 0.7
	0.80	10.81 ± 0.5	12.68 ± 0.5	12.69 ± 0.6	12.18 ± 0.6	12.39 ± 0.8
	0.85	13.07 ± 0.6	14.55 ± 0.5	14.58 ± 0.5	14.37 ± 0.6	14.56 ± 0.7
	0.90	15.38 ± 0.7	16.95 ± 0.5	16.55 ± 0.4	16.58 ± 0.5	17.06 ± 0.6
	0.95	17.95 ± 0.6	18.99 ± 0.5	18.60 ± 0.5	18.91 ± 0.5	19.48 ± 0.6
	1.00	20.77 ± 0.5	21.58 ± 0.5	20.79 ± 0.5	21.63 ± 0.4	21.75 ± 0.4
ResNet-34 (SVHN)	0.50	0.39 ± 0.1	0.76 ± 0.3	1.96 ± 0.4	0.82 ± 0.1	8.74 ± 25.2
	0.55	0.42 ± 0.1	0.72 ± 0.1	2.13 ± 0.4	0.81 ± 0.1	8.74 ± 25.2
	0.60	0.47 ± 0.1	0.78 ± 0.2	2.34 ± 0.3	0.81 ± 0.1	8.74 ± 25.2
	0.65	0.52 ± 0.1	0.68 ± 0.1	2.54 ± 0.4	0.80 ± 0.1	8.74 ± 25.2
	0.70	0.60 ± 0.1	0.67 ± 0.1	2.76 ± 0.4	0.80 ± 0.1	8.74 ± 25.2
	0.75	0.70 ± 0.1	0.69 ± 0.1	2.98 ± 0.4	0.82 ± 0.1	8.75 ± 25.2
	0.80	0.83 ± 0.1	0.69 ± 0.1	3.20 ± 0.4	0.85 ± 0.1	8.76 ± 25.2
	0.85	1.11 ± 0.1	0.94 ± 0.1	3.52 ± 0.3	0.91 ± 0.1	8.82 ± 25.2
	0.90	1.68 ± 0.1	1.30 ± 0.1	3.93 ± 0.2	1.20 ± 0.1	9.11 ± 25.1
	0.95	2.97 ± 0.2	2.55 ± 0.2	4.40 ± 0.1	2.16 ± 0.2	10.02 ± 24.8
	1.00	5.31 ± 0.1	4.35 ± 0.1	5.31 ± 0.1	4.47 ± 0.2	11.96 ± 24.1



(a) VGG-16 (CIFAR-10)

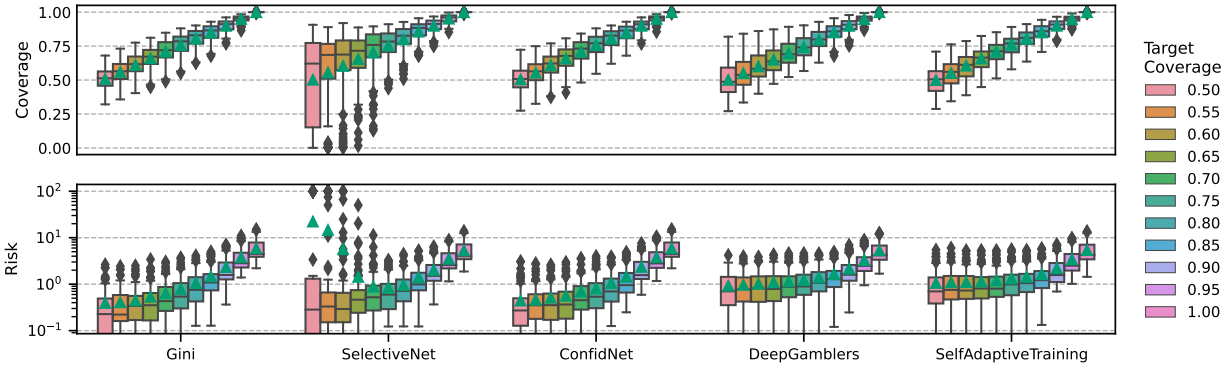


(b) VGG-16 (CIFAR-100)

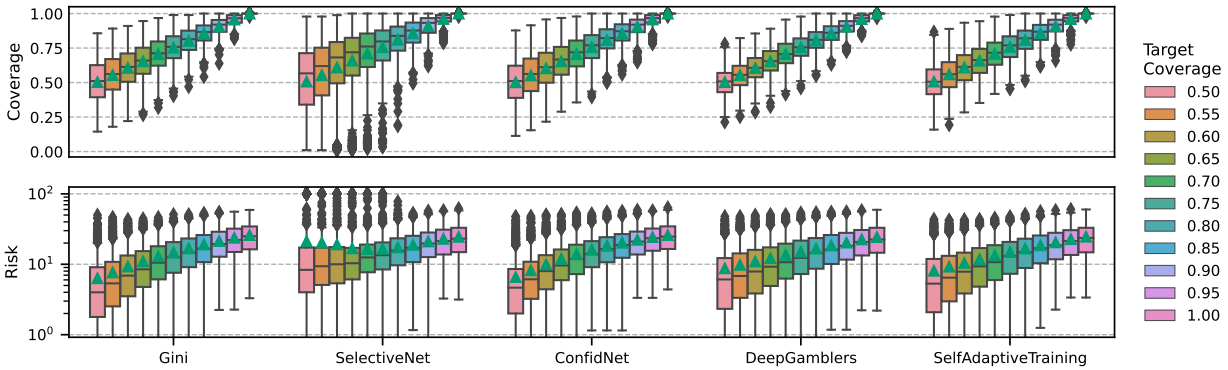


(c) VGG-16 (SVHN)

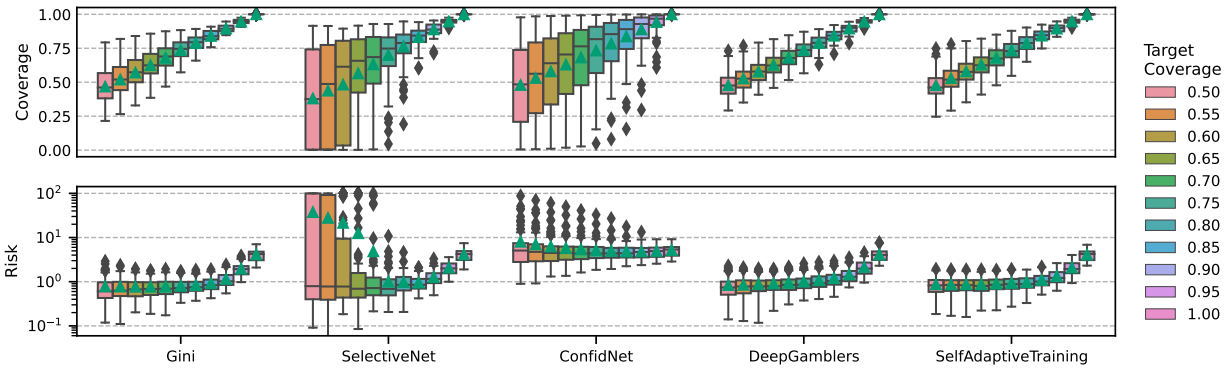
Figure 7: Class-wise calibrated coverage and risk versus target coverage for a VGG-16 model.



(a) DenseNet-121 (CIFAR-10)

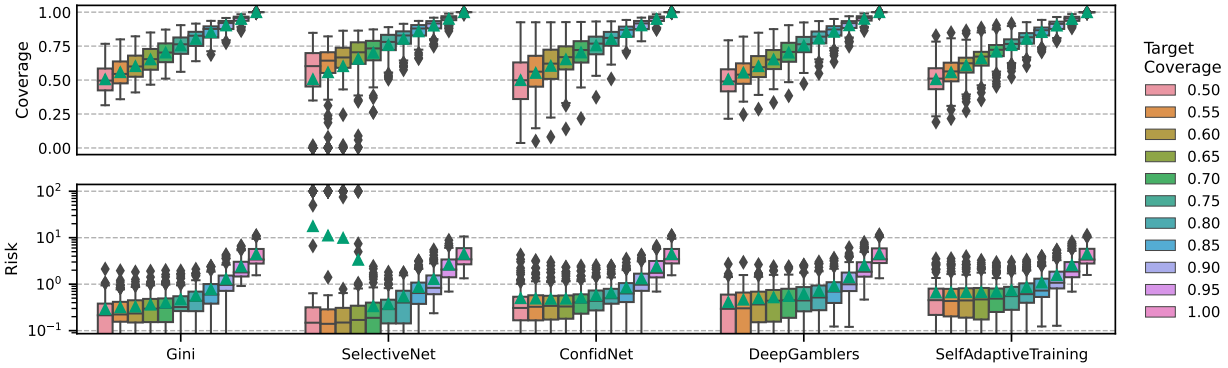


(b) DenseNet-121 (CIFAR-100)

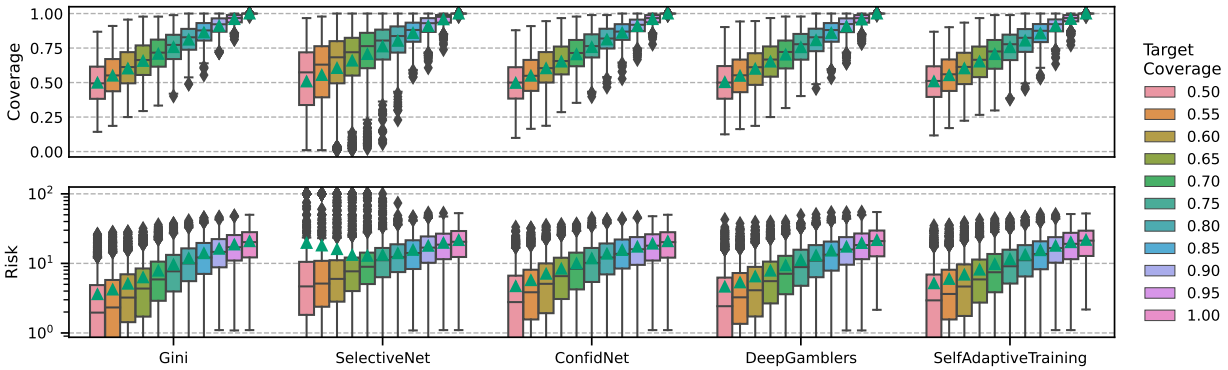


(c) DenseNet-121 (SVHN)

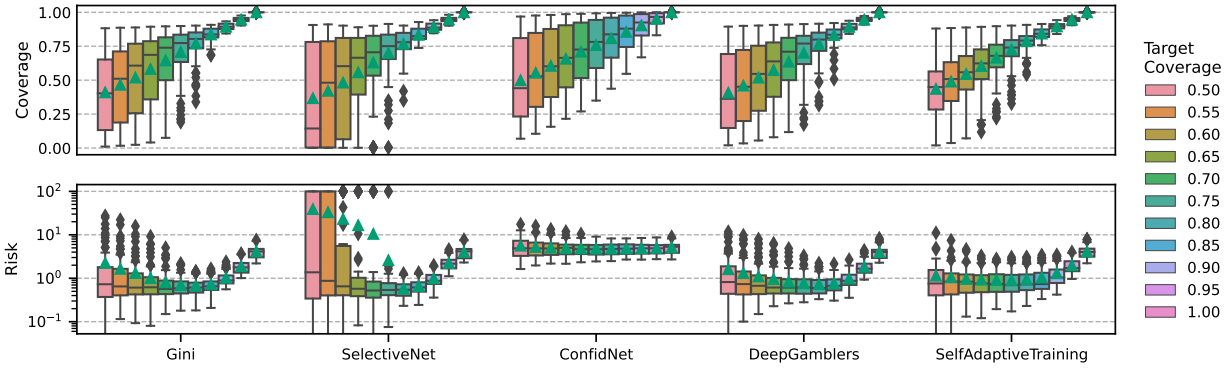
Figure 8: Class-wise calibrated coverage and risk versus target coverage for a DenseNet-121 model.



(a) ResNet-34 (CIFAR-10)

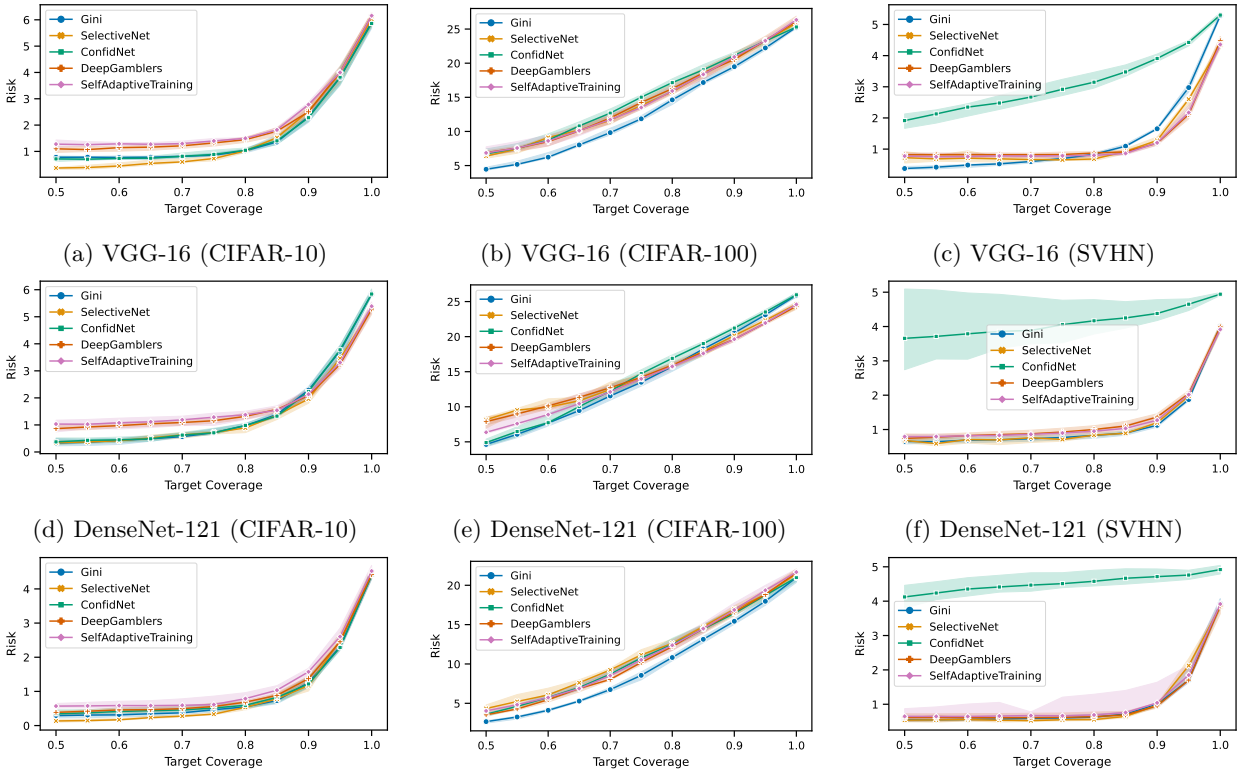


(b) ResNet-34 (CIFAR-100)



(c) ResNet-34 (SVHN)

Figure 9: Class-wise calibrated coverage and risk versus target coverage for a ResNet-34 model.

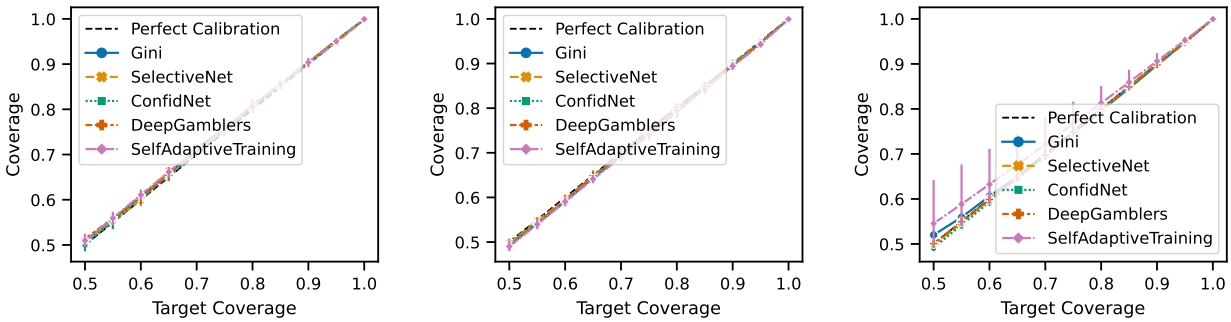


(a) VGG-16 (CIFAR-10) (b) VGG-16 (CIFAR-100) (c) VGG-16 (SVHN)

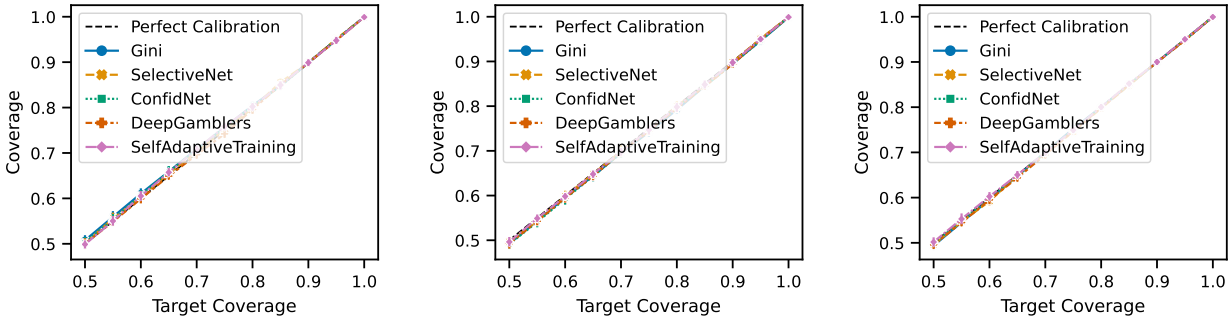
(d) DenseNet-121 (CIFAR-10) (e) DenseNet-121 (CIFAR-100) (f) DenseNet-121 (SVHN)

(g) ResNet-34 (CIFAR-10) (h) ResNet-34 (CIFAR-100) (i) ResNet-34 (SVHN)

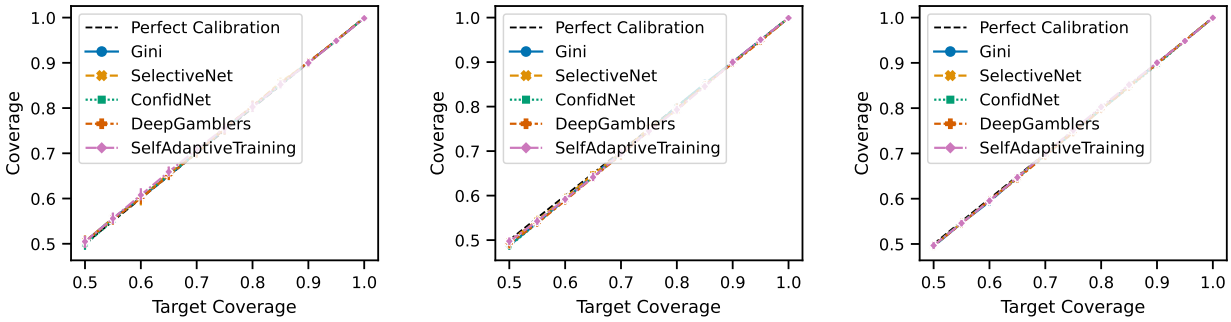
Figure 10: Global calibrated risk versus target coverage.



(a) VGG-16 (CIFAR-10) (b) VGG-16 (CIFAR-100) (c) VGG-16 (SVHN)



(d) DenseNet-121 (CIFAR-10) (e) DenseNet-121 (CIFAR-100) (f) DenseNet-121 (SVHN)



(g) ResNet-34 (CIFAR-10) (h) ResNet-34 (CIFAR-100) (i) ResNet-34 (SVHN)

Figure 11: Global calibrated coverage versus target coverage.