SCRIB: Set-classifier with Class-specific Risk Bounds for Blackbox Models

Zhen Lin[1], **Cao Xiao**[2], **Lucas Glass**[3], **M. Brandon Westover**[4], **Jimeng Sun**[1]
[1]University of Illinois at Urbana–Champaign, Urbana, IL, United States
[2] Amplitude, San Francisco, CA, United States
[3] Analytics Center of Excellence, IQVIA, Boston, MA, United States
[4] Massachusetts General Hospital, Boston, MA, United States
zhenlin4@illinois.edu, jimeng@illinois.edu

Abstract

Despite deep learning (DL) success in classification problems, DL classifiers do not provide a sound mechanism to decide when to refrain from predicting. Recent works tried to control the overall prediction risk with *classification with rejection options*. However, existing works overlook the different significance of different classes. We introduce Set-classifier with Class-specific RIsk Bounds (SCRIB) to tackle this problem, assigning multiple labels to each example. Given the output of a black-box model on the validation set, SCRIB constructs a set-classifier that controls the *class-specific* prediction risks. The key idea is to reject when the set classifier returns more than one label. We validated SCRIB on several medical applications, including sleep staging on electroencephalogram (EEG) data, X-ray COVID image classification, and atrial fibrillation detection based on electrocardiogram (ECG) data. SCRIB obtained desirable class-specific risks, which are 35%-88% closer to the target risks than baseline methods.

1 Introduction

Deep Learning (DL) has demonstrated highly discriminative power on classification tasks and has been successfully applied in many application areas, including healthcare(Hannun et al. 2019; Esteva et al. 2017; Gulshan et al. 2016; Biswal et al. 2018).

Impressive as DL is, we nevertheless hope to identify when the model might fail and take actions accordingly, which is especially important in healthcare applications. For example, suppose we are to design an automated system using pretrained DL classifiers for sleep staging on EEG data (Biswal et al. 2018), detecting diseases based on ECG data (Hong et al. 2019), or classifying X-ray images (Qiao et al. 2020). For predictions to be reliable, the model should sometimes reject the examples and yield them to human experts to decide. And when the model does predict, we want the misclassification risks to be low and controllable.

This leads to classification with a reject option, where the rejection usually happens when the confidence score is low. For example, when the base classifier's prediction is the true conditional probability, Maximum Class Probability (MCP) is the optimal confidence score as it minimizes the rejection rate for each risk level (Chow 1970). The actual decision rule, given an overall risk target (not a class-specific one), (Geifman and El-Yaniv 2017) picks a confidence threshold on the validation set. Alternative confidence measures were also proposed by training separate models (Jiang et al. 2018; Corbière et al. 2019).

However, existing works ignored that different classes have different significance. The confidence score is almost always class-agnostic, and the rejection is binary, which means there is no class-specific risk control. As a result, a difficult class can have an extremely high rejection rate, where easy classes are predicted all the time. In many applications such as medicine, this class-agnostic rejection creates problems, as difficult classes are often the most important ones that need classification. For example, the N1 class in sleep staging is challenging to classify but of great interest to the applications. It will currently be disproportionally rejected due to the difficulty of achieving low overall risk.

In this work, we aim to incorporate class-specific risk controls into classification with rejection. We propose Set-classifier with Class-specific RIsk Bounds (SCRIB), which can output multiple labels to each example based on the predicted conditional probabilities by a black-box classifier in a theoretically efficient way. Rejections happen naturally when the output set contains more than one label[1]. The multiple labels for each rejection also serve as an intuitive explanation of the underlying ambiguity, helping human experts understand the model behavior.

To construct the set classifier, SCRIB searches for the optimal thresholds by minimizing a loss designed to control class-specific risks. This set classifier can be optimal in some scenarios and naturally comes with a risk concentration bound. To the best of our knowledge, SCRIB is the first class-specific risk control method for multi-class classification tasks. SCRIB has the following desirable properties:

1. **Flexible**. It enforces class-specific risk controls by allowing different risk targets for different classes. It also works with any black-box classifier without model retraining.
2. **Fast**. We propose an efficient optimization method to choose the thresholds for the set classifier. Specifically, we proposed a novel dynamic programming-based coordinate

[1]We tackle *multi-class classification* where one class is assigned to each example. This is different from *multi-label classification* where multiple labels can be assigned to the same example.

descent method for this optimization task, which proves highly efficient.

3. **Concise**. When it rejects, it returns a set of possible labels, as the rejection explanation (Section 5.4) without unnecessary labels (Theorem 4.1).

Finally, we evaluated SCRIB on multiple real-world medical datasets. SCRIB obtained desirable class-specific risks (usually within 1% of the target risks), which are 35%-88% closer to the target risks than baseline methods (Table 3).

2 Related Works

The most related line of works is a *classification with rejection options*, which is intertwined with two other areas: *calibration* and *uncertainty quantification* (UQ). At a high level, classification with rejection options is about making classification decisions using a trained classifier, while calibration and uncertainty quantification enhance the classifier's prediction scores.

A natural method rejects if the prediction score (or uncertainty measure) is below (or above) a certain threshold. In terms of the scores, one simple choice is the predicted class probabilities by the base classifier. Many works directly use the predicted Maximum Class Probability (MCP) [2] (Geifman and El-Yaniv 2017; Gimpel 2017), which is already optimal for overall risk control if the prediction is accurate (Chow 1970). In this respect, *calibration* research (Platt and others 1999; Guo et al. 2017; Wenger, Kjellström, and Triebel) 2020; Kull et al. 2019; Kumar, Liang, and Ma 2019) is thus related as they aim to transform the classifier output to true probabilities. However, *calibration* research is orthogonal to our problem - our work focuses on the decision (rejection) rules and does not require calibrated outputs.

Measures other than predicted probabilities have also been explored. In classification, uncertainty is almost a synonym to (the opposite of) confidence, and such research is related to *uncertainty quantification*. Monte-Carlo Dropout (MC-Dropout) (Gal and Ghahramani 2016) is one of the most popular uncertainty quantification methods because it is relatively lightweight. MCDropout was used in rejection literature (Geifman and El-Yaniv 2017; Corbière et al. 2019). However, most related methods (including MCDropout) (Neal 1996; Gal and Ghahramani 2016; Blundell et al. 2015; Wilson et al. 2016; Lakshminarayanan, Pritzel, and Blundell 2017; Moon et al. 2020; Corbière et al. 2019) need to simultaneously train the base classifier and confidence/uncertainty estimator, which greatly limits the applicability and might even affect the performance, especially when the base classifier is a complicated deep learning model. An exception is (Jiang et al. 2018), which trains a second classifier but is very expensive and only works for low-dimensional data. Works in *uncertainty quantification* are still complementary to our problem because uncertainty measures are inputs to the rejection rules, which will be demonstrated in our experiments.

Almost all score-based rejection works focus on finding better confidence measures and (Geifman and El-Yaniv 2017; Fumera, Roli, and Giacinto 2000) focus on decision rules (e.g., threshold finding). Apart from confidence-based rejection, a good number of works jointly learn the classifier and the rejector without an explicit confidence score at all—(Fumera and Roli 2002; Wegkamp and Yuan 2011; Grandvalet et al. 2009; Bartlett and Wegkamp 2008; Herbei and Wegkamp 2006; Cortes, De Salvo, and Mohri 2016; Cortes, DeSalvo, and Mohri 2016; Geifman and El-Yaniv 2019), many of which focusing on binary classification and SVM. Such methods also tend to have limited applicability and do not work for blackbox classifiers.

Most importantly, all works reviewed here focus on *overall* risk. To the best of our knowledge, our work is the first to find decision rules for class-specific risk controls given a blackbox classifier.

A secondary issue of existing works is that the rejection is typically a binary decision. When rejection happens, we only know that the most likely class is selected or rejected. On the contrary, when our set-classifier rejects (i.e., when it contains more than one label), it informs the human inspector what competing predictions are causing the rejection given our risk targets (Section 5.4). Set classifiers have also recently been applied in UQ for DL (Angelopoulos et al. 2021), but to the best of our knowledge, no such application considers the rejection possibility nor class-specific risk targets. Such problem (i.e. constructing confidence sets without rejection) is however well-studied with-closed form solution (Mortier et al. 2019; Sadinle, Lei, and Wasserman 2019), but set-classifier in the context of rejection is a much more difficult problem.

3 Problem Formulation

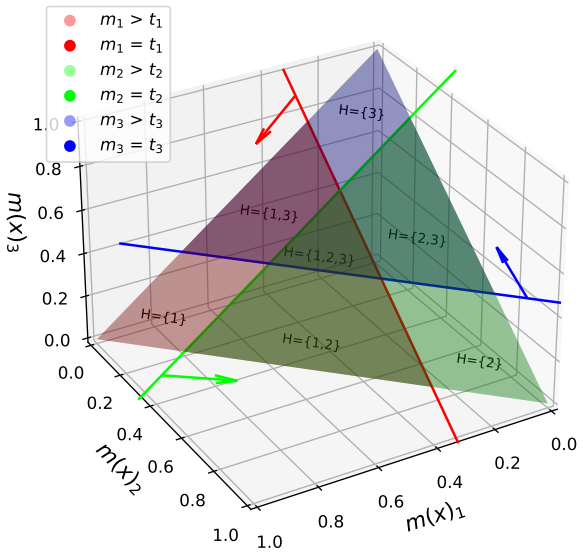

Figure 1: All possible set-valued predictions, with $K = 3$ and $\mathbf{t} = [0.25, 0.2, 0.3]$. Intuitively, K hyperplanes $(m_k(x) \geq \mathbf{t}_k)$ divide $\mathbb{R}^K$ into up to 2^K cells (some might be empty). In each cell, we show the value of the set classifier (E.g., $\mathbf{H} = \{1, 2\}$).

[2] In practice, usually MCP is replaced by the Maximum Softmax Response - the maximum Softmax output, as people tend to interpret Softmax output as probabilities.

Symbol	Meaning
k	Class index
$[K]$	The set $\{1, 2, \ldots, K\}$
$\mathcal{X}, \mathcal{Y}$	Data space and label space
$\mathbb{P}, \mathbb{P}_k$	Underlying data distribution (of class k)
$\mathbb{P}\{event\}$	Probability of $event$ when data follows $\mathbb{P}$
$\mathbf{1}\{event\}$	Indicator function of $event$
$\mathbf{H}$	Set classifier: $\mathcal{X} \mapsto 2^{\mathcal{Y}}$
$A(\mathbf{H})$	Ambiguity (Size- or Chance-) of $\mathbf{H}$
$r(\mathbf{H}), r_k(\mathbf{H})$	Risk of classifier $\mathbf{H}$ (of class k)
r^*, r_k^*	Target risks (overall/for class k)
$m_k(x)$	Base model prediction for $\mathbb{P}\{Y = k\|X = x\}$
$\mathbf{t}$,t_k	The threshold parameter for $\mathbf{H}$ (for class k)
$L(\mathbf{t})$	Unconstrained loss given thresholds $\mathbf{t}$
$\hat{L}, \hat{\mathbb{P}}, \hat{A}, \hat{r}, \hat{r}_k$	Empirical $L, \mathbb{P}, A, r, r_k$ on $\mathcal{S}_{\text{valid}}$
$\alpha_k(\mathbf{H})$	Mis-coverage rate for $\mathbf{H}$ of class k

Table 1: Notations used in this paper

3.1 Base Classifier and Learning Setup

In this work we situate our task in the K-class classification problem, with data space $\mathcal{X}$, label space $\mathcal{Y} = \{1, \ldots, K\}$, and the joint distribution of (X, Y) as $\mathbb{P}$ over $\mathcal{X} \times \mathcal{Y}$. We will use $[K]$ to denote the set $\{1, 2, \ldots, K\}$. We further denote the class-specific distributions $(X, Y = k)$ as $\mathbb{P}_k$, which is effectively a distribution over $\mathcal{X}$. Here $\mathbb{P}_k\{\cdot\}$ can also be viewed as $\mathbb{P}\{\cdot|Y = k\}$.

Like in many tasks, we assume the data split into a training set $\mathcal{S}_{\text{train}}$, validation set $\mathcal{S}_{\text{valid}}$ and test set $\mathcal{S}_{\text{test}}$. We will assume that data in $\mathcal{S}_{\text{valid}}$ and $\mathcal{S}_{\text{test}}$ are iid, and we can only use label information on $\mathcal{S}_{\text{train}}$ and $\mathcal{S}_{\text{valid}}$. We are given a model m (potentially a DNN) trained on $\mathcal{S}_{\text{train}}$: $\mathcal{X} \mapsto \mathbb{R}^K$, where the k-th output, $m_k(x)$, captures the conditional probability $\mathbb{P}\{Y = k|X = x\}$. As a simple example, $m(x)$ can be the Softmax output over the K classes or confidence scores generated from uncertainty quantification or calibration methods. Note that here $\mathcal{S}_{\text{valid}}$ is the validation set for this *base classifier* m, and will be used to tune SCRIB, as we will explain in Section 4). Finally, $\hat{\cdot}$ means evaluating the empirical value on $\mathcal{S}_{\text{valid}}$. For example, $\hat{\mathbb{P}}\{Y = 1\}$ means the frequency of class 1 in $\mathcal{S}_{\text{valid}}$.

3.2 Problem: Class-specific Risk Control

We will first introduce the concept of a set classifier:

Definition 1 (**Set Classifier**). *A set classifier is a mapping from data to a set of labels, denoted as* $\mathbf{H} : \mathcal{X} \mapsto 2^{\mathcal{Y}}$.

A set-valued function has been used in classification tasks for different purposes (sometimes under different names) (Wu, Lin, and Weng 2004; Vovk, Gammerman, and Shafer 2005; Del Coz, Díez, and Bahamonde 2009; Sadinle, Lei, and Wasserman 2019). Its link classification with rejection is natural: Rejections happen naturally when the set classifier contains more than one label. A set classifier is a generalization of the typical classifier that only outputs the most-likely class.

For a multi-class classification problem, ideally, we want an oracle classifier such that $\mathbb{P}\{\mathbf{H}_{oracle}(X) = \{Y\}\} = 1$. However, this is not possible in most cases. Our goal is to find a $\mathbf{H}$ that minimizes the *ambiguity* while satisfying *class-specific risk* constraints. There are many ways to define the ambiguity for a $\mathbf{H}$, and we will focus on two intuitive ones:

Definition 2 (**Chance-Ambiguity and Size-Ambiguity**). *Chance-Ambiguity of a set classifier* $\mathbf{H}$ *is the probability of it having cardinality (size) greater than 1, namely* $\mathbb{P}\{|\mathbf{H}(X)| > 1\}$. *Size-Ambiguity is the expected size of* $\mathbf{H}$, *namely* $\mathbb{E}[|\mathbf{H}(X)|]$

These two ambiguity definitions of a set classifier $\mathbf{H}$ are usually correlated[3]. Size-ambiguity, easier to analyze, is a measure used more often in the statistics literature (Sadinle, Lei, and Wasserman 2019). However, it overlooks the qualitative difference between being certain ($|\mathbf{H}| = 1$) and uncertain ($|\mathbf{H}| > 1$) - in reality, human experts need to step in as long as the model is uncertain, regardless of the size of $\mathbf{H}$. Chance-ambiguity is equivalent to the rejection rate widely used in rejection literature (Geifman and El-Yaniv 2017; Jiang et al. 2018; Corbière et al. 2019). We will use $A(\mathbf{H})$ to denote the general concept of ambiguity, either Chance-Ambiguity or Size-Ambiguity.

We define the overall risk like in existing rejection literature (Geifman and El-Yaniv 2017) and class-specific risks below:

Definition 3 (**Class-specific and Overall Risk**). *The class-specific risk* $r_k(\mathbf{H})$ *for class* k *and overall risk* $r(\mathbf{H})$ *for a set classifier* $\mathbf{H}$ *are defined to be:*

$$r_k(\mathbf{H}) := \mathbb{P}_k\{k \notin \mathbf{H}(X) \big| |\mathbf{H}| = 1\} \tag{1}$$

$$r(\mathbf{H}) := \mathbb{P}\{Y \notin \mathbf{H}(X) \big| |\mathbf{H}| = 1\} \tag{2}$$

Intuitively, $\mathbb{P}_k\{k \notin \mathbf{H}(X)\}$ means the probability of class k not in the output of set classifier $\mathbf{H}(X)$ (or formally the *mis-coverage rate* of class k). And $\mathbb{P}_k\{k \notin \mathbf{H}(X) \big| |\mathbf{H}| = 1\}$ is that probability of class k for the output set with a single label (i.e., no ambiguity).

Putting everything together, our goal is to solve the following optimization problem:

Problem 1 (**Class-specific Risk Control**).

$$\min_{\mathbf{H}} \quad A(\mathbf{H}) \tag{3}$$

$$\textit{s.t.} \quad \mathbb{P}_k\{k \notin \mathbf{H}(X) \big| |\mathbf{H}(X)| = 1\} \leq r_k^*, \forall k \in [K] \tag{4}$$

Here $A(\mathbf{H})$ is the ambiguity measure (Chance- or Size-Ambiguity, or a weighted average of both, chosen by the user depending on the task), and $r_1^*, \ldots, r_K^* \in [0, 1]$ are the user-specified risk targets.

4 The SCRIB Method

4.1 Method Overview

Given a trained base classifier m and validation set $\mathcal{S}_{\text{valid}}$, we first parameterize $\mathbf{H}$ with K thresholds, one for each class. Given $\mathbf{t} := (t_1, \ldots, t_K) \in \mathbb{R}^K$, $\mathbf{H}(x)$ is defined as

$$\mathbf{H}(x; \mathbf{t}) := \{k \in [K] : m_k(x) \geq t_k\}. \tag{5}$$

[3]Empirical results on the correlation are in the Appendix

Figure 1 illustrated an example when $K = 3$. Here $m_k(x)$ is a proxy for how confident the model thinks x is from class k. We will drop $\mathbf{t}$ in the notation for simplicity.

Next, we transform the optimization problem in Section 3.2 into an unconstrained optimization problem that minimizes the following loss $\hat{L} : \mathbb{R}^K \mapsto \mathbb{R}$:

$$\hat{L}(\mathbf{t}) := \underbrace{\hat{A}(\mathbf{H})}_{\text{ambiguity}} + \sum_{k=1}^{K} \underbrace{\lambda_k(\hat{r}_k(\mathbf{H}) - r_k^*)_+^2}_{\text{class specific risk control penalty}} \tag{6}$$

where $v_+ := \max\{v, 0\}$ and $\mathbf{H}$ is parameterized by $\mathbf{t}$ as defined above. $\hat{A}$ and $\hat{r}_k$ are the ambiguities and risks evaluated on the validation set $\mathcal{S}_{\text{valid}}$. In practice, all $\{\lambda_k\}_k$ are set to a large number unless there is an order of importance among classes.

Although the loss definition seems simple, the interaction between different classes makes it hard to simultaneously optimize all parameters. To tackle the actual optimization, we choose the thresholds $\mathbf{t}$ from the model's outputs on the validation set $\mathcal{S}_{\text{valid}}$. We compare several methods and arrive at an efficient coordinate descent algorithm with dynamic programming. A sketch of the full algorithm is provided in Algorithm 1, with the details of "QuickSearch" in the Appendix. In practice, we repeat Algorithm 1 ten times and take the lowest loss found.

Algorithm 1: Thresholds Finding for SCRIB

Input:
$\mathbf{M} \in \mathbb{R}^{N \times K}$: model output on $\mathcal{S}_{\text{valid}}$ sorted by column. $\mathbf{M}_{i,k}$ is the i-th smallest value in $\{m_k(x)\}_{x \in \mathcal{S}_{\text{valid}}}$, for class k. N denotes $|\mathcal{S}_{\text{valid}}|$.
$\hat{L} : \mathbb{R}^K \mapsto \mathbb{R}$, empirical loss function on $\mathcal{S}_{\text{valid}}$.
Output:
$\mathbf{t} \in \mathbb{R}^K$: optimal thresholds for the set classifier $\mathbf{H}$.
Algorithm:

```
For k ∈ [K], initialize t_k randomly from M_{·,k}, and evaluate current loss l ← L̂(t).
repeat
  for k = 1 to K do
    Fixing t_{k'} ∀k' ≠ k, search t'_k in M_{·,k} to minimize L̂ using QuickSearch (See Appendix)
    l'_k ← L̂(t'_k) where t'_k := (t_1, ..., t'_k, ..., t_K)
  end for
  If min_{k∈[K]} l'_k < l, update l ← l'_k and t ← t'_k
until l does not improve
return t
```

Complexity The naive search in each direction requires $O(N)$ loss evaluations, each taking $O(KN)$, leading to $O(KN^2)$ operations. We invented a novel dynamic programming trick in "QuickSearch", lowering it to only $O(KN)$ operations *in total* instead. Total complexity is thus $O(TK^2N)$ where T denotes the number of outer-iterations in Algorithm 1. Due to the space constraint, we have the pseudo-code for QuickSearch and comparison with several optimization methods (time and value) in the Appendix.

4.2 Parameterization Optimality

By parameterizing $\mathbf{H}$ using $\mathbf{t}$ as in Eq. (5), we are answering the question "Might x belong to class k?" for each class separately, as illustrated in Figure 1. This particular parameterization seems to "ignore" the potential interaction between classes. However, as we will prove next, $\mathbf{H}$ is already optimal in minimizing the mis-coverage rate. The *mis-coverage rate* for $\mathbf{H}$ for class k is defined as:

$$\alpha_k(\mathbf{H}) := \mathbb{P}_k\{k \notin \mathbf{H}(X)\} \tag{7}$$

It refers to the probability that the correct class k is not in the output of set classifier $\mathbf{H}(X)$.

Theorem 4.1. *(Adapted from (Sadinle, Lei, and Wasserman 2019)) For any $\mathbf{t}$, define $\mathbf{H}^*$ as the set classifier parameterized by $\mathbf{H}^*(x) := \{k : \mathbb{P}\{Y = k|X = x\} > t_k\}$. $\mathbf{H}^*$ has the minimum Size-Ambiguity among all set classifiers with equal or lower mis-coverage rates. That is, $\forall \mathbf{H}'$*

$$\big(\forall k, \alpha_k(\mathbf{H}') \le \alpha_k(\mathbf{H}^*)\big) \Leftrightarrow \mathbb{E}[|\mathbf{H}^*|] \le \mathbb{E}[|\mathbf{H}'|]$$

A proof using the Neyman-Pearson lemma (Neyman and Pearson 1933) is included in Appendix.

Usually, the base classifier m gives us some prediction scores (e.g., Softmax output). Un-calibrated prediction scores tend to deviate from true probabilities (Guo et al. 2017), but we only need order consistency like in (Geifman and El-Yaniv 2017). If the base classifier captures the ordering of $\mathbb{P}\{Y = k|X = x\}$, then with Theorem 4.1, our parameterization in Eq. (5) will give us an optimal $\mathbf{H}$ for minimizing Size-Ambiguity.

When the objective function contains Chance-Ambiguity, the form of $\mathbf{H}$ will depend on the distribution of the predictions (assuming they are true conditional probabilities). However, our proposed parameterization is still desirable because, empirically, Chance- and Size-Ambiguity are correlated, and this simple parameterization is also intuitive and less prone to over-fitting.

Secondary output: Another benefit of this parameterization is that for each output $\mathbf{H}$, we have the estimated mis-coverage rates $\alpha_1(\mathbf{H}), \ldots, \alpha_K(\mathbf{H})$ immediately[4]. Intuitively, the mis-coverage rate means $\mathbf{H}$ can miss class k with only probability $\alpha_k(\mathbf{H})$. As output, $\alpha_k(\mathbf{H})$ can be beneficial for human experts in classifying the rejected samples.

4.3 Risk Bounds

As mentioned in Section 4.1, the thresholds $\mathbf{t}$ are chosen by enforcing the risk constraints in Problem 1 on $\mathcal{S}_{\text{valid}}$. Roughly speaking, because m is not trained on the $\mathcal{S}_{\text{valid}}$ nor $\mathcal{S}_{\text{test}}$, if data in $\mathcal{S}_{\text{valid}}$ and $\mathcal{S}_{\text{test}}$ follow the same distribution, then the scores' distribution on $\mathcal{S}_{\text{valid}}$ for each class k can represent that at test time. We now derive a way to compute the class-specific risk bounds with the following theorem:

Theorem 4.2. *For any fixed set classifier $\mathbf{H}$ parameterized by $\mathbf{t}$, with a hold-out set $\{(X_i, Y_i)\}_{i=1}^N$ and a new test data X_{N+1} from $\mathbb{P}$, denote $k = Y_{N+1}$ as the true but unknown class of X_{N+1}, we have*

$$\mathbb{P}_k\{k \notin \mathbf{H}(X_{N+1}) \big| |\mathbf{H}(X_{N+1})| = 1\} = \mathbb{E}[\hat{r}_k(\mathbf{H})] = r_k(\mathbf{H}) \tag{8}$$

[4]This is given by the quantiles of the thresholds $\mathbf{t}$.

where $\hat{r}_k(\mathbf{H})$ is the risk on the first N data points and $r_k(\mathbf{H})$ is the true risk defined in Definition 3. Moreover, with (Hoeffding 1963) we have $\forall \epsilon > 0$:

$$\mathbb{P}\{\hat{r}_k(\mathbf{H}) \geq r_k(\mathbf{H}) + \epsilon\} \leq e^{-D(r_k(\mathbf{H})+\epsilon||r_k(\mathbf{H}))n_k} \quad (9)$$

where $D(p||q) = p \ln \frac{p}{q} + (1-p) \ln \frac{1-p}{1-q}$ is the Kullback–Leibler divergence between Bernoulli random variables parameterized by p and q, and $n_k := \sum_{i=1}^{N} \mathbf{1}\{Y_i = k\}\mathbf{1}\{|\mathbf{H}(X_i)| = 1\}$ denotes the number of data points from class k that receives a certain prediction by $\mathbf{H}$.

Proof for Theorem 4.2 is in Appendix. Note that since SCRIB chooses $\mathbf{t}$ on $\mathcal{S}_{\text{valid}}$, to compute the risk bounds, we technically cannot use $\mathcal{S}_{\text{valid}}$ as the hold-out set mentioned in Theorem 4.2, and need to reserve *another hold-out set*[5]. Since the bound decreases exponentially, the size of this hold-out set can be very small. In practice, since we have $|\mathcal{S}_{\text{valid}}| \gg K$, the bias from fitting $\mathbf{H}$ on $\mathcal{S}_{\text{valid}}$ is negligible: In our experiments, we found that $\hat{r}(\mathbf{H})$ on $\mathcal{S}_{\text{valid}}$ is very close to the true risk as measured on $\mathcal{S}_{\text{test}}$.

5 Empirical Results

We present relevant baselines to our task in Section 5.1. We will then compare these methods (when applicable) to SCRIB on a series of risk control tasks on synthetic and real-world datasets. The real-world datasets are with diverse characteristics but all from the medical domain, because we believe *classification with rejection* can have an important practical impact on that domain.

In the experiments, we aim to answer the following questions:

1. Can SCRIB control class-specific risks well empirically? (Section 5.3)
2. Does SCRIB also perform well for overall risk control? (Section 5.5) This experiment also serves as a test for our optimization method.

5.1 Baselines

We compare SCRIB with the following baselines.

- **Selective Guaranteed Risk (SGR)** (Geifman and El-Yaniv 2017) is a post-hoc method that can achieve an overall risk guarantee. The proposed version uses (predicted) Maximum Class Probability of the base classifier as the confidence score for rejection.
- **SGR + Dropout** (Geifman and El-Yaniv 2017) is a variant of SGR using the (negative) variance of Monte-Carlo Dropout (Gal and Ghahramani 2016) predictions as the confidence score.
- **LABEL** (Sadinle, Lei, and Wasserman 2019) is a set-classifier that can control the class-specific *mis-coverage rate* $\alpha_k(\mathbf{H})$, the unconditional version of $r_k(\mathbf{H})$. It is a conformal method that uses an analytical solution specific to $\alpha_k(\mathbf{H})$ (picking the α quantile of the prediction scores on the validation set).
- **SCRIB-** The same as SCRIB but we use the same threshold for all classes $t_k \equiv t$ for the same t. We include this to check the necessity of using multiple thresholds.

Compared with SGR, SCRIB can provide class-specific risk controls along with additional information (a confidence set) to human decision-makers when rejections happen. For the sake of our experiment, any method that uses one threshold (like SGR) is the same, and SGR is used only because it is one of the first to introduce rejection into deep learning. Compared with LABEL, SCRIB can control both the unconditional coverage level (as a degenerate use case, see Appendix) and the conditional risk when $|\mathbf{H}| = 1$. In addition, we want to emphasize that SCRIB can be applied to solve a lot of *more general* problems, with the specific optimization in Eq. (1) being just an instance. As an example, we will explain how SCRIB can be modified mildly to control overall risk in Experiment 5.5.

5.2 Data and Model Output

Synthetic data is created by first generating conditional probabilities and then sampling the labels from these probabilities. The synthetic data has 5 classes, with an easy class and a hard one. The exact generation process is in the Appendix.

ISRUC (Sub-group 1) (Khalighi et al. 2016) is a public Polysomnographic (PSG) dataset for sleep staging. It contains 89,283 30-second recordings from 100 subjects, classified into W/N1/N2/N3/REM (class 0-4). 75% of the data are used to train the base classifier.

Sleep-EDF (Kemp et al. 2000; Goldberger et al. 2000) is another public dataset widely used to evaluate sleep staging models. It has the same classes as ISRUC, and we use 122 recordings (331,184 samples) to train the base model.

ECG (PhysioNet2017) (Clifford et al. 2017; Goldberger et al. 2000) is a public ECG dataset with 8,528 de-identified ECG recordings sampled at 300Hz. Classes 0-3 are Normal (N), Other rhythms (O), Atrial Fibrillation (AF), and Noisy. 75% of the recordings were used for training the base classifier.

X-ray dataset is constructed from two publicly available sources, COVID Chest X-ray[6] and Kaggle Chest X-ray[7], including 5,508 chest X-ray images from 2,874 patients. Class 0-3 are COVID-19, non-COVID-19 viral pneumonia, bacterial pneumonia, and normal.

Excluding samples for model training, each class's sample counts are presented in Table 2. These are evenly split into validation and test sets.

Base Deep Learning Models For ISRUC and Sleep-EDF, we used a ResNet-based (He et al. 2016) with 3 Residual Blocks, each with 2 convolution layers. For ECG, we employed (Hong et al. 2019) and changed the last layer for a 4-classification problem. For X-ray, we directly take the DL model predictions (Qiao et al. 2020) and run experiments in a purely post-hoc manner. More training details are in the Appendix.

[5]This practice is similar to (LeRoy and Zhao 2021), which uses three hold-out sets.

[6]https://github.com/ieee8023/covid-chestxray-dataset

[7]https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia

Data \Class	0	1	2	3	4
Xray	34	547	1,002	400	N/A
ISRUC	4,907	2,857	7,255	4,476	2,985
SleepEDF	57,424	4,464	14,812	1,946	5,259
ECG	2,893	1,579	449	145	N/A

Table 2: Total sample counts for each class of the validation dataset (excluding the training samples used to train the DL model).

5.3 Experiment: Class-Specific Risks

Figure 2: The issue of unbalanced class-specific risks on ISRUC (left) and Sleep-EDF (right). Controlling only the overall risk at 5% leads to extremely high risks for harder classes like N1 (SGR). Setting class-specific risk constraints, SCRIB achieves much lower risk on N1 than baselines. (Here, the overall risk for SGR/LABEL/SCRIB are 4.8%/11%/7.6% for ISRUC, and 4.5%/6.8%/4.4% for Sleep-EDF.)

In this experiment, we check whether SCRIB can find a $\mathbf{H}$ with a risk profile similar to a set of pre-specified values. In many healthcare-related tasks, some classes are (much) harder to classify than others. For example, for sleep staging, N1 is usually the hardest-to-predict class, whereas W (wake) is usually easy, which means the risks are very high/low risk for N1/W. Figure 2 illustrates this observation and shows how to alleviate this issue with SCRIB and class specific risk targets.

Setup: To quantitatively compare different methods, we will set the target risks (r_k^*) for SCRIB to 15% for all classes for ECG and 10% for other datasets. The same numbers are used as overall risk targets (r^*) for SGR and mis-coverage targets for LABEL. Note requiring *all* classes to bear 10% risk is a much stricter condition than requiring the *overall* risk to be the same number. The target is higher for ECG because the performance of the classifier is worse (SGR rejects 90+% samples at $r^* = 15\%$). λ_k is set to 10^4 for all classes and datasets, and we use chance-ambiguity for $A(\mathbf{H})$. We choose large λ_ks to satisfy the risk constraint before optimizing ambiguities (see Eq. 6)[8].

Evaluation Metric: We will measure the excess class-specific risk defined as

$$(\Delta r_k)_+ := \max\{0, r_k(\mathbf{H}) - r_k^*\} \tag{10}$$

[8]In fact, 10^4 is not that large: 1% excess risk translates to $10^4(1\%)^2 = 1.0$ (the second term in Eq. 6), while the ambiguity term (the first term) is a value in $[0, 1]$

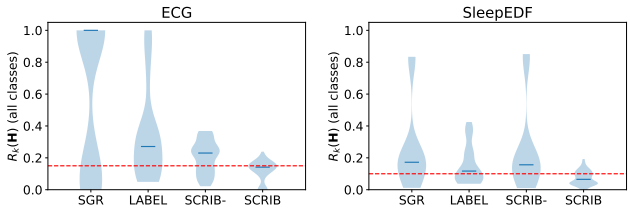

Figure 3: Distribution (violin plots) of class-specific risks for different methods with marked medians. Unlike other methods, realized class-specific risks of SCRIB are concentrated around/below target (red dashed lines).

$(\Delta r_k)_+$ (%)	SGR	LABEL	SCRIB-	SCRIB
Xray	5.67	6.43	13.69	3.71 (0.34)
ISRUC	8.60	4.23	8.79	**1.78** (0.01)
SleepEDF	16.90	7.21	16.32	**0.89** (3e-8)
ECG	46.23	21.58	7.71	**0.90** (6e-12)

Table 3: Average class specific excess risk ($(\Delta r_k)_+$ in percentage) for each methods. The p-values for the two-sample mean t-test between SCRIB and the best baseline are reported in parenthesis. SCRIB directly controls the risk and has lower deviations from targets than all baselines.

on the test set. For binary rejection like SGR, $\mathbf{H}_{SGR}(x)$ is naturally defined to be $[K]$ when rejections happen. We randomly split the samples into validation and test sets 20 times, and report mean and p-values. For ISRUC/SleepEDF/ECG, we include the results of re-splitting by recordings/subjects in the Appendix.

Results are presented in Table 3 and Figure 3. The runtime of SCRIB is detailed in the Appendix, which is generally a few seconds. SCRIB almost always controls the class-specific risks close to the target. Except for the X-ray dataset, the difference between SCRIB and the best baseline is always significant. This can also be seen from the violin-plots as well. For the X-ray dataset, the risks are much more volatile as each class size is small, especially after rejection. Comparison between SCRIB and SCRIB- suggests that using the same threshold for all classes is not enough even with the custom loss function.

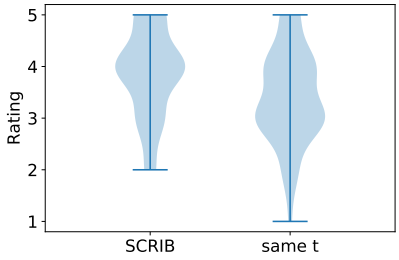

Figure 4: Distribution of ratings given by doctor. SCRIB (left) has higher ratings than using the same threshold for all classes (right).

5.4 Clinical User Study of Set Predictions

To evaluate the practical value and interpretability of a set classifier, we picked 50 samples from the ISRUC dataset[9] and asked a neurologist with a specialization in sleep medicine to score the predicted sets from 1 to 5 (with 5 being the best). The sets get lower scores if they miss a likely class or unnecessarily ambiguous (e.g., contain all labels all the time). We compare the scores with a baseline that uses the same t for all classes like SCRIB- and SGR, where t is chosen to have the same number of certain predictions as SCRIB. On average, SCRIB's score is significantly higher with p-value 0.01 (3.86 ± 0.86 vs 3.42 ± 0.91).

5.5 Experiment: Overall Risk

This experiment focuses on comparing the overall risk control between SCRIB and the baseline method SGR. The first goal is to explain how to slightly change the loss function of SCRIB for a different task, such as the overall risk control SGR was designed for. Moreover, because we know the analytic solution when the predicted probabilities are accurate, this experiment also serves as a sanity check to see whether the searched local optima are good (close to global optima).

Setup: We will use SCRIB to solve the overall risk control SGR was designed for, by changing the loss function to account for chance-ambiguity and the overall risk:

$$\hat{L}_{overall}(\mathbf{t}) := \underbrace{\hat{\mathbb{P}}\{|\mathbf{H}(X)| > 1\}}_{\text{Chance-ambiguity}} + \underbrace{\lambda(\hat{r}(\mathbf{H}) - r^*)_+^2}_{\text{Overall risk penalty}} \quad (11)$$

Setting all thresholds to the same gives the best trade-off when the base classifier is accurate, but we do not impose this prior knowledge. Therefore, an inferior search could find bad local optima/trade-offs for SCRIB because it picks K different thresholds. We repeat the experiment 20 times, each time randomly re-splitting unseen data evenly into validation and test sets. For ISRUC/SleepEDF/ECG, data for the same recording are always in the same set. λ is set to 10^4 like before.

Evaluation Metric: We will plot accuracy ($1-$ risk) as a function of coverage / chance-ambiguity and compute the area under the curve (AUC) for SGR and SCRIB, a common evaluation metric in classification with rejection literature. When the model output is the true conditional probability, using the same threshold t for all classes is theoretically optimal. As a result, we expect the SGR curve to be slightly above SCRIB for the Synthetic data (i.e., lower ambiguity with the same risk).

Results are presented in Table 4 and Figure 5. In general, SCRIB is on par with or better than SGR in our benchmark datasets. Although SGR is the theoretical optimal on the Synthetic data, the performance difference between SCRIB and SGR is small. This is also the case for Xray, but for the rest of the data, we see that SCRIB has the best trade-off[10]. SGR+Dropout is comparable with SGR. On ECG, the confidence given by MCDropout is negatively correlated with accuracy, which prevents SGR from controlling the overall risk, so we omit those results[11].

[9]For each class, we pick the most certain instance according to the base classifier, 3 instances at the 100%/90%/80% percentile for entropy, and 6 purely random instances.

[10]This is a known phenomenon (Fumera, Roli, and Giacinto 2000) and can happen if the base classifier has biases for a different class. But the focus of this experiment is that our search algorithm finds good local optima.

[11]There is no curve because it can never find a threshold such that data *above* that threshold have a low risk. Similar phenomena have been noted before (Jiang et al. 2018)

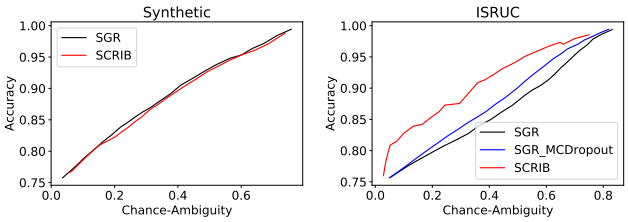

Figure 5: Accuracy-Ambiguity (reject rate) curve when we use different methods to control the overall risk. SCRIB achieves similar or higher accuracy at the same level of ambiguity as SGR and its variant.

AUC (1e-2)	SGR	SGR+Dropout	SCRIB
Synthetic	**90.12**±0.43	N/A	89.89±0.43
Xray	89.39±0.54	N/A	89.32±0.56
ISRUC	87.55±0.71	88.60±0.56	**90.77**±0.74
SleepEDF	96.50±0.60	96.48±0.51	96.62±0.65
ECG	77.03±2.13	N/A	**82.55**±0.67

Table 4: Mean and standard deviation of AUC of the accuracy-ambiguity curve for different methods ($n = 20$ experiments). Statistically significant differences (at p=0.01) are bolded. AUC of SCRIB is either comparable with SGR or higher.

6 Conclusion

In this paper, we present SCRIB, the first method for classification with rejection with class-specific risk controls. SCRIB provides a simple, effective and efficient way to construct set-classifiers for this task by choosing multiple thresholds for the base classifier's output. We demonstrated how overall risk control leads to the issue of unbalanced risks for different classes. Then, we showed that SCRIB can control the class-specific risks close to the targets on several medical datasets. Since this is a new and important new task, we also see a lot of potential research directions, examples of which include finding more efficient optimization method or alternative parameterization of the rejection criteria that is still theoretically sound. We believe that, as the first method in controlling class-specific risk in classification with rejection, SCRIB has potential applications to other fields where class-specific risks matter as well.

References

Angelopoulos, A.; Bates, S.; Malik, J.; and Jordan, M. I. 2021. Uncertainty Sets for Image Classifiers using Conformal

Prediction. arXiv:2009.14193.

Bartlett, P. L.; and Wegkamp, M. H. 2008. Classification with a reject option using a hinge loss. *Journal of Machine Learning Research*.

Biswal, S.; Sun, H.; Goparaju, B.; Westover, M. B.; Sun, J.; and Bianchi, M. T. 2018. Expert-level sleep scoring with deep neural networks. *J. Am. Med. Inform. Assoc.*

Blundell, C.; Cornebise, J.; Kavukcuoglu, K.; and Wierstra, D. 2015. Weight uncertainty in neural networks. In *32nd International Conference on Machine Learning, ICML 2015*. ISBN 9781510810587.

Chow, C. K. 1970. On Optimum Recognition Error and Reject Tradeoff. *IEEE Transactions on Information Theory*.

Clifford, G. D.; Liu, C.; Moody, B.; Lehman, L. H.; Silva, I.; Li, Q.; Johnson, A. E.; and Mark, R. G. 2017. AF classification from a short single lead ECG recording: The PhysioNet/-computing in cardiology challenge 2017. In *Computing in Cardiology*.

Corbière, C.; Thome, N.; Bar-Hen, A.; Cord, M.; and Pérez, P. 2019. Addressing Failure Prediction by Learning Model Confidence.

Cortes, C.; De Salvo, G.; and Mohri, M. 2016. Boosting with abstention. In *Advances in Neural Information Processing Systems*.

Cortes, C.; DeSalvo, G.; and Mohri, M. 2016. Learning with rejection. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. ISBN 9783319463780.

Del Coz, J. J.; Díez, J.; and Bahamonde, A. 2009. Learning nondeterministic classifiers. *Journal of Machine Learning Research*.

Esteva, A.; Kuprel, B.; Novoa, R. A.; Ko, J.; Swetter, S. M.; Blau, H. M.; and Thrun, S. 2017. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542: 115.

Fumera, G.; and Roli, F. 2002. Support vector machines with embedded reject option. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. ISBN 354044016X.

Fumera, G.; Roli, F.; and Giacinto, G. 2000. Reject option with multiple thresholds. *Pattern Recognition*.

Gal, Y.; and Ghahramani, Z. 2016. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *33rd International Conference on Machine Learning, ICML 2016*. ISBN 9781510829008.

Geifman, Y.; and El-Yaniv, R. 2017. Selective classification for deep neural networks. In *Advances in Neural Information Processing Systems*.

Geifman, Y.; and El-Yaniv, R. 2019. SelectiveNet: A deep neural network with an integrated reject option. In *36th International Conference on Machine Learning, ICML 2019*. ISBN 9781510886988.

Gimpel, K. 2017. A Baseline for Detecting Misclassified Out-of-Distribution Examples. *Iclr*.

Goldberger, A. L.; Amaral, L. A.; Glass, L.; Hausdorff, J. M.; Ivanov, P. C.; Mark, R. G.; Mietus, J. E.; Moody, G. B.; Peng, C. K.; and Stanley, H. E. 2000. PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals. *Circulation*.

Grandvalet, Y.; Rakotomamonjy, A.; Keshet, J.; and Canu, S. 2009. Support vector machines with a reject option. In *Advances in Neural Information Processing Systems 21 - Proceedings of the 2008 Conference*. ISBN 9781605609492.

Gulshan, V.; Peng, L.; Coram, M.; Stumpe, M. C.; Wu, D.; Narayanaswamy, A.; Venugopalan, S.; Widner, K.; Madams, T.; Cuadros, J.; Kim, R.; Raman, R.; Nelson, P. C.; Mega, J. L.; and Webster, D. R. 2016. Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs. *JAMA*, 316(22): 2402–2410.

Guo, C.; Pleiss, G.; Sun, Y.; and Weinberger, K. Q. 2017. On calibration of modern neural networks. In *34th International Conference on Machine Learning, ICML 2017*. ISBN 9781510855144.

Hannun, A. Y.; Rajpurkar, P.; Haghpanahi, M.; Tison, G. H.; Bourn, C.; Turakhia, M. P.; and Ng, A. Y. 2019. Cardiologist-level Arrhythmia Detection and Classification in Ambulatory Electrocardiograms using a Deep Neural Network. *Nature medicine*, 25(1): 65.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. ISBN 9781467388504.

Herbei, R.; and Wegkamp, M. H. 2006. Classification with reject option. *Canadian Journal of Statistics*.

Hoeffding, W. 1963. Probability Inequalities for Sums of Bounded Random Variables. *Journal of the American Statistical Association*.

Hong, S.; Xiao, C.; Ma, T.; Li, H.; and Sun, J. 2019. Mina: Multilevel knowledge-guided attention for modeling electrocardiography signals. In *IJCAI International Joint Conference on Artificial Intelligence*. ISBN 9780999241141.

Jiang, H.; Kim, B.; Gupta, M.; and Guan, M. Y. 2018. To trust or not to trust a classifier. In *Advances in Neural Information Processing Systems*.

Kemp, B.; Zwinderman, A. H.; Tuk, B.; Kamphuisen, H. A.; and Oberyé, J. J. 2000. Analysis of a sleep-dependent neuronal feedback loop: The slow-wave microcontinuity of the EEG. *IEEE Transactions on Biomedical Engineering*.

Khalighi, S.; Sousa, T.; Santos, J. M.; and Nunes, U. 2016. ISRUC-Sleep: A comprehensive public dataset for sleep researchers. *Computer Methods and Programs in Biomedicine*.

Kull, M.; Perello-Nieto, M.; Kängsepp, M.; Filho, T. S.; Song, H.; and Flach, P. 2019. Beyond temperature scaling: Obtaining well-calibrated multiclass probabilities with Dirichlet calibration.

Kumar, A.; Liang, P.; and Ma, T. 2019. Verified uncertainty calibration.

Lakshminarayanan, B.; Pritzel, A.; and Blundell, C. 2017. Simple and scalable predictive uncertainty estimation using

deep ensembles. In *Advances in Neural Information Processing Systems*.

LeRoy, B.; and Zhao, D. 2021. MD-split+: Practical Local Conformal Inference in High Dimensions. arXiv:2107.03280.

Moon, J.; Kim, J.; Shin, Y.; and Hwang, S. 2020. Confidence-Aware Learning for Deep Neural Networks.

Mortier, T.; Wydmuch, M.; Hüllermeier, E.; Dembczynski, K.; and Waegeman, W. 2019. Efficient Algorithms for Set-Valued Prediction in Multi-Class Classification. *CoRR*, abs/1906.08129.

Neal, R. 1996. Bayesian Learning for Neural Networks. *LECTURE NOTES IN STATISTICS -NEW YORK- SPRINGER VERLAG-*.

Neyman, J.; and Pearson, E. S. 1933. IX. On the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 231(694-706).

Platt, J.; and others. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*.

Qiao, Z.; Bae, A.; Glass, L. M.; Xiao, C.; and Sun, J. 2020. FLANNEL (Focal Loss bAsed Neural Network EnsembLe) for COVID-19 detection . *Journal of the American Medical Informatics Association*.

Sadinle, M.; Lei, J.; and Wasserman, L. 2019. Least Ambiguous Set-Valued Classifiers With Bounded Error Levels. *Journal of the American Statistical Association*, 114(525).

Vovk, V.; Gammerman, A.; and Shafer, G. 2005. *Algorithmic learning in a random world*. Springer US. ISBN 0387001522.

Wegkamp, M.; and Yuan, M. 2011. Support vector machines with a reject option. *Bernoulli*.

Wenger, J.; Kjellström, H.; and Triebel), R. 2020. Non-Parametric Calibration for Classification. In Chiappa, S.; and Calandra, R., eds., *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, 178–190. PMLR.

Wilson, A. G.; Hu, Z.; Salakhutdinov, R.; and Xing, E. P. 2016. Deep kernel learning. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, AISTATS 2016*.

Wu, T. F.; Lin, C. J.; and Weng, R. C. 2004. Probability estimates for multi-class classification by pairwise coupling. *Journal of Machine Learning Research*.