Training Uncertainty-Aware Classifiers with Conformalized Deep Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

Deep neural networks are powerful tools to detect hidden patterns in data and leverage them to make predictions, but they are not designed to understand uncertainty and estimate reliable probabilities. In particular, they tend to be overconfident. We address this problem by developing a novel training algorithm that can lead to more dependable uncertainty estimates, without sacrificing predictive power. The idea is to mitigate overconfidence by minimizing a loss function, inspired by advances in conformal inference, that quantifies model uncertainty by carefully leveraging hold-out data. Experiments with synthetic and real data demonstrate this method leads to smaller conformal prediction sets with higher conditional coverage, after exact calibration with hold-out data, compared to state-of-the-art alternatives.

1 Introduction

The predictions of deep neural networks and other complex machine learning (ML) models affect important decisions in many applications [1–6], including autonomous driving, medical diagnostics, or security monitoring. Prediction errors in those contexts can be costly or even dangerous, which makes reliable and explainable uncertainty estimation essential. Unfortunately, deep neural networks are not designed to understand uncertainty and they are easily prone to overfitting; consequently, they may lead to overconfidence [7–9]. Overconfidence is especially problematic if the data are intrinsically noisy and perfect predictions are impossible; for instance, think of prognosticating COVID-19 outcomes [10, 11], assessing genetic disease predisposition [12–14], or anticipating credit card defaults [15]. In those applications, the outcome of interest is potentially complicated and likely depends on many unmeasured variables; therefore, unpredictable randomness must be expected and ML models should account for it. Among many existing techniques for estimating uncertainty in ML [7–9], conformal inference [16], stands out for its ability to provide finite-sample guarantees without strong assumptions about the data generating process or unrealistic algorithmic simplifications.

Conformal inference is designed to convert the output of any ML model into a *prediction set* of likely outcomes whose size is automatically tuned using *hold-out data*, in such a way that the same procedure applied to future test data will yield prediction sets that are well calibrated in a frequentist sense. In particular, these sets have provable *marginal coverage*; i.e., at the 90% level this means the outcome for a new random test point is contained in the output set 90% of the time. The hold-out data are utilized to evaluate *conformity scores*, or goodness-of-fit scores, whose ordering determines how the sets are to be expanded so that the desired fraction of test points is guaranteed to be covered.

A limitation of conformal inference is that it involves of two distinct phases, training and calibration, that are not designed to work together efficiently. The calibration algorithm takes as input pre-trained

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

models that may already be overconfident, and this is sub-optimal because bad habits are harder to correct after they become entrenched. As a result of this two-step approach, conformal predictions may be either unnecessarily conservative or overconfident for certain types of test cases, which can make them unreliable [17, 18] and unfair [19]. We address this challenge by synergetically combining the learning and calibration phases of conformal inference, reducing the overconfidence of the trained ML model and thereby obtaining more informative (smaller) prediction sets with more accurate coverage for all test points. The idea is to minimize a new loss function designed to measure the discrepancy in distribution between the conformity scores computed by the current model estimate and those of an imaginary oracle that leverages perfect knowledge of the data generating process to construct the most informative and reliable possible predictions.

Related work

Many methods have been developed to mitigate overconfidence in ML [20–34], for example by allowing an *agnostic* output, by suitable post-processing [7, 35–40], or through early stopping [41]. Additional relevant research includes that of [42–46, 46–52], and these will provide us with informative benchmarks. However, unlike conformal inference, these methods have no frequentist guarantees in finite samples, and largely rely on loss functions targeting the accuracy of best-guess predictions, without explicitly addressing uncertainty during training. We build upon conformal inference [53–58], pioneered by [16], which typically deals with off-the-shelf models [16, 18, 59, 60]. Although other very recent works have proposed leveraging ideas from this field to improve training [61–66], this paper is novel as it combines the adaptive conformity scores of [18] with a completely new uncertainty-aware loss function. This departs from [62, 63], which sought to minimize the cardinality of the prediction sets, and from [61, 64–66], which utilized conformal inference ideas for tuning low-dimensional hyper-parameters as opposed to fully guiding the training of all model parameters. Although we focus on classification [18], our method could be repurposed for regression [58, 67] and other supervised tasks [60, 68, 69]. Conformal inference can also be utilized to test hypotheses and calibrate probabilities [70], and our work could be extended to those problems.

2 Relevant background on conformal inference

2.1 Uncertainty quantification via conformal prediction sets

Consider a data set of i.i.d. (or sometimes simply *exchangeable*) observations $(X_i, Y_i)_{i=1}^{n+1}$ sampled from an arbitrary unknown distribution P_{XY}. Here, $X_i \in \mathbb{R}^p$ contains p features for the ith sample, and $Y_i \in \{1, \ldots, K\} = [K]$ denotes its label, which we assume to be one of K possible categories. The goal is to train a model on n data points, $(X_i, Y_i)_{i=1}^{n}$, and construct a reasonably small prediction set $\hat{\mathcal{C}}_{n,\alpha} \subseteq [K]$ for Y_{n+1} given X_{n+1} such that, for some fixed level $\alpha \in (0, 1)$,

$$\mathbb{P}\left[Y_{n+1} \in \hat{\mathcal{C}}_{n,\alpha}(X_{n+1})\right] \geq 1 - \alpha. \tag{1}$$

This property is called *marginal coverage* because it treats $(X_i, Y_i)_{i=1}^{n+1}$ as all random. If $\alpha = 0.1$, it ensures Y_{n+1} is contained in the prediction sets 90% of the time. Marginal coverage is practically feasible but not fully satisfactory, as it is not as reliable and informative as *conditional coverage*:

$$\mathbb{P}\left[Y_{n+1} \in \hat{\mathcal{C}}_{n,\alpha}(x) \mid X_{n+1} = x\right] \geq 1 - \alpha, \qquad \forall x \in \mathbb{R}^p. \tag{2}$$

Conditional coverage would give one confidence that $\hat{\mathcal{C}}_{n,\alpha}(x)$ contains the true Y for any individual data point, which is stronger than (1). For example, imagine a population in which color is a feature and 90% of samples are blue while the others are red. Then, 90% marginal coverage is attained by any prediction sets that contain the true Y for all blue samples but never do for the red ones. Valid coverage can be obtained conditional on a given *protected category* [19], but it is impossible to guarantee (2) more generally without unrealistically strong assumptions [71, 72]. Thus, a typical compromise is to construct prediction sets with marginal coverage and hope they are also reasonably valid conditional on X. For multi-class classification, a solution is offered by the conformity scores developed in [18], which are reviewed below as the starting point of our contribution.

2.2 Review of adaptive conformity scores for classification

Imagine an *oracle* knowing the conditional distribution of Y given X, namely $P_{Y|X}$, and think of how it would construct the smallest possible prediction sets $\mathcal{C}_\alpha^{\text{oracle}}(x)$ with exact $1-\alpha$ conditional coverage. For any $x \in \mathcal{X}$ and $y \in [K]$, define $\pi_y(x) = \mathbb{P}[Y = y \mid X = x]$. Then, the oracle would output the smallest subset $S \subseteq [K]$ such that $\sum_{y \in S} \pi_y(x) \geq 1-\alpha$. In truth, this set may have coverage strictly larger than $1-\alpha$ due to the discreteness of Y; however, exact coverage can be achieved by introducing a little extra randomness [19]. For any $\tau \in (0,1)$ and $u \in (0,1)$, let $\mathcal{S}$ be the function with input x, u, π, and τ that computes the set of most likely labels up to (but possibly excluding) the one identified by the above randomized oracle; this is written explicitly in Appendix A1.1. If U is a uniform random variable independent of everything else, it is easy to verify that the following oracle prediction sets have exact conditional coverage at level $1-\alpha$:

$$C_\alpha^{\text{oracle}}(x) = \mathcal{S}(x, U; \pi, 1-\alpha). \tag{3}$$

For example, if $\pi_1(x) = 0.3$, $\pi_2(x) = 0.6$, and $\pi_3(x) = 0.1$, then $C_{0.2}^{\text{oracle}}(x) = \{1,2\}$ with probability $2/3$, and $C_{0.2}^{\text{oracle}}(x) = \{2\}$ with probability $1/3$. Of course, this oracle is only a thought experiment because $P_{Y|X}$ is generally unknown. Therefore, to construct practical prediction intervals, π must be replaced by an ML model $\hat{\pi}$. Then, conformal inference is needed to ensure the output sets based on the possibly inaccurate ML model at least satisfy marginal coverage (1).

Conformal inference [18] begins by training any classifier on part of the data, indexed by $\mathcal{D}_1 \subset [n]$, to fit an approximation $\hat{\pi}$ of the unknown π. After substituting $\hat{\pi}$ into the oracle rule $\mathcal{S}$, the hold-out data indexed by $\mathcal{D}_2 = [n] \setminus \mathcal{D}_1$ are leveraged to adjust the prediction sets as to empirically achieve the desired coverage. More precisely, define the following *conformity scores* W_i for all observations in $\mathcal{D}_2$, and for the test point $n+1$. Intuitively, W_i is the smallest $\tau \in [0,1]$ such that $\mathcal{S}(X_i, U_i; \hat{\pi}, \tau)$ contains the true Y_i, while U_i is a uniform random variable independent of everything else:

$$W_i = W(X_i, Y_i, U_i; \hat{\pi}) = \min\left\{\tau \in [0,1] : Y_i \in \mathcal{S}(X_i, U_i; \hat{\pi}, \tau)\right\}, \qquad i \in \mathcal{D}_2 \cup \{n+1\}. \tag{4}$$

These statistics are observed for all $i \in \mathcal{D}_2$, but not for $n+1$. Define also $\hat{\tau}_{n,\alpha}$ as the $\lceil (1-\alpha)(1+|\mathcal{D}_2|) \rceil$ largest element of $\{W_i\}_{i \in \mathcal{D}_2}$. Intuitively, $\hat{\tau}_{n,\alpha}$ is the smallest $\tau \in [0,1]$ such that $\mathcal{S}(X_i; \hat{\pi}, \tau)$ contains a fraction $1-\alpha$ of the hold-out data in $\mathcal{D}_2$. Then, the output prediction set for a new X_{n+1} is $\mathcal{C}_{n,\alpha}(X_{n+1}) = \mathcal{S}(X_{n+1}, U_{n+1}; \hat{\pi}, \hat{\tau}_{n,\alpha})$. This has $1-\alpha$ marginal coverage due to the exchangeability of the calibration and test data [18]. In fact, $Y_{n+1} \notin \mathcal{C}_{n,\alpha}(X_{n_1})$ implies $W_{n+1} > \hat{\tau}_{n,\alpha}$, and by exchangeability the probability of this event is smaller than α; see [58]. Unlike alternative approaches based on different scores [16, 17, 53], this solution would yield prediction sets equivalent to those of the oracle [18] if $\hat{\pi} = \pi$. Although generally $\hat{\pi} \neq \pi$, the above prediction sets often achieve relatively high conditional coverage [18] in practice. Our goal is to further improve their empirical performance by training the ML model to be more deliberately aware of uncertainty.

3 Methods

3.1 The distribution of the adaptive conformity scores

The conformity scores defined in (4) are uniformly distributed conditional on $X = x$, for any x, if $\hat{\pi} = \pi$. This property was hinted without proof in [18] and it serves as the starting point of our contribution. Note that all mathematical proofs can be found in Appendix A2.

Proposition 1. *The distribution of the conformity scores W_i in* (4) *is uniform conditional on X_i if $\hat{\pi} = \pi$. That is, $\mathbb{P}[W(X,Y,U;\pi) \leq \beta \mid X = x] = \beta$ for all $\beta \in (0,1)$, where (X,Y) is a random sample from $P_{X,Y}$, and $U \sim \text{Uniform}[0,1]$ independent of everything else. Further, $W_i \mid X_i$ is uniform if and only if $\mathcal{S}(X_i; \hat{\pi}, 1-\alpha)$ has conditional coverage at level $1-\alpha$ for all $\alpha \in [0,1]$.*

As we seek accurate conditional coverage, this result suggests training $\hat{\pi}$ as to produce scores that are approximately uniform on hold-out data, at least marginally. Therefore, we will evaluate (4) on hold-out data *while training* $\hat{\pi}$, encouraging the conformity scores to follow a uniform distribution.

3.2 An uncertainty-aware conformal loss function

We develop a loss function that approximately measures the deviation from uniformity of the conformity scores defined in (4) by combining classical non-parametric tests for equality in distribution with fast algorithms for smooth sorting and ranking [73, 74]. This loss is combined with the traditional cross entropy as to also promote accurate predictions, and it can be approximately optimized by stochastic gradient descent (it will generally be non-convex). The novel uncertainty-aware component of this loss only sees the hold-out samples through the lens of a non-parametric test applied to the empirical score distribution within a subset of the data. Therefore, it provides little incentive to overfit compared to a traditional loss targeting point-wise predictive accuracy, such as the cross entropy. By contrast, it discourages overconfident predictions which would yield non-uniform scores, as we shall see below. This solution is outlined in Figure A1, Appendix A1.2, and detailed below.

First, the n training samples are partitioned into two subsets, $\mathcal{I}_1$ and $\mathcal{I}_2$ such that $\mathcal{I}_1 \cup \mathcal{I}_2 = [n] = \mathcal{D}_1$. Here we assume the training data are indexed by $\mathcal{D}_1 = [n]$; this notation is slightly different from Section 2, but it is simple and does not introduce ambiguity because the additional calibration data in $\mathcal{D}_2$ remain untouched during training. The training algorithm approximately minimizes a loss function ℓ consisting of two additive components, each evaluated on one subset of the data:

$$\ell = (1-\lambda) \cdot \ell_{\mathrm{a}}(\mathcal{I}_1) + \lambda \cdot \ell_{\mathrm{u}}(\mathcal{I}_2). \tag{5}$$

Above, the hyper-parameter $\lambda \in [0, 1]$ controls the relative weights of the two components. The ℓ_{a} component is evaluated on the data in $\mathcal{I}_1$, and its purpose is to seek high predictive accuracy, as customary. For example, this could be the cross entropy:

$$\ell_{\mathrm{a}} = -\frac{1}{|\mathcal{I}_1|} \sum_{i \in \mathcal{I}_1} \sum_{c=1}^{K} \mathbb{1}\left[Y_i = c\right] \log \hat{\pi}_c(X_i), \tag{6}$$

where $\hat{\pi}$ is the output of the final softmax layer. The novel uncertainty-aware component ℓ_{u} is evaluated on $\mathcal{I}_2$, and its role is to mitigate overconfidence. Concretely, conformity scores W_i are evaluated according to (4) for all $i \in \mathcal{I}_2$, and their empirical distribution is compared to the ideal uniformity expected if $\hat{\pi} = \pi$. Ideally, we would like to quantify this discrepancy by directly applying a powerful non-parametric test, for example by computing the Cramér-von Mises or [75, 76] Kolmogorov-Smirnov [77, 78] test statistics. Concretely, in the latter case,

$$\ell_{\mathrm{u}} = \sup_{w \in [0,1]} \left| \hat{F}_{|\mathcal{I}_2|}(w) - w \right|, \tag{7}$$

where $\hat{F}_{|\mathcal{I}_2|}(\cdot)$ is the empirical cumulative distribution function (CDF) of W_i for $i \in \mathcal{I}_2$: $\hat{F}_{|\mathcal{I}_2|}(w) = (1/|\mathcal{I}_2|) \sum_{i \in \mathcal{I}_2} \mathbb{1}\left[W_i \leq w\right]$. Unfortunately, $\hat{F}_{|\mathcal{I}_2|}(\cdot)$ is not differentiable, which makes the overall loss intractable to minimize. This requires introducing some approximations in ℓ_{u}, as explained next.

3.3 Differentiable approximations

The empirical CDF of the conformity scores is not differentiable because it involves sorting, which is a non-smooth operation. Further, these scores themselves are not differentiable in the model parameters θ because they involve ranking and sorting the estimated class probabilities $\hat{\pi}$. In fact, the score (4) can be computed in a closed form,

$$W_i = \hat{\pi}_{(1)}(X_i) + \hat{\pi}_{(2)}(X_i) + \ldots + \hat{\pi}_{(r(Y_i, \hat{\pi}(X_i)))}(X_i) - U_i \cdot \hat{\pi}_{(r(Y_i, \hat{\pi}(X_i)))}(X_i), \tag{8}$$

where U_i is a uniform random variable independent of everything else; see Appendix A1.1 for details. Fortunately, there exist fast approximate algorithms for differentiable sorting and ranking that work well in combination with standard back-propagation [73, 74]. Note that evaluating W_i in (8) requires accessing elements of $(\hat{\pi}_{(1)}(X_i), \ldots, \hat{\pi}_{(K)}(X_i))$ through a θ-dependent index, $r(Y_i, \hat{\pi}(X_i))$, which is also non-differentiable. Therefore, indexing by $r(Y_i, \hat{\pi}(X_i))$ must be approximated with a smooth linear interpolation. In conclusion, the ℓ_{u} loss in (7) is approximated by evaluating a differentiable version of the scores in (4) as described above, and then by replacing their empirical CDF with a differentiable approximation obtained with the same techniques from [74]. This procedure, combined with stochastic gradient descent for fitting the model parameters θ, is summarized in Algorithm 1. Although here we assume $\mathcal{I}_1 = \mathcal{I}_2$ for simplicity, this algorithm can easily accommodate $\mathcal{I}_1 \neq \mathcal{I}_2$.

Algorithm 1: Conformalized uncertainty-aware training of deep multi-class classifiers

Input: Data $\{X_i, Y_i\}_{i=1}^n$; hyper-parameter $\lambda \in [0, 1]$, learning rate $\gamma > 0$, batch size M;
Randomly initialize the model parameters $\theta^{(0)}$;
Randomly split the data into two disjoint subsets, $\mathcal{I}_1, \mathcal{I}_2$, such that $\mathcal{I}_1 \cup \mathcal{I}_2 = [n]$;
Set the number of batches to $B = (n/2)/M$ (assuming for simplicity that $|\mathcal{I}_1| = |\mathcal{I}_2|$);
for $t = 1, \ldots, T$ **do**
  Randomly divide $\mathcal{I}_1$ and $\mathcal{I}_2$ into B batches;
  for $b = 1, \ldots, B$ **do**
    Evaluate (softmax) conditional probabilities $\hat{\pi}(X_i)$ for all i in batch b of $\mathcal{I}_1 \cup \mathcal{I}_2$;
    Generate a uniform independent random variable U_i for all i in batch b of $\mathcal{I}_2$;
    Evaluate $\tilde{W}_i$ for all i in batch b of $\mathcal{I}_2$, using U_i and a differentiable approximation of (4);
    Evaluate the gradient $\nabla \ell_{\mathrm{a}}(\theta^{(t)})$ of ℓ_{a} in (6) using the data in batch b of $\mathcal{I}_1$;
    Evaluate the gradient $\nabla \tilde{\ell}_{\mathrm{u}}(\theta^{(t)})$ of a differentiable approximation $\tilde{\ell}_{\mathrm{u}}$ of ℓ_{u} in (7) using the differentiable scores $\tilde{W}_i$ in batch b of $\mathcal{I}_2$;
    Define $\nabla \tilde{\ell}(\theta^{(t)}) = (1 - \lambda) \cdot \nabla \ell_{\mathrm{a}}(\theta^{(t)}) + \lambda \cdot \nabla \tilde{\ell}_{\mathrm{u}}(\theta^{(t)})$ based on (5);
    Update the model parameters: $\theta^{(t)} \leftarrow \theta^{(t-1)} - \gamma \nabla \tilde{\ell}(\theta^{(t-1)})$.
  end
end
Output: The model $\hat{\pi}$ corresponding to the fitted parameters $\theta^{(T)}$.

3.4 Theoretical analysis

The uncertainty-aware loss function in Algorithm 1 can be justified theoretically by noting that it is approximately minimized (although possibly non-uniquely) by the imaginary oracle model π, which yields the smallest possible prediction sets with exact conditional coverage. This analysis focuses on the original version of the loss function defined in (5)–(7), ignoring for simplicity the additional subtleties introduced by the differentiable approximations described in Section 3.3.

Proposition 2. *The loss function $\ell \geq 0$ in* (5) *is bound from above by $\ell^0 + \delta\ell$, where $\ell^0 \geq 0$ attains value zero if $\hat{\pi} = \pi$, and $\delta\ell = \mathcal{O}_{\mathbb{P}}(1/\sqrt{M})$ as $\mathcal{I}_2 \to \infty$.*

Of course, Algorithm 1 does not minimize (5) exactly because it involves solving a high-dimensional non-convex optimization problem that is difficult to study theoretically. Yet, it is possible to prove at least a weak form of convergence for its stochastic gradient descent, whose solution may not however necessarily approach a global minimum. This analysis is in Appendix A2 for lack of space.

4 Numerical experiments

4.1 Experiments with synthetic data

The performance of Algorithm 1 is investigated here on synthetic data that mimic a multi-class classification problem in which most samples are relatively easy to classify but a few are unpredictable. Specifically, data are simulated with 100 independent and uniformly distributed features $X = (X_1, \ldots, X_{100}) \in [0, 1]^{100}$ and a label $Y \in [K]$, for $K = 6$. The first feature controls whether the sample is intrinsically difficult to classify, while the next two features determine the most likely labels; all other features are useless. On average, 20% of the samples are impossible to classify with absolute confidence. This conditional distribution is written explicitly in Appendix A3.1.

The conditional class probabilities $\hat{\pi}$ are estimated as the output of a final softmax layer in a fully connected neural network implemented with PyTorch [79]; see Appendix A3.1 for more information about network architecture and training details. This model is fitted separately with Algorithm 1 and three benchmark techniques including traditional cross entropy minimization and focal loss minimization [80]. Unfortunately, we cannot directly compare to the recent methods of [62, 63] as originally implemented by those authors for lack of openly available computer code. Instead, we

consider a hybrid benchmark that combines elements of Algorithm 1 with the main idea of [63], essentially seeking a model that yields small conformal prediction sets, irrespective of conditional coverage; see Appendix A1.3 for details about this benchmark. As early stopping can help mitigate overfitting [41], it is informative to also investigate its effect on each of the aforementioned learning algorithms. For this purpose, we generate an additional validation set of 2000 independent data points and use it to preview the out-of-sample accuracy and loss value at each epoch. Then, the best versions of each model according to these two early stopping criteria are saved during training. After training each model, 10,000 additional independent samples are utilized to calibrate split-conformal prediction sets with 90% marginal coverage, as explained in Section 2.2.

Figure 1 compares the performance of conformal prediction sets obtained with each model for 2000 test points as a function of the number of training samples, averaging over 50 independent experiments. The prediction sets are evaluated in terms of their average size and coverage conditional on $X_1 \leq \delta$; i.e., separately for the "hard" samples. For each learning algorithm, the results corresponding to the model achieving the highest conditional coverage among the fully trained and two early stopped alternatives are reported. This allows us to focus on the overall behaviour of different losses while accounting for possible differences in the optimal choices of early-stopping strategies.

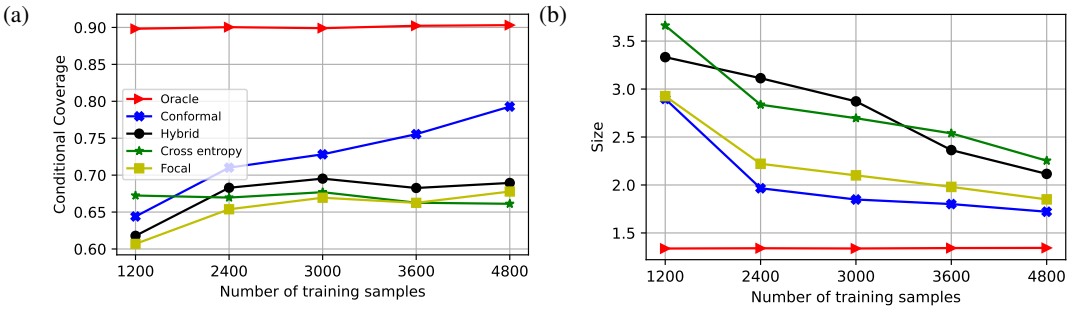

Figure 1: Performance of conformal prediction sets with 90% marginal coverage based on the ideal oracle model and a deep neural network trained with four alternative algorithms. (a): Conditional coverage for intrinsically hard samples. (b): Average size of prediction sets. The results are averaged over 50 independent experiments and the standard errors are below 0.1 (not shown explicitly).

Figure 1 does not visualize marginal coverage because that is guaranteed to be above 90%. The results show our algorithm yields the prediction sets with highest conditional coverage and smallest size, especially when the number of training samples is large. By contrast, the conditional coverage obtained with the cross entropy does not visibly improve as the training data set grows, suggesting systematic over-confidence with hard-to-classify samples. The focal loss improves upon the cross entropy by reducing the average size of the prediction sets, but it does not lead to higher conditional coverage. Finally, the hybrid algorithm increases conditional coverage slightly compared to the cross entropy, but it does not lead to smaller prediction sets. This is likely because the hybrid loss does not introduce very different incentives compared to the cross entropy, as the latter already attempts to maximize the estimated probability of the observed labels, thus effectively seeking small conformal prediction sets without necessarily high conditional coverage. Note that the focal loss is applied here with hyper-parameter $\gamma = 1$, as we have found larger values to yield lower accuracy; see Figure A2.

The improved performance of our conformal learning method can be understood by looking at the distribution of the corresponding conformity scores evaluated on test data, as shown in Figure A3, Appendix A3.2, for the model trained on 2400 data points. These results demonstrate our method leads to scores that are more uniformly distributed than those obtained with the cross entropy, which is indicative of more reliable uncertainty quantification. In fact, the conditional class probabilities $\hat{\pi}$ estimated by our model are more similar to the true oracle probabilities π compared to those obtained by minimizing the cross entropy loss; see Figure A4. Note that these figures refer to fully trained models, without early stopping. Analogous results obtained with early stopping are presented later.

Figure A5 compares the performance of the prediction sets obtained with each method when covariate shift occurs in the test data. In particular, here we imagine that at test time the uncertainty-controlling feature X_1 is sampled uniformly from $[0, a]$ with $a \leq 1$, so that lower values of a correspond to higher proportions of intrinsically hard-to-classify samples. Of course, in this case marginal coverage is no longer guaranteed for the same reason why conditional coverage in Figure 1 is not always controlled. As expected, all models produce smaller set sizes with higher marginal coverage for a closer to 1, consistently with Figure 1, but our method outperforms the benchmarks.

Several additional results are in Appendix A3.2. Figure A6 reports on experiments in which the number K of labels is varied, ranging from 4 to 12. These results show our methods leads to prediction sets with consistently smaller size and typically higher conditional coverage compared to all benchmarks. Figure A7 reports on experiments in which the proportion of hard-to-classify samples is varied, ranging from 0.1 to 0.5. All models lead to higher conditional coverage for larger δ, as implied by the fixed marginal coverage, but our method consistently achieves it with smaller prediction sets. Figures A8–A11 report on experiments with models trained using early stopping based on maximum prediction accuracy on the validation data. Again, our method achieves higher conditional coverage and smaller prediction sets relative to the benchmarks. Figures A12–A15 report analogous results obtained with early stopping based on minimum validation loss. Figures A16–A17 (resp. A18–A19) show the distribution of conformity scores and the estimated class probabilities, as in Figures A3–A4, from models trained with early stopping based on validation predictive accuracy (resp. loss). Finally, Figures A20–A21 (resp. A22–A23) show the distribution of conformity scores and the estimated class probabilities from the focal loss (res. hybrid) models.

4.2 Experiments with CIFAR-10 data

Convolutional neural networks guided by the conformal loss are trained on the publicly available CIFAR-10 image classification data set [81] (10 classes), and the models thus obtained are compared to those targeting the three benchmark losses considered before. As these data are not too hard to classify, we make the problem more interesting by randomly applying RandomErasing [82] to some images—a form of corruption that makes images very hard to recognize. The number of training samples is varied from 3000 to 45000. See Appendix A3.3 for details. To measure the performance of conformal prediction sets based on each model, we set aside 5000 calibration and test observations prior to training. The proportions of corrupt images in the calibration and test sets are 0.2, while that in the training set is varied. All models are calibrated as in [18], seeking 90% marginal coverage. Their performance is measured on the test data in terms of the size of the output prediction sets and their coverage conditional on the indicator of corruption. Further, the test accuracy of each model is evaluated based on the misclassification rate of its best-guess label. Figure 2 showcases two example test images, respectively intact and corrupted by RandomErasing, along with their corresponding conditional class probabilities calculated by different models fully trained on 45000 data points. This shows the model trained with our conformal loss is not as overconfident when dealing with the corrupted images as that minimizing the cross entropy.

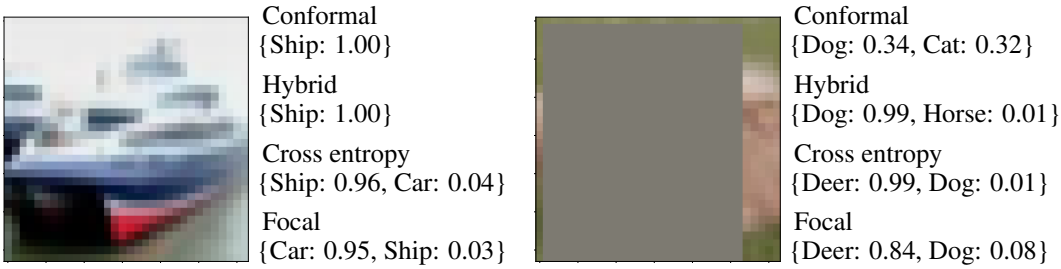

Figure 2: Two example test images from the CIFAR-10 data set, with their top two estimated class probabilities computed by the output softmax layer of convolutional neural networks trained to minimize different loss functions. Left: intact image of a ship. Right: corrupted image of a dog.

Figure A24 in Appendix A3.4 summarizes the performance of the prediction sets obtained with each fully trained model, as a function of the number of training samples. Median performance measures are reported over 10 experiments with random data subsets. The conformal loss leads to prediction sets with higher coverage for intrinsically hard images, and smaller size for the easy ones. As shown in Figure A25, this improvement is associated with higher test accuracy for the models targeting our loss, consistently with their increased robustness to overfitting. As overfitting can also be mitigated by early stopping, we report in Figure A26 the corresponding results obtained with early stopping based on maximum accuracy on a validation data set of size 2000 (or 5000, training with 45000 samples). In this case, all models achieve similar test accuracy, but those trained with our method lead to conformal prediction sets with (slightly) higher conditional coverage and smaller size, especially if trained on many samples and compared to the focal loss. These conclusions are summarized in Table 1 in the case of 45000 training samples, which includes also the results corresponding to models trained with early stopping based on validation accuracy; see Table A1 for analogous results with early stopping based on validation loss. Overall, the models targeting the conformal loss perform best when trained fully or with early stopping based on accuracy, and they tend to achieve higher coverage for the hard images while enabling smaller prediction sets with valid coverage for the easy cases. A similar summary for the models trained with 3000 samples is in Table A2; there, the advantage of the conformal loss compared to the hybrid method is not as marked as in the large-sample experiments.

	Coverage intact/corrupted		Size intact/corrupted		Accuracy intact/corrupted	
	Full	ES (acc)	Full	ES (acc)	Full	ES (acc)
Conformal	0.90/0.90	0.90/0.87	1.41/5.95	1.41/6.09	0.81/0.35	0.81/0.35
Hybrid	0.92/0.81	0.91/0.86	5.54/5.75	1.38/5.30	0.65/0.40	0.83/0.37
Cross Entropy	0.92/0.84	0.92/0.81	3.30/4.43	1.50/5.05	0.68/0.43	0.82/0.36
Focal	0.91/0.83	0.93/0.77	2.57/3.99	1.91/4.25	0.68/0.42	0.77/0.34

Table 1: Conditional coverage and size of conformal prediction sets with 90% marginal coverage on CIFAR-10 data, based on models trained on 45000 data points. The models are trained either fully for many epochs, or with early stopping (ES) based on classification accuracy (acc). The last two columns report the best-guess classification accuracy of the underlying models applied to test data.

Figure A27 reports performance measures as in Figure A24, after fixing the number of training samples to 45000 and varying the corruption proportion. The model trained with the conformal loss leads to smaller prediction sets with higher conditional coverage compared to the benchmarks, and its test accuracy does not decrease as the proportion of corrupted training images increases. Figure A28 demonstrates early stopping mitigates overfitting and allows the benchmarks to maintain relatively high accuracy as the training corruption proportion increases. However, the conformal loss outperforms even if all models are trained with early stopping based on validation accuracy. The distributions of conformity scores on test data are compared in Figures A29–A32. These results confirm the scores obtained with the conformal loss tend to be more uniform compared to the benchmarks, especially if the training sample size is large. If few training samples are available, the focal loss leads to scores that are slightly closer to being uniform, but at the cost of lower accuracy and conditional coverage (Table A2). In Figures A33–A36, the performances of all models are compared separately for each of the 10 possible test labels. These results suggest corrupt images with true labels 4,5, or 6 are the most difficult ones to classify, and their prediction sets have the lowest coverage. However, this issue is mitigated by models fully trained to minimize the proposed conformal loss. Further, the results confirm that models trained with our loss yield more informative prediction sets for the easier unperturbed images, while achieving the desired coverage rate.

4.3 Experiments with credit card data

In this section, we analyze a publicly available credit card default data set [83] containing 30000 observations of 23 features and a binary label. Approximately 22% of the labels are equal to 1.

The data are randomly divided into 16800 training samples, 4500 calibration samples, and 4500 test samples, while the remaining 4200 samples are utilized to determine early stopping rules. All experiments are repeated 20 times with independent data subsets. The conformal and hybrid losses are implemented using 70% of the training samples for evaluating the cross entropy component. The model architecture is as in Section 4.1; see Appendix A3.5 for further implementation details.

The performances of conformal prediction sets for models trained with different losses are measured in terms of their respective sizes and coverage conditional on the true label being equal to 1. As the samples with label 1 are a minority, constructing prediction sets with valid coverage for their group is an interesting problem that may be relevant in the context of algorithmic fairness [19]. Note that it is possible to *calibrate* conformal prediction sets in such a way as to achieve perfect label-conditional coverage [16, 18], at the cost of higher data usage, but here we focus on *training* uncertainty-aware models as to approximately achieve this goal without explicit label-conditional calibration. Therefore, we apply a slightly modified version of Algorithm 1 in which the empirical distribution of the hold-out conformity scores is evaluated separately on the samples from each class, and then the sum of these two instances of (7) is utilized as ℓ_u. The same strategy is also incorporated into the hybrid method.

Table 2 compares the average performances of each model and shows our method achieves the highest conditional coverage. Here, early stopping is applied based on minimum validation loss, as the alternative accuracy criterion achieved lower conditional coverage for all models on these data. Figure A37 and Table A3 in Appendix A3.6 demonstrate our model leads to conformity scores that are closer to being uniformly distributed when evaluated on test data, as expected.

	Coverage				Size		Classification	
	Marginal		Conditional		all labels/0/1		error	
	Full	ES	Full	ES	Full	ES	Full	ES
Conformal	0.83	0.83	0.60	0.52	1.34/1.31/1.45	1.29/1.27/1.39	33.05	28.52
Hybrid	0.83	0.83	0.51	0.53	1.27/1.24/1.37	1.28/1.25/1.38	27.00	27.52
Cross Entropy	0.82	0.84	0.51	0.42	1.25/1.23/1.33	1.24/1.21/1.35	26.16	24.36
Focal	0.81	0.82	0.48	0.50	1.22/1.20/1.28	1.25/1.23/1.32	26.56	26.30

Table 2: Performance of conformal prediction sets with 80% marginal coverage on credit card default data, based on convolutional neural networks targeting different losses. Other details are as in Table 1.

5 Discussion

The conformal loss function presented in this paper mitigates overconfidence in deep neural networks and can lead to smaller prediction sets with higher conditional coverage compared to standard benchmarks. This contribution is practically relevant for many applications in which overfitting may occur, but it is especially useful when dealing with noisy data that do not allow very accurate out-of-sample predictions. One limitation of the proposed method is that it is more computationally expensive than its benchmarks. For example, training a conformal loss model on 45000 images in the CIFAR-10 data set took us approximately 20 hours on an Nvidia P100 GPU, while training models with the same architecture to minimize the cross entropy or focal loss only took about 11 hours. Another limitation of the conformal loss is that a relatively large number of training samples appears to be required for significant performance improvements. Nonetheless, the ability of an ML model to more openly admit ignorance when asked to provide an unknown answer is a valuable achievement for which it may sometimes be worth investing additional training resources. Interesting future research may explore extensions of this work to problems beyond multi-class classification or to ML models other than deep neural networks, as well as further applications to real-world data sets.

A software implementation of the proposed method written in Python and based on the PyTorch [79] framework is made available, along with all code needed to reproduce the numerical experiments.

References

[1] Jonathan Schmidt, Mário RG Marques, Silvana Botti, and Miguel AL Marques. Recent advances and applications of machine learning in solid-state materials science. *npj Comput. Mater.*, 5(1): 1–36, 2019.

[2] Alexandre Tkatchenko. Machine learning for chemical discovery. *Nat. Commun.*, 11(1):1–4, 2020.

[3] Gisbert Schneider. Mind and machine in drug design. *Nat. Mach. Intell.*, 1(3):128–130, 2019.

[4] Sorin Grigorescu, Bogdan Trasnea, Tiberiu Cocias, and Gigel Macesanu. A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, 37(3):362–386, 2020.

[5] Daniel S Hoadley and Nathan J Lucas. Artificial intelligence and national security, 2018.

[6] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nat. Mach. Intell.*, 1(5):206–215, 2019.

[7] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1321–1330. JMLR. org, 2017.

[8] Sunil Thulasidasan, Gopinath Chennupati, Jeff A Bilmes, Tanmoy Bhattacharya, and Sarah Michalak. On mixup training: Improved calibration and predictive uncertainty for deep neural networks. In *Adv. Neural. Inf. Process. Syst.*, pages 13888–13899, 2019.

[9] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *Adv. Neural. Inf. Process. Syst.*, pages 13991–14002, 2019.

[10] Chansik An, Hyunsun Lim, Dong-Wook Kim, Jung Hyun Chang, Yoon Jung Choi, and Seong Woo Kim. Machine learning prediction for mortality of patients diagnosed with covid-19: a nationwide korean cohort study. *Scientific Reports*, 10(1):18716, 2020. ISSN 2045-2322.

[11] Yue Gao, Guang-Yao Cai, Wei Fang, Hua-Yi Li, Si-Yuan Wang, Lingxi Chen, Yang Yu, Dan Liu, Sen Xu, Peng-Fei Cui, Shao-Qing Zeng, Xin-Xia Feng, Rui-Di Yu, Ya Wang, Yuan Yuan, Xiao-Fei Jiao, Jian-Hua Chi, Jia-Hao Liu, Ru-Yuan Li, Xu Zheng, Chun-Yan Song, Ning Jin, Wen-Jian Gong, Xing-Yu Liu, Lei Huang, Xun Tian, Lin Li, Hui Xing, Ding Ma, Chun-Rui Li, Fei Ye, and Qing-Lei Gao. Machine learning based early warning system enables accurate mortality risk prediction for covid-19. *Nat. Commun.*, 11(1):5033, 2020. ISSN 2041-1723.

[12] Michela Carlotta Massi, Francesca Gasperoni, Francesca Ieva, Anna Maria Paganoni, Paolo Zunino, Andrea Manzoni, Nicola Rares Franco, Liv Veldeman, Piet Ost, Valérie Fonteyne, et al. A deep learning approach validates genetic risk factors for late toxicity after prostate cancer radiotherapy in a requite multi-national cohort. *Frontiers in oncology*, 10, 2020.

[13] Adrien Badré, Li Zhang, Wellington Muchero, Justin C Reynolds, and Chongle Pan. Deep neural network improves the estimation of polygenic risk scores for breast cancer. *Journal of Human Genetics*, 66(4):359–369, 2021.

[14] Arno van Hilten, Steven A Kushner, Manfred Kayser, M Arfan Ikram, Hieab HH Adams, Caroline CW Klaver, Wiro J Niessen, and Gennady V Roshchupkin. Gennet framework: interpretable deep learning for predicting phenotypes from genetic data. *Communications biology*, 4(1):1–9, 2021.

[15] Ting Sun and Miklos A Vasarhalyi. Predicting credit card delinquencies: An application of deep neural networks. In *Handbook of Financial Econometrics, Mathematics, Statistics, and Machine Learning*, pages 4349–4381. World Scientific, 2021.

[16] Vladimir Vovk, Alex Gammerman, and Glenn Shafer. *Algorithmic learning in a random world*. Springer, 2005.

[17] Maxime Cauchois, Suyash Gupta, and John Duchi. Knowing what you know: valid confidence sets in multiclass and multilabel prediction. *arXiv:2004.10181*, 2020.

[18] Yaniv Romano, Matteo Sesia, and Emmanuel J. Candès. Classification with valid and adaptive coverage. In *Adv. Neural. Inf. Process. Syst.*, 2020.

[19] Yaniv Romano, Rina Foygel Barber, Chiara Sabatti, and Emmanuel Candès. With malice toward none: Assessing uncertainty via equalized coverage. *Harvard Data Science Review*, 2020.

[20] Marten Wegkamp. Lasso type classifiers with a reject option. *Electron. J. Statist.*, 1:155–168, 2007.

[21] Yves Grandvalet, Alain Rakotomamonjy, Joseph Keshet, and Stéphane Canu. Support vector machines with a reject option. In *Adv. Neural. Inf. Process. Syst.*, pages 537–544, 2009.

[22] Corinna Cortes, Giulia DeSalvo, and Mehryar Mohri. Boosting with abstention. In *Adv. Neural. Inf. Process. Syst.*, pages 1660–1668, 2016.

[23] Cláudio Rebelo de Sá, Carlos Soares, Arno Knobbe, and Paulo Cortez. Label ranking forests. *Expert systems*, 34(1):e12166, 2017.

[24] Leying Guan and Rob Tibshirani. Prediction and outlier detection in classification problems. *preprint arXiv:1905.04396*, 2019.

[25] Peter L Bartlett and Marten H Wegkamp. Classification with a reject option using a hinge loss. *J. Mach. Learn. Res.*, 9:1823–1840, 2008.

[26] Juan José Del Coz, Jorge Díez, and Antonio Bahamonde. Learning nondeterministic classifiers. *J. Mach. Learn. Res.*, 10(10), 2009.

[27] Yair Wiener and Ran El-Yaniv. Agnostic selective classification. In *Adv. Neural. Inf. Process. Syst.*, pages 1665–1673, 2011.

[28] Heinrich Jiang, Been Kim, Melody Guan, and Maya Gupta. To trust or not to trust a classifier. In *Adv. Neural. Inf. Process. Syst.*, pages 5541–5552, 2018.

[29] Yotam Hechtlinger, Barnabás Póczos, and Larry Wasserman. Cautious deep learning. *arXiv:1805.09460*, 2018.

[30] Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning*, 110(3):457–506, 2021.

[31] Ziyin Liu, Zhikang Wang, Paul Pu Liang, Russ R Salakhutdinov, Louis-Philippe Morency, and Masahito Ueda. Deep gamblers: Learning to abstain with portfolio theory. In *Adv. Neural. Inf. Process. Syst.*, pages 10623–10633, 2019.

[32] Ron Slossberg, Oron Anschel, Amir Markovitz, Ron Litman, Aviad Aberdam, Shahar Tsiper, Shai Mazor, Jon Wu, and R Manmatha. On calibration of scene-text recognition models. *arXiv preprint arXiv:2012.12643*, 2020.

[33] Lijing Wang, Dipanjan Ghosh, Maria Gonzalez Diaz, Ahmed Farahat, Mahbubul Alam, Chetan Gupta, Jiangzhuo Chen, and Madhav Marathe. Wisdom of the ensemble: Improving consistency of deep learning models. *Advances in Neural Information Processing Systems*, 33:19750–19761, 2020.

[34] Sijie Yan, Yuanjun Xiong, Kaustav Kundu, Shuo Yang, Siqi Deng, Meng Wang, Wei Xia, and Stefano Soatto. Positive-congruent training: Towards regression-free model updates. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14299–14308, 2021.

[35] John Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.

[36] Bianca Zadrozny and Charles Elkan. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *Icml*, volume 1, pages 609–616. Citeseer, 2001.

[37] Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 694–699, 2002.

[38] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1613–1622, Lille, France, 2015. PMLR.

[39] Juozas Vaicenavicius, David Widmann, Carl Andersson, Fredrik Lindsten, Jacob Roll, and Thomas Schön. Evaluating model calibration in classification. In Kamalika Chaudhuri and Masashi Sugiyama, editors, *Proceedings of Machine Learning Research*, volume 89 of *Proceedings of Machine Learning Research*, pages 3459–3467. PMLR, 2019.

[40] Lukas Neumann, Andrew Zisserman, and Andrea Vedaldi. Relaxed softmax: Efficient confidence auto-calibration for safe pedestrian detection, 2018.

[41] Lutz Prechelt. Early stopping-but when? In *Neural Networks: Tricks of the trade*, pages 55–69. Springer, 1998.

[42] Antonio Bella, Cèsar Ferri, José Hernández-Orallo, and María José Ramírez-Quintana. Calibration of machine learning models. In *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*, pages 128–146. IGI Global, 2010.

[43] Yarin Gal. Uncertainty in deep learning. *University of Cambridge*, 1(3), 2016.

[44] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Adv. Neural. Inf. Process. Syst.*, pages 6402–6413, 2017.

[45] Ananya Kumar, Percy S Liang, and Tengyu Ma. Verified uncertainty calibration. In *Adv. Neural. Inf. Process. Syst.*, pages 3787–3798, 2019.

[46] Natasa Tagasovska and David Lopez-Paz. Single-model uncertainties for deep learning. In *Adv. Neural. Inf. Process. Syst.*, pages 6417–6428, 2019.

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

[48] Wesley J Maddox, Pavel Izmailov, Timur Garipov, Dmitry P Vetrov, and Andrew Gordon Wilson. A simple baseline for bayesian uncertainty in deep learning. In *Adv. Neural. Inf. Process. Syst.*, pages 13153–13164, 2019.

[49] José Mena, Oriol Pujol, and Jordi Vitrià. Uncertainty-based rejection wrappers for black-box classifiers. *IEEE Access*, 2020.

[50] Yukun Ding, Jinglan Liu, Xiaowei Xu, Meiping Huang, Jian Zhuang, Jinjun Xiong, and Yiyu Shi. Uncertainty-aware training of neural networks for selective medical image segmentation. In *Medical Imaging with Deep Learning*, 2020.

[51] Biraja Ghoshal and Allan Tucker. Estimating uncertainty and interpretability in deep learning for coronavirus (covid-19) detection. *arXiv:2003.10769*, 2020.

[52] Eli Simhayev, Gilad Katz, and Lior Rokach. PIVEN: A deep neural network for prediction intervals with specific value prediction. *arXiv:2006.05139*, 2020.

[53] Jing Lei, James Robins, and Larry Wasserman. Distribution-free prediction sets. *J. Am. Stat. Assoc.*, 108(501):278–287, 2013.

[54] Jing Lei and Larry Wasserman. Distribution-free prediction bands for non-parametric regression. *J. Royal Stat. Soc. B*, 76(1):71–96, 2014.

[55] Jing Lei, Max G'Sell, Alessandro Rinaldo, Ryan J Tibshirani, and Larry Wasserman. Distribution-free predictive inference for regression. *J. Am. Stat. Assoc.*, 113(523):1094–1111, 2018.

[56] Rina Foygel Barber, Emmanuel J Candès, Aaditya Ramdas, and Ryan J Tibshirani. Predictive inference with the jackknife+. *The Annals of Statistics*, 49(1):486–507, 2021.

[57] Ryan J Tibshirani, Rina Foygel Barber, Emmanuel Candès, and Aaditya Ramdas. Conformal prediction under covariate shift. *Advances in neural information processing systems*, 32, 2019.

[58] Yaniv Romano, Evan Patterson, and Emmanuel J Candès. Conformalized quantile regression. In *Adv. Neural. Inf. Process. Syst.*, pages 3538–3548, 2019.

[59] Anastasios Angelopoulos, Stephen Bates, Jitendra Malik, and Michael I Jordan. Uncertainty sets for image classifiers using conformal prediction. *arXiv preprint arXiv:2009.14193*, 2020.

[60] Stephen Bates, Anastasios Angelopoulos, Lihua Lei, Jitendra Malik, and Michael Jordan. Distribution-free, risk-controlling prediction sets. *Journal of the ACM (JACM)*, 68(6):1–34, 2021.

[61] Nicolo Colombo and Vladimir Vovk. Training conformal predictors. In *Conformal and Probabilistic Prediction and Applications*, pages 55–64. PMLR, 2020.

[62] Anthony Bellotti. Optimized conformal classification using gradient descent approximation. *arXiv preprint arXiv:2105.11255*, 2021.

[63] David Stutz, Ali Taylan Cemgil, Arnaud Doucet, et al. Learning optimal conformal classifiers. *arXiv preprint arXiv:2110.09192*, 2021.

[64] Haoxian Chen, Ziyi Huang, Henry Lam, Huajie Qian, and Haofeng Zhang. Learning prediction intervals for regression: Generalization and calibration. In *International Conference on Artificial Intelligence and Statistics*, pages 820–828. PMLR, 2021.

[65] Yachong Yang and Arun Kumar Kuchibhotla. Finite-sample efficient conformal prediction. *arXiv preprint arXiv:2104.13871*, 2021.

[66] Yu Bai, Song Mei, Huan Wang, Yingbo Zhou, and Caiming Xiong. Efficient and differentiable conformal prediction with general function classes. *arXiv preprint arXiv:2202.11091*, 2022.

[67] Matteo Sesia and Yaniv Romano. Conformal prediction using conditional histograms. In *Adv. Neural. Inf. Process. Syst.*, volume 34, 2021.

[68] Anastasios N Angelopoulos, Amit P Kohli, Stephen Bates, Michael I Jordan, Jitendra Malik, Thayer Alshaabi, Srigokul Upadhyayula, and Yaniv Romano. Image-to-image regression with distribution-free uncertainty quantification and applications in imaging. *arXiv preprint arXiv:2202.05265*, 2022.

[69] Matteo Sesia and Stefano Favaro. Conformalized frequency estimation from sketched data. *arXiv preprint arXiv:2204.04270*, 2022.

[70] Stephen Bates, Emmanuel Candès, Lihua Lei, Yaniv Romano, and Matteo Sesia. Testing for outliers with conformal p-values. *arXiv preprint arXiv:2104.08279*, 2021.

[71] Vladimir Vovk. Conditional validity of inductive conformal predictors. In *Asian conference on machine learning*, pages 475–490, 2012.

[72] Rina Foygel Barber, Emmanuel J Candès, Aaditya Ramdas, and Ryan J Tibshirani. The limits of distribution-free conditional predictive inference. *Information and Inference: A Journal of the IMA*, 10(2):455–482, 2021.

[73] Marco Cuturi, Olivier Teboul, and Jean-Philippe Vert. Differentiable ranking and sorting using optimal transport. In *Adv. Neural. Inf. Process. Syst.*, pages 6861–6871, 2019.

[74] Mathieu Blondel, Olivier Teboul, Quentin Berthet, and Josip Djolonga. Fast differentiable sorting and ranking, 2020.

[75] Harald Cramér. On the composition of elementary errors: First paper: Mathematical deductions. *Scandinavian Actuarial Journal*, 1928(1):13–74, 1928.

[76] Richard Von Mises. Statistik und wahrheit. *Julius Springer*, 20, 1928.

[77] Nikolai V Smirnov. Estimate of deviation between empirical distribution functions in two independent samples. *Bulletin Moscow University*, 2(2):3–16, 1939.

[78] Frank J. Massey Jr. The kolmogorov-smirnov test for goodness of fit. *J. Am. Stat. Assoc.*, 46 (253):68–78, 1951.

[79] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[80] Jishnu Mukhoti, Viveka Kulharia, Amartya Sanyal, Stuart Golodetz, Philip Torr, and Puneet Dokania. Calibrating deep neural networks using focal loss. *Advances in Neural Information Processing Systems*, 33:15288–15299, 2020.

[81] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. CIFAR-10 data set, 2009. URL `http://www.cs.toronto.edu/~kriz/cifar.html`.

[82] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 13001–13008, 2020.

[83] I-Cheng Yeh and Che-hui Lien. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert systems with applications*, 36(2): 2473–2480, 2009.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...

(a) Did you state the full set of assumptions of all theoretical results? [Yes]
(b) Did you include complete proofs of all theoretical results? [Yes]

3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]