A novel Deep Learning approach for one-step Conformal Prediction approximation

Julia A. Meister[1*], Khuong An Nguyen[1†], Stelios Kapetanakis[2†] and Zhiyuan Luo[3†]

[1*]Computing Division, University of Brighton, Brighton, BN2 4GJ, East Sussex, United Kingdom.
[2]Distributed Analytics Solutions, 17 Fawe Street, London, E14 6FD, United Kingdom.
[3]Department of Computer Science, Royal Holloway University of London, Egham, TW20 0EX, Surrey, United Kingdom.

*Corresponding author(s). E-mail(s): J.Meister@brighton.ac.uk;
Contributing authors: K.A.Nguyen@brighton.ac.uk;
Stelios@distributedanalytics.co.uk; Zhiyuan.Luo@rhul.ac.uk;
†These authors contributed equally to this work.

Abstract

Deep Learning predictions with measurable confidence are increasingly desirable for real-world problems, especially in high-risk settings. The Conformal Prediction (CP) framework is a versatile solution that automatically guarantees a maximum error rate [1]. However, CP suffers from computational inefficiencies that limit its application to large-scale datasets [2]. In this paper, we propose a novel conformal loss function that approximates the traditionally two-step CP approach in a single step. By evaluating and penalising deviations from the stringent expected CP output distribution, a Deep Learning model may learn the direct relationship between input data and conformal p-values. Our approach achieves significant training time reductions up to 86% compared to Aggregated Conformal Prediction (ACP, [3]), an accepted CP approximation variant. In terms of approximate validity and predictive efficiency, we carry out a comprehensive empirical evaluation to show our novel loss function's competitiveness with ACP for binary and multi-class classification on the well-established MNIST dataset [4].

Keywords: Prediction confidence, Deep Learning, Conformal Prediction

1 Introduction

How confident are Deep Learning predictions? In most cases, we assume that a model performs as well on new data as it has on average in the past [5–8]. The Conformal Prediction (CP) framework presents an enticing alternative, providing per-prediction confidence based on statistical hypothesis testing [9]. Given a user-specified significance level, CP guarantees a corresponding maximum error rate with more relaxed assumptions than commonly assumed in Deep Learning [1]. However, guaranteed absolute validity comes with computational efficiency drawbacks, making CP in its original form unrealistic for large-scale datasets [2].

Several proposals have been made in recent years that successfully address CP challenges, although they tend to introduce a trade-off with other model characteristics [10]. For example, inductive variants remove the need for repeated leave-one-out training [11]. Aggregated CP (ACP) improves predictive efficiency (i.e., the precision of the predictions), at the cost of training multiple ensemble models and loosing absolute validity [3]. However, to the best of our knowledge, such approaches maintain the underlying two-step CP algorithm: calculating data strangeness with an intermediate non-conformity measure, and subsequently transforming scores into p-values.

Distribution approximation provides an interesting avenue to address challenges related to the two-step nature of the Conformal Prediction algorithm. In this paper, we propose a one-step CP approximation approach with a novel, model-agnostic conformal loss function. By evaluating deviation of the model output from the expected output distribution, we may circumvent the algorithmic constraints of the CP framework. Deep Learning is especially promising for this approach because the model's versatility and potential to model complex data relationships is well-established [12–14].

1.1 Contributions

To extend CP usability on real-world datasets, we make the following contributions:

- *We propose a unique loss function that approximates traditional two-step CP in only one step* (Sections 2 and 3).
  By enabling a Deep Learning (DL) neural network to learn the direct relationship between input data and the conformal p-values, we may skip the intermediate non-conformity scores altogether.
- *We carry out a rigorous and comprehensive evaluation of our proposed method for binary and multi-class classification on the well-known MNIST dataset* (Section 4).
  Empirical analysis of a DL model trained with our loss function confirms competitiveness with Aggregated Conformal Prediction (ACP) in terms of approximate validity and predictive efficiency. However, our model is significantly more computationally efficient than ACP.

- *Finally, our one-step approach to CP approximation introduces new opportunities for CP improvement* (Section 5).
  By modelling the expected output distribution directly, our novel loss function approach circumvents current CP algorithmic constraints. This is of particular research interest, as CP performance optimisation is traditionally a challenging task due to the interactions between the underlying model and the non-conformity measure [15].

The code and results described in this article are fully available in a GitHub repository[1].

2 Conformal Prediction

This section outlines the CP background which the article builds on.

2.1 Background

The CP framework answers a ubiquitous question in Machine Learning: *How confident are we that a model's prediction is correct?* In this article, the focus lies on a computationally efficient variant, Inductive Conformal Prediction [16]. Interested readers may refer to [9] for a detailed description and context of the original transductive approach.

Under minimal data assumptions, CP provides guaranteed confidence for individual predictions, based on statistical hypothesis testing [1]. Specifically, it guarantees that a predictor makes errors only up to a maximum user specified error rate ϵ if the input data is exchangeable [17]. CP achieves this by outputting prediction sets with all plausible labels for classification problems (interested readers may refer to [18] for regression). An error occurs when the true label y_i is not included in the CP prediction set $\Gamma_i^{1-\epsilon}$. By the law of large numbers, the probability of an error occurring approaches the upper limit ϵ as the number of predictions grows (Equation (1)), subject to statistical fluctuations [19].

$$Pr(y_i \notin \Gamma_i^{1-\epsilon}) \leq \epsilon \tag{1}$$

In short, the CP framework acts as a wrapper for any prediction model, also called the underlying model [20]. Predictions are transformed into p-values $p_n^{y^*}$ that describe the likelihood of sample x_i's extension with each possible label $y_i^* \in Y$, denoted $z_i = (x_i, y_i^*)$. The computationally efficient inductive CP variant requires three sub-datasets. Let the proper training set be $(x_1, ..., x_n)$, the calibration set $(x_{n+1}, ..., x_m)$, and the test set $(x_{m+1}, ..., x_l)$. For each sample extension z_i, the CP model first evaluates how strange it is compared to the known training data [21]. The extension's 'strangeness', or non-conformity, is measured with a score $\alpha_i^{y^*}$ as a function of the extended sample z_i and its true label y_i. An example of a straightforward but versatile non-conformity

[1] https://github.com/juliameister/dl-confident-loss-function

measure (NCM) is the Margin Error function A_{ME}, given in Equation (3). A non-conforming example has a low probability estimate for the true label y_i, and/or a high probability estimate for the false label $y_i^* \neq y_i$ [22].

$$\alpha_i^{y^*} = A(z_i, y_i) \tag{2}$$

$$A_{ME}(z_i, y_i) = 0.5 - \frac{\max_{y_i \neq y_i^*} Pr(y_i|x_i) - Pr(y_i^*|x_i)}{2} \tag{3}$$

For inductive CP, the model is fit to the data just once [23]. This includes training the underlying algorithm with the proper training set $(x_1, ..., x_n)$, and calculating the NCM scores $\alpha_{n+1}^{y_{n+1}}, ..., \alpha_m^{y_m}$ for the calibration set $(x_{n+1}, ..., x_m)$. Given a test score $\alpha_i^{y_i^*}$, we calculate the p-values $p_i^{y_i^*}$ for each $z_i \in (z_{m+1}, ..., z_l)$, as shown in Equation (4). The numerator is also known as a 'rank'. The larger the rank, the larger the p-value is, and the more conforming a sample extension z_i is, compared to the training and calibration sets [24]. Furthermore, Equation (4) guarantees that all p-values corresponding to the true labels y_i are uniformly distributed in $\mathcal{U}_{[0,1]}$ [25].

$$p_i^{y^*} = \frac{|\{j = n+1, ..., m : \alpha_j^{y^*} \leq \alpha_i^y\}| + 1}{|\{j = n+1, ..., m : y_j^*\}| + 1} \tag{4}$$

Finally, given a test sample's p-values $p_{m+1}^{y_i^*}, y_i^* \in Y$, its CP prediction set Γ_i is obtained with Equation (5). The optimal and most precise CP classification output is a prediction set with exactly one class label.

$$\Gamma_i^{1-\epsilon} = \{y^* \in Y | p_i^{y^*} > \epsilon\} \tag{5}$$

$$|\Gamma_{optimal}^{1-\epsilon}| = 1 \tag{6}$$

Note that, since the error rate and its inverse accuracy are automatically guaranteed by the validity characteristic, CP performance evaluation is traditionally based on predictive efficiency, i.e., prediction set size [26]. Section 4.4 discusses metrics that may be used to evaluate CP performance.

2.2 Related work in conformal validity approximation

Optimising CP for both classification and regression tasks is an active research area [27–29] because the framework has promising applications in high-risk, confidence-sensitive settings. CP is versatile and a small selection of use-cases include pharmaceutical drug discovery [30] respiratory health monitoring [31], and financial risk prediction [32].

Since prediction accuracy is automatically guaranteed for traditional 'full' or Transductive CP (TCP), the primary optimisation objective is to increase predictive efficiency by outputting more precise prediction ranges (Section 2.1). Unfortunately, optimisation is a difficult task. There are two related but

sometimes conflicting steps which must be considered simultaneously: the calibration of the underlying predictive model and the non-conformity measure. While promising approaches have been proposed that take both steps into account [15, 33, 34], they tend to be specific to certain underlying models and not generally applicable.

Additionally, TCP has computational efficiency challenges built into the algorithm and is not scalable to large, real-world datasets [2]. CP variants that address the inefficiency of the transductive leave-one-out retraining approach tend to accept a trade-off with other limitations [10]. For example, Inductive Conformal Prediction (ICP) trades predictive efficiency for increased computational efficiency [11].

In contrast, Aggregated Conformal Prediction (ACP) improves on predictive efficiency by limiting prediction set sizes [3]. It is a generalisation of previously proposed CP variants (Cross-conformal and Bootstrap conformal predictors, introduced in [35]), inspired by ML prediction ensembles to reduce variance. For each test sample and possible label, the p-values p_n^* generated by N ensemble models are aggregated into one ACP p-value (e.g. with Equation (7), [36]). The intermediate p-values p_n^* are calculated as shown in Section 2.1.

$$p_{ACP} = \frac{1}{N} \sum_{n=1}^{N} p_n^* \tag{7}$$

However, more precise ACP predictions come at the cost of losing automatic validity. While exact validity may be achieved in some situations with more stringent requirements, approximate validity is observed empirically [37]. In particular, ACP tends to be conservative for low significance levels and invalid for high significance levels, because p-values close to the mean tend to be more common [10]. There have been some effective proposals to recover validity [38–40] but these approaches do not address the significantly increased training times for multiple ensemble models compared to one ICP model.

In a recent and particularly relevant work, non-conformity scores were directly approximated by estimating their influence on the underlying model's loss with Influence Functions [41]. The purpose was to make the algorithm scalable to large datasets. The method successfully achieves validity on par with TCP, while removing the need for leave-one-out training procedure for every test point. As the number of training samples increased, the authors observed an increasingly minor approximation error. In empirical experiments, the error became negligible at 10,000 training samples.

This confirms that validity and improved computational efficiency are possible by approximating individual steps of the TCP algorithm. In this article, we propose a novel conformal loss function that approximates the entire TCP two-step procedure by teaching a Deep Learning model the direct relationship between input data and conformal p-values.

3 Approximating conformal p-values directly with Deep Learning

Deep Learning (DL) is a highly versatile framework that automatically identifies and learns relationships between input data and output data, by extracting informative data representations [12]. A neural network h with weights θ is made up of one or more hidden and interlinked layers (Equation (8)). Each hidden neuron Z_i in a layer performs a simple linear transformation on the incoming data vector X with weight vector θ_i, offset by an absolute bias B_i, as shown in Equation (9) [42]. An activation function (e.g. ReLU) may then be applied to model non-linear data relationships.

$$\hat{y} = h_\theta(X) \tag{8}$$

$$Z_i = B_i + \theta_{i,1} \cdot X_1 + \theta_{i,2} \cdot X_2 + ... + \theta_{i,j} \cdot X_j \tag{9}$$

Supervised training may be considered as an optimisation function with which the model attempts to minimise the deviation of its current output prediction $\hat{y}$ to the expected output label y [43]. For DL models trained with gradient descent, the difference between them is quantified via a loss function f (Equation (10)).

$$C = f(\hat{y}, y) \tag{10}$$

The loss C is then back-propagated through the model. In Equation (11), each weight θ_i is updated with $\Delta\theta_i$ in proportion to its negative gradient [44], reducing the output's deviation from the true labels in the next iteration.

$$\Delta\theta_i = -\frac{\partial C}{\partial \theta_i} \tag{11}$$

Once training is complete, a well-calibrated model will have optimised its learned weights θ so that Equation (8) accurately reflects the relationship between the input and its expected output label.

DL's versatility and ability to represent complex data relationships makes it a promising choice to model conformal p-values directly from the input, skipping the non-conformity scores (see Section 2). We hypothesise that a function composition of the two standard CP transformation steps (from data input to non-conformity measures, and from non-conformity measures to p-values) could be approximated with a suitably designed and trained model. *In other words, the right DL architecture and loss function may be able to model the complex relationship between the input and conformal p-value output directly.*

DL architectures have been used in the past to predict probability distributions of the target variable, e.g. with Mixture Density Networks [45–47]. However, to our knowledge, there is currently no model architecture (e.g., activation function) or loss function that can guarantee a uniform model output

distribution, which is required for true-class conformal p-values (Section 2.1). Therefore, we propose an architecture-agnostic conformal loss function which ensures that the model output approximates the uniform distribution $\mathcal{U}_{[0,1]}$.

The purpose of our novel loss function is to achieve approximate validity and high precision, while significantly reducing the high algorithmic and computational complexity of similar CP variants. For example, ACP uses an ensemble approach to improve predictive efficiency [3]. However, as the number of ensemble models grows, computational time increases significantly. Additionally, approximate validity becomes weaker because the distribution of aggregated p-values shifts away from $\mathcal{U}_{[0,1]}$ (e.g., to a unimodal Bates distribution with the aggregation procedure shown in Equation (7)) [10].

In contrast, a single DL model with our proposed conformal loss function is competitive with ACP up to 10 ensemble models in terms of approximate validity and high predictive efficiency, while significantly reducing on algorithmic complexity and training time. Sections 3.1 and 3.2 describe the background and components of our proposed method in detail, and Section 4 provides a rigorous empirical evaluation of our results on the well-established MNIST datasets [4].

3.1 Proposed method requirements and constraints

This article's contribution is to develop a unique conformal loss function for Deep Learning (DL). The novelty of our approach is that we simplify the two-step CP algorithm (Section 2) to only one step. Our loss function approximates conformal p-values directly from the input data, skipping the intermediate non-conformity measure. Therefore, the neural network's output should follow the same distribution requirements as CP p-values:

- To *interpret the neural network output as conformal p-values*, predictions should fall in the range (0, 1), and class outputs for one sample should be independent (Section 2.1).
- To *ensure validity and consequently a guaranteed error rate*, approximated p-values of the true class should be uniformly distributed in $\mathcal{U}_{[0,1]}$ [10].
- To *maintain high predictive efficiency*, false-class approximated p-values should be close to 0 [1].

Additionally, we must work within the constraints enforced by DL:

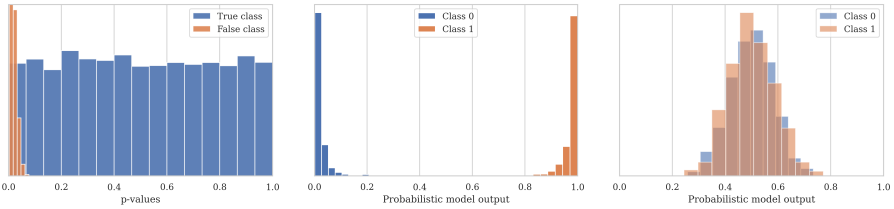
- The *loss function should be differentiable* for effective backpropagation with gradient descent [12].
- The *trade-off between model genralisability* (small batch size, [48]) and *output distribution evaluation precision* (large batch size, [49]) should be considered.

3.2 Proposed conformal approximation loss function and background

To approximate conformal p-values directly from the input data, we propose a conformal loss function that is compatible with a probabilistic, binary

classification neural network (architecture details in Section 4). Given the requirements of the CP framework (Section 2), we know that well-calibrated CP classifiers are marked by a uniform distribution of the true-class p-values, and a distribution peaking at 0 with little variance for false-class p-values [10], as visualised in Figure 1a. In contrast, typical well-calibrated neural network (NN) classifiers have a distinctive maximum distance between true-class (peak at 1) and false-class (peak at 0) output distributions, as shown in Figure 1b. The challenge for the conformal loss function is to accurately describe the CP output distribution, so that the neural network may emulate it, avoiding the normal-like distribution that tends to occur when under-fit models make average random guesses (Figure 1c). Therefore, we propose a loss function that is made up of two components, optimising the true-class and false-class target distributions respectively:

$$loss_{conformal} = loss_{false} + loss_{true} \tag{12}$$



(a) Well-calibrated CP. (b) Well-calibrated NN. (c) Underfit NN.

Fig. 1: Characteristic Neural Network (NN) and Conformal Prediction (CP) output distributions of binary probabilistic classifiers.

Since approximated p-values of the false class should follow the same pattern as traditional NN false-class predictions, we calculate standard Binary Cross Entropy (BCE) to minimise the values towards 0. Equation (13) defines BCE as a function of the model's predictions $\hat{y}$ and the true labels y for N training samples [50]. For the purposes of the conformal loss function, we calculate BCE only for the false-class approximated p-value predictions $\hat{y}_{false}$ (Equations (14) and (15)).

$$BCE(y, \hat{y}) = -\frac{1}{N}\sum_{n=1}^{N}(y_n \cdot \log \hat{y}_n + (1 - y_n) \cdot \log(1 - \hat{y}_n)) \tag{13}$$

$$\hat{y}_{false} = [n = 1, ..., N, y_n \in \mathbf{Y}^{\mathbb{N}} : \hat{y}_{nc} \text{ s.t. } c \neq y_n] \tag{14}$$

$$loss_{false} = BCE(0, \hat{y}_{false}) \tag{15}$$

Unlike traditional classification, the true labels $y \in \mathbf{Y}^{\mathbb{N}}$ are not the target model output in CP. Instead, true-class predictions $\hat{y}_{true}$ (Equation (16))

should follow the distribution $\mathcal{U}_{[0,1]}$ over N samples (Section 2). As a consequence, the loss component $loss_{true}$ should evaluate and quantify the distance of the model output distribution from the target uniform distribution, rather than the distance to a concrete label per sample. Note that the trade-off between large sample counts for distribution estimation [51] and small batch-sizes for optimised learning [48] must be carefully considered.

Even though function moments do not necessarily uniquely characterise a distribution, they may be a sufficient heuristic for a distribution distance evaluation [52]. Lower order moments require less samples to estimate, and are therefore suitable for the proposed true-class conformal loss component, shown in Equation (10). When $loss_{true}$ is minimised, the distribution of $\hat{y}_{true}$ approximates $\mathcal{U}_{[0,1]}$. In Section 4, we empirically confirm that our one-step approximation approach is competitive with ACP in terms of uniformity and predictive efficiency, and significantly improves the computational efficiency.

$$\hat{y}_{true} = [n = 1, ..., N, y_n \in \mathbf{Y}^{\mathbb{N}} : \hat{y}_{nc} \text{ s.t. } c = y_n] \tag{16}$$

$$loss_{true} = w_1 \cdot loss_{mean} + w_2 \cdot loss_{var} + w_3 \cdot loss_{l2} + w_4 \cdot loss_{huber} \tag{17}$$

Including the *mean* μ of true-class model outputs $\hat{y}_{true}$ ensures that the output distribution is centred around 0.5 (Equation (19)). Because model training minimises the overall loss, we square and take the root, so that any deviation is represented by a positive value. A larger value indicates a greater distance between the model's output distribution and the expected mean of the uniform distribution $\mathbb{E}[\mathcal{U}_{[0,1]}] = 0.5$.

$$\mu(x) = \frac{\sum_{n=1}^{N} x_n}{N} \tag{18}$$

$$loss_{mean} = \sqrt{(\mu(\hat{y}_{true}) - 0.5)^2} \tag{19}$$

Similarly, for the *variance* σ^2, the $loss_{var}$ component regulates the dispersion of $\hat{y}_{true}$ so that it matches the expected value $\text{Var}[\mathcal{U}_{[0,1]}] = \frac{1}{12}$.

$$\sigma^2(x) = \frac{\sum_{n=1}^{N}(x_n - \mu)^2}{N} \tag{20}$$

$$loss_{var} = \sqrt{\left(\sigma^2(\hat{y}_{true}) - \frac{1}{12}\right)^2} \tag{21}$$

To measure uniformity while maintaining differentiability (i.e., without ranking or sorting), we leverage the work in [53]. The authors suggested that for an unknown distribution, measuring sample collisions with moments (such as l_p-norms) was a successful approximation for uniformity testing.

We choose the l_2-*norm* as our heuristic, since it is fully differentiable and sensitive to outliers, i.e., it will penalise large differences more than smaller ones [54]. In Equation (23), $\hat{y}_{true}$ is normalised to ensure that minimising

the loss component increases the uniformity of the model output distribution. Note that $loss_{l2}$ does not necessarily measure uniformity in the range $(0, 1)$, only the distribution probability's smoothness between the input vector's extreme elements. The distribution's mean, variance, and the next loss component $loss_{huber}$ counteract this drawback by encouraging values throughout the entire range.

$$l_2(x) = \sqrt{\sum_{n=1}^{N} (x_n)^2} \tag{22}$$

$$loss_{l_2} = l_2 \left(\frac{\hat{y}_{true}}{\Sigma \hat{y}_{true}} \right) \tag{23}$$

Traditionally, *Huber loss* (Equation (24)) is used to narrow the prediction region to the true label. In Equation (25), we negate the value instead to disincentivise the trend to normal-like distributions on the average output δ of under-fit models. The parameter $\alpha \in \mathbb{R}^+$ regulates the threshold for the transition between quadratic and linear [54]. Since the target distribution for $\hat{y}_{true}$ is $\mathcal{U}_{[0,1]}$, we expect the over-represented prediction to be $\delta \approx 0.5$ (see Section 4). As with previous loss components, Huber loss is fully differentiable and robust to outliers.

$$Huber_\alpha(x) = \begin{cases} \frac{1}{2}x^2, & |x| \leq \alpha \\ \alpha(|x| - \frac{1}{2}\alpha), & |x| > \alpha \end{cases} \tag{24}$$

$$loss_{huber} = -Huber_\alpha(\delta, \hat{y}_{true}) \tag{25}$$

Algorithm 1 illustrates how the described loss components come together to produce our novel conformal loss function (also shown in Equation (10)). The function is fully differentiable and therefore may be used in combination with any backpropagation model. All individual evaluation metrics are inbuilt into `tensorflow`, meaning that our proposed loss is compatible with any `tensorflow` model. Algorithm 2 shows the `Python 3.9` implementation as a `tensorflow v2.4` custom loss function.

Algorithm 1 Proposed conformal loss function components. By evaluating the deviation from the expected Conformal Prediction output distributions (Section 2.1), the neural network model may approximate conformal p-values directly from the input.

Require: $\hat{y} \in (0,1),\ y \in \mathbb{N},\ \theta_{huber} = (\alpha, \delta),\ \theta = (\theta_0, \theta_1, ..., \theta_5)$

1: $\hat{y}_{false} \Leftarrow [n = 1, ..., N : \hat{y}_{nc} \text{ s.t. } c \neq y_n]$
2: $\hat{y}_{true} \Leftarrow [n = 1, ..., N : \hat{y}_{nc} \text{ s.t. } c = y_n]$
3: $loss_{false} = BCE(zerosLike(\hat{y}_{false}), \hat{y}_{false})$
4: $loss_{mean} \Leftarrow \sqrt{(mean(\hat{y}_{true}) - 0.5)^2}$
5: $loss_{var} \Leftarrow \sqrt{(var(\hat{y}_{true}) - 1/12)^2}$
6: $loss_{l2} \Leftarrow l2(y_{true}/\Sigma\hat{y}_{true})$
7: $loss_{huber} \Leftarrow -Huber_{\alpha}(\delta, \hat{y}_{true})$
8: $loss_{true} \Leftarrow \theta_2 \cdot loss_{mean} + \theta_3 \cdot loss_{var} + \theta_4 \cdot loss_{l2} + \theta_5 \cdot loss_{huber}$
9: $loss_{conformal} \Leftarrow \theta_0 \cdot loss_{false} + \theta_1 \cdot loss_{true}$

Algorithm 2: Conformal loss implementation of Algorithm 1 in Python 3.9. All classes and methods were imported from tensorflow v2.4.

```
def conformal_loss(y_true, y_pred, alpha, delta, ws):
    # label formatting
    y_onehot = one_hot(reshape(y_true, [-1], depth=
        n_class)
    mask_true = cast(y_onehot, bool)
    mask_false = logical_not(mask_true)
    y_true = reshape(boolean_mask(y_pred, mask_true,
        axis=0), [-1, 1])
    y_false = reshape(boolean_mask(y_pred, mask_false,
        axis=0), [-1, n_class-1])

    # calculating loss components
    loss_false = BinaryCrossentropy(
        zeros_like(y_false), y_false)
    loss_l2 = l2_loss(y_true/reduce_sum(y_true))
    loss_mean = sqrt(square(reduce_mean(y_true)-0.5))
    loss_var = sqrt(square(reduce_variance(y_true)
        -1/12))
    loss_huber = -Huber(alpha)(fill(shape(y_true),
        delta), y_true)
    loss_true = w_l2 * loss_l2 + w_h * loss_huber +
        w_m * loss_mean + w_v * loss_var

    return w_t * loss_true + w_f * loss_false
```

4 Empirical results

This section comprehensively evaluates the empirical performance of our proposed one-step conformal loss function (Section 3.2). In particular, we compare the validity, predictive, and computational efficiency to the well-established ACP approximation technique (Section 2.1) for both binary and multi-class classification.

4.1 Research questions

The following research questions informed and structured the performance evaluation of our novel conformal loss function:

- Can we directly model the relationship between the input data and the conformal p-values, skipping the intermediate non-conformity measure?
- Is our proposed approach competitive with the established ACP approximation in terms of validity and predictive efficiency?
- What benefits may approximating CP with a single-step approach have, compared to the traditional two-step calculations?

4.2 The MNIST dataset

We rigorously evaluate our proposed method on the well-established benchmarking MNIST dataset [4]. The dataset is an extensive collection of 70,000 images, each with a handwritten digit between 0-9. The images are in greyscale, formatted as 28x28 binary vectors (see Figure 2).

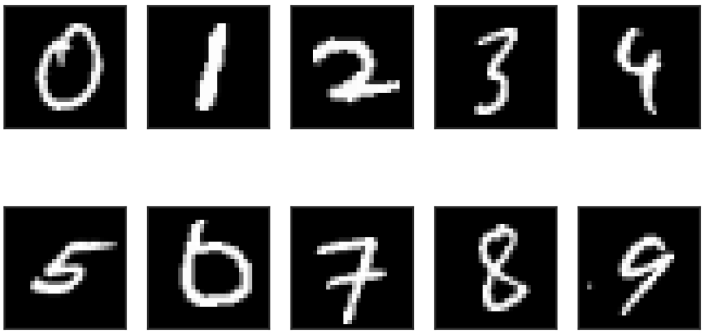


Fig. 2: Examples of MNIST handwritten digits.

Since we work with a linear feedforward network (detailed in Section 4.3), each image is flattened to a feature vector with dimensions (784 x 1). No further data processing is applied. We evaluate our proposed method for both binary classification (images of 0 and 1) and multi-class classification with all 10 digits. The sample counts and class distributions are reported in Table 1. To reduce spurious noise in the results, the same train/test split was used for

all test runs. Following insights from [41] that CP approximation may become more successful with larger training sets, we chose a 85%/15% split.

Table 1: MNIST sample counts for classification. A relatively large proportion of samples is assigned to the training set following observations that larger quantities of training samples may achieve better CP approximation [41].

Samples	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	$\sum$
Train	5,923	6,742	5,958	6,131	5,842	5,421	5,918	6,265	5,851	5,949	60,000 (85.71%)
Test	980	1,135	1.032	1,010	982	892	958	1,028	974	1,009	10,000 (14.29%)
$\sum$	6,903	7,877	6,990	7,141	6,824	6,313	6,876	7,293	6,825	6,958	70,000 (100.0%)

4.3 Neural network architecture and hyperparameter optimisation

We opt for a shallow feedforward neural network (NN) architecture as shown in Figure 3. Similar architectures have been previously successfully used for MNIST classification [55–57], supporting our decision for a simpler image classification model compared to the traditional Convolutional Neural Networks [58–60]. A test with the defined architecture, no hyperparameter optimisation, and standard Sparse Categorical Cross Entropy shows that the model has enough complexity to accurately represent and learn the data relationships, see Figure 4.

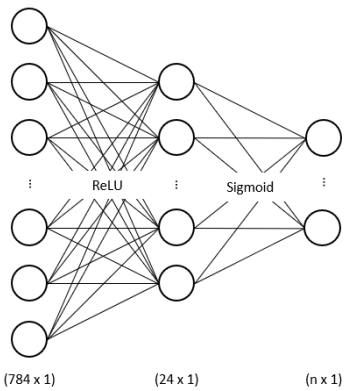
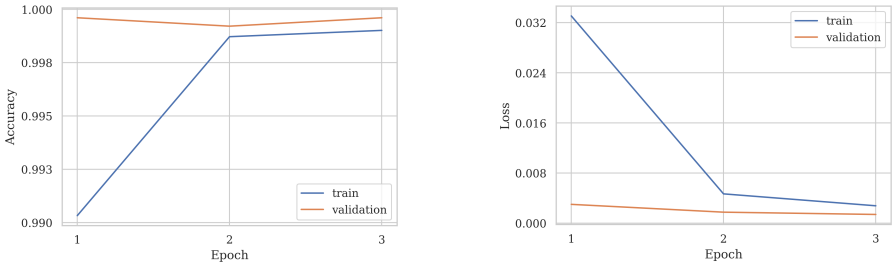


Fig. 3: Neural network feedforward architecture. Our proposed conformal loss function requires the final activation to be sigmoid, and the number of output neurons to match the number of classes n.



(a) NN accuracy.

(b) NN loss.

Fig. 4: Training statistics of the neural network visualised in Figure 3 for a standard MNIST classification task. The model has enough complexity to accurately represent relationships between the input data and class labels.

There are two simple architecture requirements for compatibility with our conformal loss: The final activation must be a sigmoid function, which limits the output to the range (0, 1) (see Section 2.1); And the number of output neurons n must be equal to the number of classes. Since we are interested in binary and multi-class classification, we use two models: one with $n = 2$, and one with $n = 10$.

The model's optimal hyperparameters were identified with a gridsearch based on the uniformity of the true-class output (Kolmogorov-Smirnov, see Section 4.4). The parameter values were as follows:

- For the NN: optimiser = adam, learning rate = 0.001, batch size = 128, epochs = 3.
- For the conformal loss function (see Equations (12) and (17) in Section 3.2):

$$loss = 1 \cdot loss_{false} + 1 \cdot loss_{true} \tag{26}$$

$$loss_{true} = 1 \cdot (loss_{mean} + loss_{var}) + 5 \cdot loss_{l2} + 0.25 \cdot loss_{huber}(\delta = 0.125) \tag{27}$$

4.4 Evaluation metrics

CP related methods requires unique metrics for performance evaluation, since validity is guaranteed unlike traditional Machine Learning models (Section 2). Instead, we employ an intuitive performance measure relates to the size of the CP models prediction sets [61]. For classification, the larger the output sets Γ_n are, the less precisely the model has predicted a sample n's target y_n [31]. Relevant CP evaluation metrics are defined in Table 2.

For CP approximation as with our proposed loss function, validity is not guaranteed but instead approximated. As a result, characteristics related to the success of the approximation should be measured in addition. This includes the Kolmogorov-Smirnov test for uniformity [62] and the miscalibration rate

(distance between the expected and achieved error curves) to measures the p-value distributions.

Table 2: Evaluation metrics for Conformal Prediction (CP) predictive efficiency, and distribution metrics to evaluate CP approximation.

Metric	Formula	Intuition
Error rate	$(\sum_{n=1}^{N} y_n \notin \Gamma_n)/N$	True label not in prediction set.
Empty rate	$(\sum_{n=1}^{N} \lvert\Gamma_n\rvert = 0)/N$	Outlier samples.
Single rate	$(\sum_{n=1}^{N} \lvert\Gamma_n\rvert = 1)/N$	Efficient predictions.
Multi rate	$(\sum_{n=1}^{N} \lvert\Gamma_n\rvert > 1)/N$	Samples on the class boundary.
KS-test	$D = \max \lvert P - \mathcal{U}\rvert$	Statistical uniformity test for distribution P.
Miscalib. rate	$error_\epsilon - \epsilon$	Distance between error line and expected error.

4.5 Empirical evaluation of the proposed conformal loss function

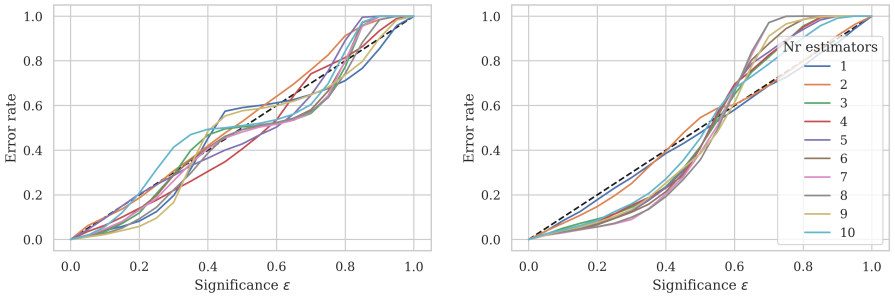
For the most direct comparison between our conformal loss function (Section 3.2) and Aggregated Conformal Prediction (ACP, Section 2.1), we use the same feedforward neural network (NN) model in both tests: as a standalone Deep Learning (DL) model and as the underlying model for ACP, since any point predictor may be used (Section 2). The ACP p-value aggregation procedure and non-conformity-measure are given in Equations (3) and (7) respectively.

Ten iterations were evaluated in both scenarios. For DL, each iteration represents an independent run and the average is reported; For ACP, each run had a different number of ensemble models from $1 < n < 10$. Each NN model was trained with our novel conformal loss function after parameter optimisation (Section 4). The models were evaluated with metrics suited to Conformal Prediction, as described in Section 4.4. We start with a detailed overview of the binary classification results, and then explore how the increase of classes and training samples affected the model performance. Overall, our analysis had a stronger focus on calibration adherence at lower significance levels $\epsilon \in \{0.05, 0.1, 0.2\}$, since low error rates are especially relevant to most prediction tasks with confidence requirements.

Calibration

Figure 5 confirms that the error rate of the CP approximation with our proposed NN loss function roughly followed the calibration curve (dashed line) for binary classification. Unlike ACP, which with $n > 2$ has a very distinctive S-shaped curve (as previously noted in [10]), our proposal seemed to conform much more closely to the ideal validity curve at any given point. While NN

error rates were conservative at low significance ($\epsilon < 0.3$), the gap was overall smaller than ACP with $n > 2$.



(a) 10 CP approximation NNs.

(b) ACP with $1 \leq n \leq 10$.

Fig. 5: The calibration curves to evaluate approximate validity of NN models trained with our conformal loss function compared to ACP. Both methods tend to be conservative for low significance levels.

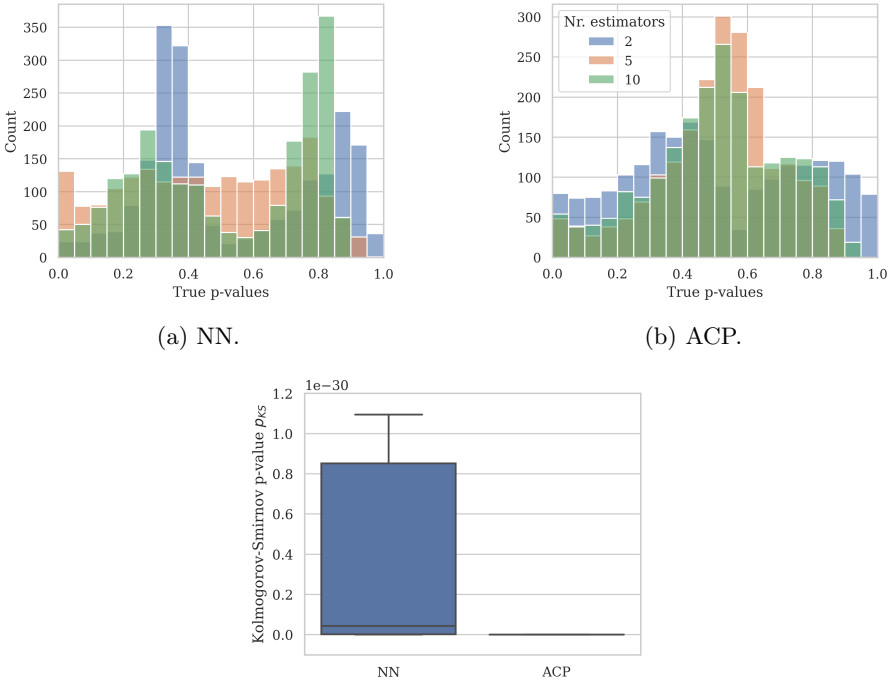
Hence, the results confirm that our conformal loss achieved approximate validity directly, without computing intermediate non-conformity measures. The calibration curve deviations and their concrete effects will be examined in more detail when we evaluate the calibration/efficiency trade-off later in this section.

Approximate validity

To preserve validity, p-values of the true class should follow the uniform distribution $\mathcal{U}_{[0,1]}$. On the other hand, false-class p-values should be as close to 0 as possible to maximise predictive efficiency (Section 3.1). An NN trained with our conformal loss function tended to a bi-modal distribution of true-class p-values, dipping around 0.5 (Figure 6), which may be a side-effect of the Huber loss component. Inversely, ACP tended to a unimodal distribution, peaking at 0.5 (Figure 6b). However, we note that $n = 2$ was also bimodal and distinctly similar to the NN trends.

These findings are directly related to the calibration lines and explain previous observations in Figure 1: NN with true p-values concentrated towards the extremes tended to be slightly conservative at lower ($\epsilon < 0.3$) and higher ($\epsilon > 0.7$) significance levels. In contrast, ACP $n > 1$ had a distinctive S-shape and was significantly conservative until $\epsilon = 0.5$, after which the calibration curve crossed the diagonal and became invalid.

Neither method passed the Kolmogorov-Smirnov test of uniformity [62], since all p-values p_{KS} were lower than our significance level $\alpha_{KS} = 0.01$ (Figure 6c). However, NN p_{KS} were distributed across a larger range than ACP $n > 2$ (p_{KS} has an inverse relationship with n), which shows that our



(a) NN.

(b) ACP.

(c) Uniformity test.

Fig. 6: True p-value distribution of three randomly selected NN models and three representative CP models. Neither model type achieves the expected uniform distribution $\mathcal{U}_{[0,1]}$, which is statistically confirmed with the Kolmogorov-Smirnov test for uniformtiy.

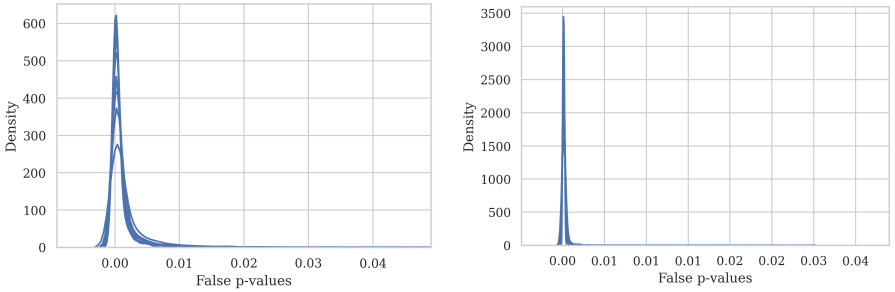
method has the potential to achieve stronger approximate validity with future loss function optimisations.

Predictive efficiency

Figure 7 shows that the false-class p-values were centred around 0 for both methods. Surprisingly, increasing the number of models n for ACP did not seem to improve predictive efficiency in this case, as all distribution curves were almost indistinguishably overlapped (Figure 7b). While NN false p-values had slightly more spread up to 0.0005 (Figure 7a), the values were still orders of magnitude under the minimum $\epsilon = 0.05$ threshold considered in this study, and therefore did not affect the prediction set sizes.

Calibration vs predictive efficiency trade-off

After examining the p-value distributions, we may more meaningfully compare our proposed one-step conformal p-value approximation to traditional ACP



(a) NN. (b) ACP.

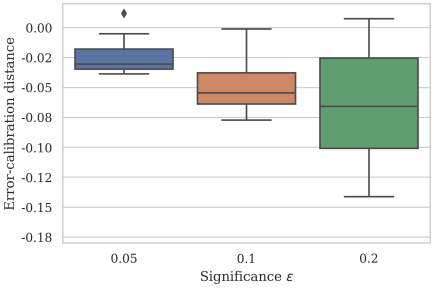
Fig. 7: False p-value distribution of the NN and ACP model types. P-values produced by our conformal loss function are negligably larger than ACP. However, the predictive efficiency is not affected, since they remain smaller than the lowest considered significance level $\epsilon = 0.05$.

on a more holistic level. We evaluated the predictive efficiency gains in the context of distance of the error line to the calibration curve.

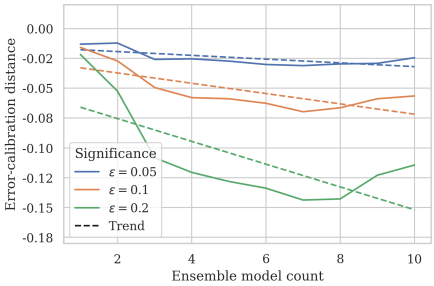
As discussed in Section 2, ACP improves predictive efficiency by increasing set sizes towards the optimal as the number of ensemble models grows, with the side-effect of a weaker validity approximation (Figure 8b). Although the effects were relatively minor for small significance levels ($\epsilon = 0.05$), the positive efficiency and negative calibration trends became much more pronounced as ϵ grew.

Promisingly, the average calibration of our conformal loss NN with $\epsilon = 0.05$ was competitive with the equivalent ACP trend line across all n, and the range improved the distance towards 0 in some iterations (Figure 8a). At $\epsilon = 0.1$, NN again improved on ACP, with the mean equivalent to $n = 5$ and some iterations reaching almost perfect calibration (calibration distance ≈ 0). The largest improvement on average was achieved at $\epsilon = 0.2$, which on average significantly outperformed the ACP equivalent for $n > 2$.

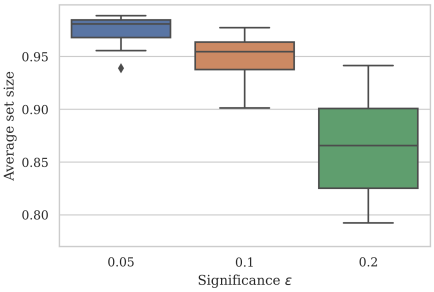
In predictive efficiency, we may confirm again that our proposed loss function is competitive with ACP without major deficits. A CP's optimal prediction set size is 1, and Figures 8c and 8d highlight that both models performed very well, especially at low significance levels (NN=0.97 and ACP=0.96-0.97 for $\epsilon = 0.05$. As expected, ACP average set sizes improved as the number of ensemble models n increased. Similarly to the calibration distance, the trend slopes were shallower at low significance levels, and taking the calibration trade-off by increasing n had a higher value of return at higher significance levels. Similarly, the NN model's predictive efficiency also showed larger ranges and variability as significance levels increased. This supports our previous assessment that the marginally increased p-values compared to ACP do not affect predictive efficiency.



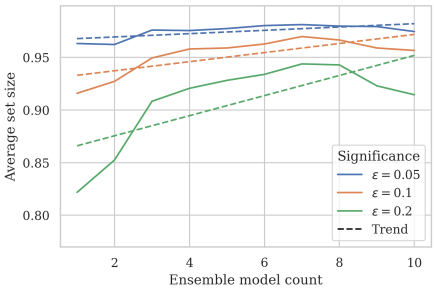
(a) NN miscalibration rate.



(b) ACP miscalibration rate.



(c) NN average prediction set size.



(d) ACP average prediction set size.

Fig. 8: Calibration vs predictive efficiency trade-off. Note that the optimal miscalibration rate is 0, and the optimal predictive efficiency is achieved with sets including exactly one label. Interestingly, the competitiveness of our method increased with the significance level. As ϵ grew, the average calibration improvement of our model increased and was on par with ACP at lower n.

Standard CP performance metrics in Table 3 confirm our observations so far. Both NN and ACP were conservative for low significance levels, although ACP achieved error scores closer to the expected maximum. Additionally, the minor increase in NN false p-values was negligible even at $\epsilon = 0.05$, all significance levels show 0% multi-sets and high single set rate. In combination with lower error rates, we conclude that our proposed conformal loss function successfully trains a simple NN model to confidently assign only the correct class to the vast majority of samples, without first calculating an additional non-conformity measure (Section 3.2).

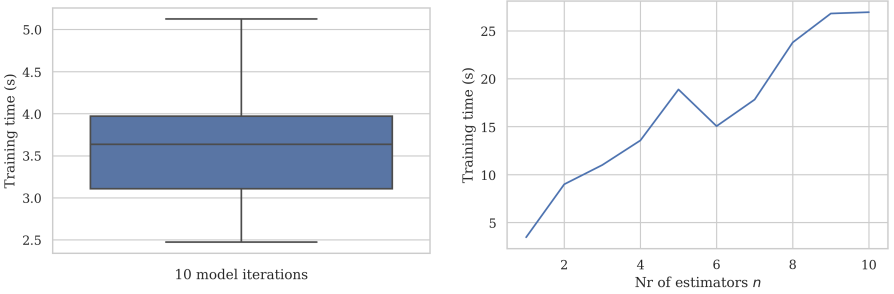
Computational efficiency

After confirming that our proposed conformal loss function successfully maintains approximate validity and predictive efficiency on par with ACP for $3 \leq n \leq 10$, we evaluate the two methods' computational efficiency. This is where our model has two advantages: It skips the intermediate non-conformity measure calculation, and requires training only one model. In contrast, ACP's

Table 3: Overall Conformal Prediction approximation performances. NN results are averaged across 10 iterations. n is the number of ACP ensemble classifiers. Results are given in %.

	NN			**ACP,** $n = 2$			**ACP,** $n = 5$			**ACP,** $n = 10$		
ϵ	**0.05**	**0.1**	**0.2**	**0.05**	**0.1**	**0.2**	**0.05**	**0.1**	**0.2**	**0.05**	**0.1**	**0.2**
Error	2.61	5.31	13.58	6.19	9.88	18.63	4.49	9.50	20.38	1.89	5.77	20.76
Empty	2.59	5.31	13.58	6.10	9.88	18.63	4.44	9.50	20.38	1.89	5.77	20.76
Single	97.40	94.69	86.42	93.90	90.12	81.37	95.56	90.50	79.62	98.11	94.23	79.24
Multi	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

training time increases linearly as the number of ensemble models n increases. As a consequence, all test iterations of our proposed model significantly outperformed ACP with $n > 1$ (3.5 seconds on average compared to up to over 25 seconds). Depending on n and parallel computing capabilities, the training time gap may be narrowed, but ACP would nonetheless require significantly more computational power to train its ensemble models.



(a) NN. (b) ACP.

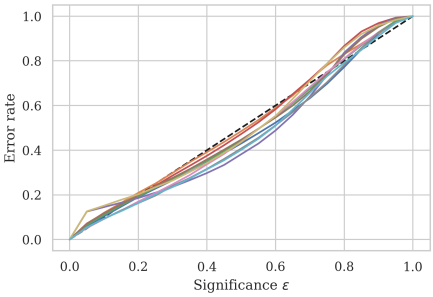
Fig. 9: Training times for the NN and ACP models. NN has a significant benefit, as only one model is trained in each iteration. The trianing time reduction by NN grwos as the number of ACP ensemble models increases, up to 86% when $n = 10$.

Increasing the number of classes and samples

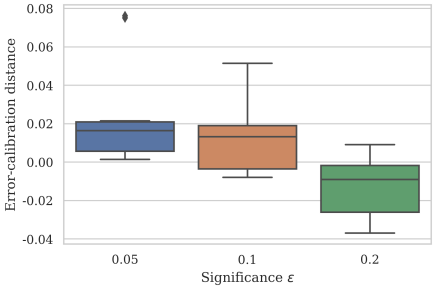
Figure 10 presents or proposed method's performance on the entire MNIST dataset. This involves differentiating between 10 classes (digits 0 to 9) instead of two (0, 1), and a consequent increase of training samples form around 13,000 to 60,000 (see Table 1).

The calibration curves became visibly closer to the diagonal (exact validity), smooth, and consistent between model iterations compared to binary classification (Figure 5a). Additionally, all ten models achieve very close to

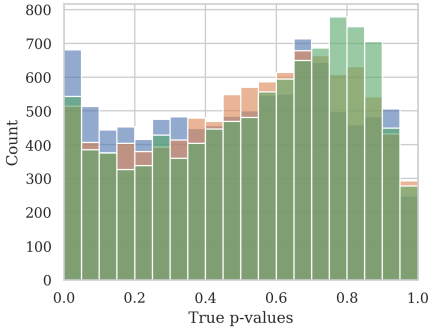
exact validity for $0.1 \leq \epsilon \leq 0.3$. The smoothness of the calibration curves was a reflection of the true-class p-value distribution, which more closely followed the uniform distribution $\mathcal{U}_{[0,1]}$.



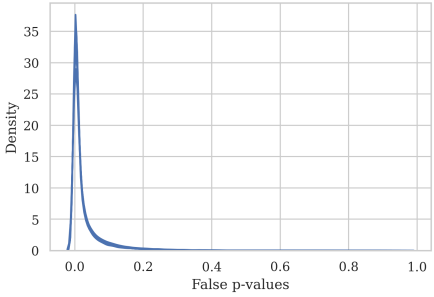
(a) NN calibration.



(b) NN calibration distance.



(c) NN true-class p-values.



(d) NN false-class p-values.

Fig. 10: 10-class classification with our proposed conformal loss function. The increased training sample count from around 13,000 with binary classification to 60,000 significantly improved our models' calibration and consequently their approximate validity significantly.

5 Discussion

We presented a comprehensive empirical evaluation of our proposal on the well-established MNIST benchmark dataset [4]. Results across ten experiment iterations found that our conformal loss function is competitive with ACP for approximate validity and predictive efficiency. The crucial difference is that our direct approach needs only one model and therefore has significantly improved computational efficiency, solving a long-standing challenge with ensemble CP variants. The computational benefit increases proportionally as the number of ensemble models n in ACP grows.

Our novel loss function minimises the difference between the model output with the expected CP output distribution (Section 3). Because the false-class p-values follow the same trend as false-class probability outputs in standard DL classification (values close to 0), our loss component $loss_{false}$ is very successful. Consequently, we achieve very high predictive efficiency and are competitive with ACP up to $n = 10$ ensemble models at low significance levels (Figure 8c). The optimal and most precise prediction for CP models is $|\Gamma^{1-\epsilon}| = 1$ (Section 2.1), and our conformal DL models achieve on average $|\Gamma^{1-0.05}| = 0.97$, exactly on par with ACP ($n = 10$) at $epsilon = 0.05$.

In terms of true-class outputs, we achieve approximate validity on par with ACP for $n > 5$ (Table 3), measured with the Kolmogorov-Smirnov test for uniformity [62]. ACP tends to be conservative for low significance levels and invalid for higher levels [10]. In contrast, our proposed method has a less-smooth calibration line, which is however on average closer to the expected diagonal at most individual points (Figure 5a).

A limitation of our work is how the loss component $loss_{true}$ evaluates the output distribution's uniformity. To maintain differentiability, we measure the deviation through distribution moments (Section 3) which are not necessarily unique to the expected distribution $\mathcal{U}_{[0,1]}$. Empirically, we achieve approximate uniformity and therefore approximate validity, but the distribution measure may be improved on in future work. However, this limitation may be significantly reduced by increasing the number of training samples, as shown in Figure 10.

6 Conclusion and future directions

We propose a novel conformal loss function which directly approximates the two-step CP framework for classification with one conformal loss function. By learning to output predictions in the distribution characteristic to CP (Section 2) from the input data, models trained with our loss function skip the intermediate non-conformity score, hence, reducing the inherent algorithmic complexity. The loss function is fully differentiable and compatible with any gradient descent-based Deep Learning neural network (Section 3.2). Our novel approach is most successful for small significance levels, which in practice are of high interest to guarantee low error rates. Additionally, our direct conformal p-value prediction has the potential to simplify CP optimisation.

We carried out an extensive and rigorous empirical evaluation of our proposed method on the benchmark MNIST dataset [4], with three main takeaways, with respect to the proposed research questions:

- Our proposal is competitive with Aggregated Conformal Prediction (ACP), a common CP variant that has been successful on real world datasets with high predictive efficiency [3, 36, 37]. We achieve results on par with ACP for both approximate validity and predictive efficiency.
- Notably, our conformal loss function significantly improves on ACP's computational efficiency without compromising on CP performance. We train

only one model, compared to n ACP ensemble models. As n increases, the computational savings of our method grow proportionally (ca. x% per extra model).
- Finally, our one-step CP approximation reduces the algorithmic complexity of the traditionally two-step CP framework. This provides a new avenue for CP optimisation, which is traditionally a difficult task since the interaction between the underlying algorithm and the non-conformity measure have to be considered simultaneously [15, 33, 34].

Future work includes improving the output uniformity measure in the conformal loss function to be more precise (Section 5), which would address the approximation techniques conservative tendencies. Additionally, our empirical study may be extended with a theoretical evaluation and optimisation of the loss function for gradient descent to improve model convergence.

Acknowledgements. This research is funded by University of Brighton's 'Connected Futures' initiative, and the 'Rising Stars' research grant.

Declarations

The dataset analysed during the current study is publicly available at the following link: http://yann.lecun.com/exdb/mnist/.

References

[1] Shafer, G., Vovk, V.: A tutorial on conformal prediction. Journal of Machine Learning Research **9**(3) (2008)

[2] Riquelme-Granada, N., Nguyen, K., Luo, Z.: Coreset-based conformal prediction for large-scale learning. In: Conformal and Probabilistic Prediction and Applications, pp. 142–162 (2019). PMLR

[3] Norinder, U., Boyer, S.: Conformal prediction classification of a large data set of environmental chemicals from ToxCast and Tox21 estrogen receptor assays. Chemical research in toxicology **29**(6), 1003–1010 (2016)

[4] Deng, L.: The MNIST database of handwritten digit images for machine learning research. IEEE Signal Processing Magazine **29**(6), 141–142 (2012)

[5] Rechkemmer, A., Yin, M.: When confidence meets accuracy: Exploring the effects of multiple performance indicators on trust in machine learning models. In: CHI Conference on Human Factors in Computing Systems, pp. 1–14 (2022)

[6] Zhang, T., Sun, M., Cremer, J.L., Zhang, N., Strbac, G., Kang, C.: A confidence-aware machine learning framework for dynamic security

assessment. IEEE Transactions on Power Systems **36**(5), 3907–3920 (2021)

[7] Meister, J.A., Nguyen, K.A., Luo, Z.: Audio feature ranking for sound-based COVID-19 patient detection. In: Progress in Artificial Intelligence (2022). Springer

[8] Yin, M., Wortman Vaughan, J., Wallach, H.: Understanding the effect of accuracy on trust in machine learning models. In: Proceedings of the 2019 Chi Conference on Human Factors in Computing Systems, pp. 1–12 (2019)

[9] Vovk, V., Gammerman, A., Shafer, G.: Algorithmic Learning in a Random World. Springer, US (2005)

[10] Linusson, H., Norinder, U., Boström, H., Johansson, U., Löfström, T.: On the calibration of aggregated conformal predictors. In: Conformal and Probabilistic Prediction and Applications, pp. 154–173 (2017). PMLR

[11] Papadopoulos, H., Vovk, V., Gammerman, A.: Conformal prediction with neural networks. In: 19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2007), vol. 2, pp. 388–395 (2007). IEEE

[12] LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)

[13] Maskara, N., Kubica, A., Jochym-O'Connor, T.: Advantages of versatile neural-network decoding for topological codes. Physical Review A **99**(5), 052351 (2019)

[14] Khatri, N., Khatri, K.K., Sharma, A.: Prediction of effluent quality in iceas-sequential batch reactor using feedforward artificial neural network. Water science and technology **80**(2), 213–222 (2019)

[15] Cherubin, G., Chatzikokolakis, K., Jaggi, M.: Exact optimization of conformal predictors via incremental and decremental learning. In: International Conference on Machine Learning, pp. 1836–1845 (2021). PMLR

[16] Vovk, V.: Conditional validity of inductive conformal predictors. In: Asian Conference on Machine Learning, pp. 475–490 (2012). PMLR

[17] Gupta, C., Kuchibhotla, A.K., Ramdas, A.: Nested conformal prediction and quantile out-of-bag ensemble methods. Pattern Recognition **127**, 108496 (2022)

[18] Johansson, U., Boström, H., Löfström, T., Linusson, H.: Regression conformal prediction with random forests. Machine learning **97**(1), 155–176

(2014)

[19] Fisch, A., Schuster, T., Jaakkola, T., Barzilay, R.: Few-shot conformal prediction with auxiliary tasks. In: International Conference on Machine Learning, pp. 3329–3339 (2021). PMLR

[20] Löfström, T., Johansson, U., Boström, H.: Effective utilization of data in inductive conformal prediction using ensembles of neural networks. In: The 2013 International Joint Conference on Neural Networks (IJCNN), pp. 1–8 (2013). IEEE

[21] Norinder, U., Boyer, S.: Binary classification of imbalanced datasets using conformal prediction. Journal of Molecular Graphics and Modelling **72**, 256–265 (2017)

[22] Johansson, U., Linusson, H., Löfström, T., Boström, H.: Model-agnostic nonconformity functions for conformal classification. In: 2017 International Joint Conference on Neural Networks (IJCNN), pp. 2072–2079 (2017). IEEE

[23] Johansson, U., Boström, H., Löfström, T.: Conformal prediction using decision trees. In: 2013 IEEE 13th International Conference on Data Mining, pp. 330–339 (2013). IEEE

[24] Vovk, V.: Transductive conformal predictors. In: IFIP International Conference on Artificial Intelligence Applications and Innovations, pp. 348–360 (2013). Springer

[25] Angelopoulos, A.N., Bates, S., Jordan, M., Malik, J.: Uncertainty sets for image classifiers using conformal prediction. In: International Conference on Learning Representations (2020)

[26] Krstajic, D.: Critical assessment of conformal prediction methods applied in binary classification settings. Journal of Chemical Information and Modeling **61**(10), 4823–4826 (2021)

[27] Sesia, M., Romano, Y.: Conformal prediction using conditional histograms. Advances in Neural Information Processing Systems **34**, 6304–6315 (2021)

[28] Papadopoulos, H., Vovk, V., Gammerman, A.: Regression conformal prediction with nearest neighbours. Journal of Artificial Intelligence Research **40**, 815–840 (2011)

[29] Linusson, H.: Nonconformity measures and ensemble strategies: An analysis of conformal predictor efficiency and validity. PhD thesis, Department of Computer and Systems Sciences, Stockholm University (2021)

[30] Eklund, M., Norinder, U., Boyer, S., Carlsson, L.: The application of conformal prediction to the drug discovery process. Annals of Mathematics and Artificial Intelligence **74**(1), 117–132 (2015)

[31] Nguyen, K.A., Luo, Z.: Cover your cough: Detection of respiratory events with confidence using a smartwatch. In: Conformal and Probabilistic Prediction and Applications, pp. 114–131 (2018). PMLR

[32] Wisniewski, W., Lindsay, D., Lindsay, S.: Application of conformal prediction interval estimations to market makers' net positions. In: Conformal and Probabilistic Prediction and Applications, pp. 285–301 (2020). PMLR

[33] Makili, L., Vega, J., Dormido-Canto, S.: Incremental Support Vector Machines for fast reliable image recognition. Fusion Engineering and Design **88**(6-8), 1170–1173 (2013)

[34] Papadopoulos, H., Gammerman, A., Vovk, V.: Reliable diagnosis of acute abdominal pain with conformal prediction. Engineering Intelligent Systems **17**(2), 127 (2009)

[35] Vovk, V.: Cross-conformal predictors. Annals of Mathematics and Artificial Intelligence **74**(1), 9–28 (2015)

[36] Carlsson, L., Eklund, M., Norinder, U.: Aggregated conformal prediction. In: IFIP International Conference on Artificial Intelligence Applications and Innovations, pp. 231–240 (2014). Springer

[37] Wilm, A., Norinder, U., Agea, M.I., de Bruyn Kops, C., Stork, C., Kühnl, J., Kirchmair, J.: Skin Doctor Cp: conformal prediction of the skin sensitization potential of small organic molecules. Chemical Research in Toxicology **34**(2), 330–344 (2020)

[38] Solari, A., Djordjilović, V.: Multi split conformal prediction. Statistics & Probability Letters **184**, 109395 (2022)

[39] Toccaceli, P., Gammerman, A.: Combination of inductive mondrian conformal predictors. Machine Learning **108**(3), 489–510 (2019)

[40] Balasubramanian, V.N., Chakraborty, S., Panchanathan, S.: Conformal predictions for information fusion. Annals of Mathematics and Artificial Intelligence **74**(1), 45–65 (2015)

[41] Abad, J., Bhatt, U., Weller, A., Cherubin, G.: Approximating full conformal prediction at scale via influence functions. arXiv preprint arXiv:2202.01315 (2022)

[42] Ghalambaz, M., Noghrehabadi, A., Behrang, M., Assareh, E., Ghanbarzadeh, A., Hedayat, N.: A hybrid neural network and gravitational search algorithm (HNNGSA) method to solve well known Wessinger's equation. International Journal of Mechanical and Mechatronics Engineering **5**(1), 147–151 (2011)

[43] Olivé, D.M., Huynh, D.Q., Reynolds, M., Dougiamas, M., Wiese, D.: A supervised learning framework for learning management systems. In: Proceedings of the First International Conference on Data Science, E-Learning and Information Systems, pp. 1–8 (2018)

[44] Lillicrap, T.P., Santoro, A., Marris, L., Akerman, C.J., Hinton, G.: Backpropagation and the brain. Nature Reviews Neuroscience **21**(6), 335–346 (2020)

[45] Makansi, O., Ilg, E., Cicek, O., Brox, T.: Overcoming limitations of mixture density networks: A sampling and fitting framework for multimodal future prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7144–7153 (2019)

[46] Li, C., Lee, G.H.: Generating multiple hypotheses for 3d human pose estimation with mixture density network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9887–9895 (2019)

[47] Zhang, H., Liu, Y., Yan, J., Han, S., Li, L., Long, Q.: Improved deep mixture density network for regional wind power probabilistic forecasting. IEEE Transactions on Power Systems **35**(4), 2549–2560 (2020)

[48] Kandel, I., Castelli, M.: The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset. ICT express **6**(4), 312–315 (2020)

[49] Hao, Y., Orlitsky, A.: Doubly-competitive distribution estimation. In: International Conference on Machine Learning, pp. 2614–2623 (2019). PMLR

[50] Ho, Y., Wookey, S.: The real-world-weight cross-entropy loss function: Modeling the costs of mislabeling. IEEE Access **8**, 4806–4813 (2019)

[51] Jaki, T., Kim, M., Lamont, A., George, M., Chang, C., Feaster, D., Van Horn, M.L.: The effects of sample size on the estimation of regression mixture models. Educational and Psychological Measurement **79**(2), 358–384 (2019)

[52] Malz, A., Marshall, P., DeRose, J., Graham, M., Schmidt, S., Wechsler, R., Collaboration, L.D.E.S., *et al.*: Approximating photo-z PDFs for large

surveys. The Astronomical Journal **156**(1), 35 (2018)

[53] Batu, T., Canonne, C.L.: Generalized uniformity testing. In: 2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS), pp. 880–889 (2017). IEEE

[54] Meyer, G.P.: An alternative probabilistic interpretation of the Huber loss. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5261–5269 (2021)

[55] de Chazal, P., Tapson, J., van Schaik, A.: A comparison of extreme learning machines and back-propagation trained feed-forward networks processing the MNIST database. In: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2165–2168 (2015). https://doi.org/10.1109/ICASSP.2015.7178354

[56] Lejeune, E.: Mechanical MNIST: A benchmark dataset for mechanical metamodels. Extreme Mechanics Letters **36**, 100659 (2020)

[57] Gabella, M.: Topology of learning in feedforward neural networks. IEEE Transactions on Neural Networks and Learning Systems **32**(8), 3588–3592 (2020)

[58] Kayed, M., Anter, A., Mohamed, H.: Classification of garments from fashion MNIST dataset using CNN LeNet-5 architecture. In: 2020 International Conference on Innovative Trends in Communication and Computer Engineering (ITCE), pp. 238–243 (2020). IEEE

[59] Kadam, S.S., Adamuthe, A.C., Patil, A.B.: CNN model for image classification on MNIST and fashion-MNIST dataset. Journal of scientific research **64**(2), 374–384 (2020)

[60] Garg, A., Gupta, D., Saxena, S., Sahadev, P.P.: Validation of random dataset using an efficient CNN model trained on MNIST handwritten dataset. In: 2019 6th International Conference on Signal Processing and Integrated Networks (SPIN), pp. 602–606 (2019). IEEE

[61] Ashby, A.E., Meister, J.A., Nguyen, K.A., Luo, Z., Gentzke, W.: Cough-based COVID-19 detection with audio quality clustering and confidence measure based learning. In: Conformal and Probabilistic Prediction with Applications (2022). PMLR

[62] Zhang, G., Wang, X., Liang, Y.-C., Liu, J.: Fast and robust spectrum sensing via kolmogorov-smirnov test. IEEE Transactions on Communications **58**(12), 3410–3416 (2010)