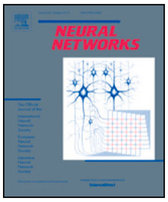
Neural Networks 168 (2023) 194–205

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet



Reliable prediction intervals with directly optimized inductive conformal regression for deep learning

Haocheng Lei, Anthony Bellotti [*]

School of Computer Science, University of Nottingham Ningbo China, 199 Taikang East Road, Ningbo, 315100, Zhejiang, China

ARTICLE INFO

Keywords:
Conformal prediction
Deep learning
Prediction intervals
Neural networks
Uncertainty estimation

ABSTRACT

By generating prediction intervals (PIs) to quantify the uncertainty of each prediction in deep learning regression, the risk of wrong predictions can be effectively controlled. High-quality PIs need to be as narrow as possible, whilst covering a preset proportion of real labels. At present, many approaches to improve the quality of PIs can effectively reduce the width of PIs, but they do not ensure that enough real labels are captured. Inductive Conformal Predictor (ICP) is an algorithm that can generate effective PIs which is theoretically guaranteed to cover a preset proportion of data. However, typically ICP is not directly optimized to yield minimal PI width. In this study, we propose Directly Optimized Inductive Conformal Regression (DOICR) for neural networks that takes only the average width of PIs as the loss function and increases the quality of PIs through an optimized scheme, under the validity condition that sufficient real labels are captured in the PIs. Benchmark experiments show that DOICR outperforms current state-of-the-art algorithms for regression problems using underlying Deep Neural Network structures for both tabular and image data.

1. Introduction

Deep Neural Networks (DNNs) have achieved remarkable performance in various application fields in recent years, making them popular machine learning algorithms. This success is typically measured using aggregate measures on the accuracy of point predictions, such as Mean Square Error or R^2 for regression problems. However, in many real-world problems, such as autonomous driving (Deruyttere et al., 2021), finance systems (Hansen & Borch, 2021), and medical diagnostics (Zhou et al., 2021), it is not only the point predictions that matter, but also the measure of uncertainty at the individual prediction level. For high-risk applications, incorrect predictions can have a significant negative impact; and hence understanding the uncertainty of each individual prediction becomes crucial. This can assist users in making more favorable judgments and even decide whether to disregard the model's predictions based on the computed uncertainty, leaving high-risk decisions to humans (Geifman & El-Yaniv, 2017). Therefore, in many applications, there is a need to quantify the uncertainty of each prediction (Krzywinski et al., 2013). One way to achieve this is to provide a prediction interval (PI), instead of a simple point prediction. The width of the PI gives a measure of uncertainty: the wider the interval the more uncertain we consider the predictor for that particular example. In this study, the Inductive Conformal Predictor (ICP) framework for regression problems is utilized, since it gives a theoretical guarantee of validity under mild exchangeability conditions (Vovk et al., 2005). A general method for Directly Optimized Inductive Conformal Regression (DOICR) is proposed based on a DNN model structures and compared against traditional ICP and two other alternatives proposed in the literature: Quality-Driven-soft and Surrogate Conformal Predictor Optimization (SCPO). The DOICR method is found to be superior across multiple data sets. Similar work for classification problems can also be found, although the problem for classification, which requires prediction sets, instead of intervals, is somewhat different and presents its own challenges (Bellotti, 2021b; Stutz et al., 2022).

The main contributions of this paper are:

1. Presenting a novel approach employing PI width as a sole loss function in PIs optimization for the first time, leveraging the guaranteed validity offered by ICP;
2. Advancing beyond conventional ICP frameworks by introducing a neural network (NN) architecture capable of directly generating valid prediction intervals and efficiently integrating cutting-edge deep learning models;
3. Executing deep learning models thoroughly trained using the DOICR loss function;
4. Conducting a comprehensive comparative analysis of DOICR against various alternative PI generators based on NNs across multiple datasets.

[*] Corresponding author.
E-mail address: anthony-graham.bellotti@nottingham.edu.cn (A. Bellotti).

https://doi.org/10.1016/j.neunet.2023.09.008
Received 31 January 2023; Received in revised form 9 August 2023; Accepted 3 September 2023
Available online 19 September 2023

The remainder of this paper is organized as follows. Section 2 provides background and motivation for the problem. In Section 3, the algorithms QD-soft, ICP, and SCPO are introduced. Section 4 details how to construct our proposed method, DOICR. The experimental setup and the performance of the four different algorithms on six public datasets will be presented in Section 5, and then, conclusions will be made in Section 6. Since multiple topic areas are covered, the extensive use of technical abbreviations cannot be avoided. A glossary is provided in Appendix to assist the reader.

2. Background

Outputting PIs instead of point predictions is an intuitive way to quantify uncertainty. PIs convey uncertainty directly, providing a lower and an upper bound with a certain probability that the target value is within this interval. A simple example: in a house price prediction problem, a traditional neural network will only output a point prediction of say £300,000, but PIs may give a price range of say £280,000 to £320,000 with a probability of 80%, say, that the true label of the house price is within this range. The probability of coverage is usually set by the user in advance. This predetermined probability is referred to as the *confidence level* (CL) in this paper. Although the use of PIs can greatly assist people to make better decisions, it is challenging to make traditional deep learning models such as Neural Networks (NNs) and Convolutional Neural Networks (CNNs) output PIs because they usually only make point predictions. There has been some research to improve traditional machine learning models so that they can output high-quality PIs. For those algorithms that can generate PIs, we refer to them collectively as PI generators.

Generally, when assessing PI generators, two qualities are important:

1. Predictive efficiency: on average, the PIs are as narrow as possible, and
2. Validity: the observed probability that the true outcome value is within the PI is in accordance with the user-defined confidence level.

The term *predictive efficiency* is used in the conformal prediction literature to broadly refer to the effectiveness of prediction sets and is related to smaller prediction set size; e.g. Johansson et al. (2014), Shafer and Vovk (2008) and Vovk et al. (2016). For PIs, it can be measured using Mean Prediction Interval Width ($MPIW$). For the second quality, the Prediction Interval Coverage Probability ($PICP$) is used. The closer that $PICP$ is to CL, the closer to meeting validity. Both measures are defined in the next section. We expect a PI generator to provide valid PIs with high predictive efficiency. These two criteria are called High-Quality principles by Pearce et al. (2018) .

The focus of this study is exploring PI generators for popular deep learning NNs using fixed NN structure and gradient descent, that have been widely adopted in recent years. Nonetheless, there have been advances with dynamic NNs that have a flexible structure, notably in Stochastic Configuration Networks (Wang & Li, 2017) and 2D-Stochastic Configuration Networks (Li & Wang, 2021). These networks are distinct in that they do not rely on gradient descent, and they offer unparalleled adaptability and the ability to dynamically alter their structures during training. The dynamic NNs are an interesting development that could also be extended further by considering reliable PIs. For example, based on Stochastic Configuration Networks, Lu et al. (2020) as well as Lu and Ding (2019) have achieved robust and comparable performance with mixed distribution and bootstrap ensemble methods that take the quality of PIs as the training objective.

A variety of methods have been proposed to measure uncertainty of NNs from Bayesian Neural Networks (MacKay, 1992) to the Bootstrap method (Heskes, 1996). However, these algorithms entail a significant computational burden. For instance, the Bootstrap method gauges model uncertainty by training numerous NNs on diverse resampled versions of the training dataset, each with distinct parameter initializations. This approach of training multiple NNs considerably escalates the computational expense. Similarly, Bayesian Neural Networks present a challenge due to the time-consuming nature of running Markov Chain Monte Carlo (MCMC) simulations for training. The swelling number of parameters in current large DNNs necessitates considerably more training time compared to traditional NNs. Regardless of whether one is training multiple DNNs or employing MCMC for training, the computational expense for DNNs with high parameter counts becomes substantially greater. Consider an application example: when predicting bone age from X-ray images, the integration of PIs can enhance the practical utility of the model in the decision-making process (Bagnall & Davis, 2014). However, when training with state-of-the-art CNNs, which are more accurate but notably time-consuming, the use of methods like Bootstrap and Bayesian Neural Networks may not be the best choice, given their substantial time consumption.

Consequently, in our study, we focus on the generation of high-quality PIs as an optimization problem. To be specific, our attention is concentrated on algorithms capable of directly producing high-quality PIs after a single training session, without the need for bootstrapping or MCMC. Furthermore, these algorithms can be optimized using gradient descent methods, which are the optimizers predominantly employed by complex DNNs. When training such algorithms, the accuracy of the point estimate will not be the goal, and optimizing the quality of the PIs is seen as the aim of the training. There are already a number of methods that focus on the optimization of PIs for DNNs. By directly incorporating the High-Quality principles into the loss function, Pearce et al. (2018) proposed the Quality-Driven-soft (QD-soft) method, and achieves excellent performance. Like QD-soft, many subsequent studies have built their loss function on the basis of the $MPIW + \lambda PICP$ structure (Bellotti, 2020; Lai et al., 2022), where λ is a balancing factor controlling the importance of $PICP$ during the optimization. This is an intuitive approach with a central idea of minimizing the loss function such that the $PICP$ is as close as possible to CL and the $MPIW$ is minimized. However, such loss functions are usually not stable, since the convergence direction of gradient descent is highly uncertain, the optimization process may be biased to optimize only $PICP$ or $MPIW$, and the final results do not guarantee that $PICP$ are close to the predetermined confidence level. In other words, these methods are not valid PI generators. Also, the hyperparameter search for λ may require a lot of work for ensuring that the resulting $PICP$ conforms to the High-Quality principles. The experimental results also show that the $PICP$ of PIs generated by QD-soft could deviate seriously from the confidence level from time to time. However, the DOICR method proposed in this paper moves away from the strategy of jointly optimizing $MPIW$ with $PICP$ by only taking the $MPIW$ as the loss function. To be specific, DOICR is a PI generator that ensures the validity of predictions.

The reason why DOICR can be optimized for $MPIW$ only is that ICP is itself able to theoretically guarantee the validity under mild exchangeability conditions (Shafer & Vovk, 2008; Vovk et al., 2005). In a traditional ICP for regression, two machine learning models that can perform point prediction need to be trained: model m for predicting the target, and σ for modeling the uncertainty of the prediction in m. Then, ICP will utilize the output of the two models to generate PIs. Generally speaking, therefore, traditional ICP is a wrapper on top of already trained models. Typically, traditional ICP for regression performs well, but the underlying models have not been optimized as PI generators, hence we expect they will not perform so well on such problems, relative to algorithms that have (Bellotti, 2020). Interestingly, ICP can be defined without requiring specific underlying models, as it only necessitates a model structure. Provided the exchangeability assumption is satisfied, the PIs are ensured to be valid, albeit with potentially poor predictive efficiency due to arbitrary model parameters (i.e., the predetermined confidence level will be maintained even if the underlying machine learning models of the ICPs are untrained or poorly trained, but at the expense of potentially wide PIs). A concise proof of this concept was provided by Toccaceli (2022) . The DOICR exploits this feature by exploring possible ICPs across their parameter space to find the one with low $MPIW$.

The DOICR is a general method and can not only be used in simple Multilayer Perceptron (MLP), but can also be easily combined with CNNs and other complex DNN structures. In contrast to QD-soft and SCPO, DOICR does not require any extra hyperparameters inside the loss function and is not prone to computational problems (e.g. divide by zero) during the training process. Moreover, it is perhaps more intuitive and easier to understand. Extensive experimental results demonstrate that DOICR outperforms previous algorithms when used with both MLP and CNN model structures.

3. Related methodology

After some preliminary notation is provided and $PICP$ and $MPIW$ are defined, the two related algorithms, ICP and QD-soft, are introduced.

A data set with n instances is given with ith input features and target are denoted as $\mathbf{x}_i$ and y_i respectively, for $i \in \{1, \dots, n\}$. A PI can be expressed by upper and lower bounds, $\hat{y}_{l_i}$ and $\hat{y}_{u_i}$. According to the first term of the High-Quality principles, PIs need to include as many data points as possible at a pre-defined confidence level, $(1-\varepsilon)$, which can be expressed as follows:

$$Pr(\hat{y}_{l_i} \le y_i \le \hat{y}_{u_i}) \ge (1-\varepsilon) \tag{1}$$

This is also known as the *validity* property in the conformal prediction literature.

The intention is that the confidence level is set by the end-user of the system. In this paper, the notation utilized to represent it is $(1-\varepsilon)$, since this is used extensively in the ICP literature. To illustrate the meaning of the confidence level, consider the following example for medical diagnosis.

> Prior to using a PI generator, a medical doctor sets her confidence level to 0.9. This will mean that she expects the PI to have at least 90% chance of being right: i.e. the PI will contain the correct outcome value at least 90% of the time. If the PI generator meets this requirement across multiple predictions, we say it is *valid*. The confidence level is entirely her choice. If she needs greater reliability, she can increase the confidence level. The ICP guarantees to be valid for any confidence level she chooses. However, this can only be achieved at the expense of generating wider PIs that are less informative.

The formal definition of $PICP$ and $MPIW$ can be given as

$$PICP := \frac{1}{n}\sum_{i=1}^{n} k_i \tag{2}$$

$$MPIW := \frac{1}{n}\sum_{i=1}^{n} \hat{y}_{u_i} - \hat{y}_{l_i} \tag{3}$$

where

$$k_i = \begin{cases} 1, & \text{if} \quad \hat{y}_{l_i} \le y_i \le \hat{y}_{u_i} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

Notice that $PICP$ can be expressed using Heaviside functions as

$$PICP := \frac{1}{n}\sum_{i=1}^{n} H(y_i - \hat{y}_{l_i}) \cdot H(\hat{y}_{u_i} - y_i) \tag{5}$$

3.1. QD-soft

The starting point of QD-soft is to use a loss function that jointly penalizes for deviations of $PICP$ from CL and large values of $MPIW$. However, since $PICP$ is composed of a series of Heaviside step functions, it has many discontinuities and cannot be optimized with gradient descent. This problem can be solved by approximating the step function with a sigmoid function $S(\gamma x)$, where $\gamma > 0$ is some softening factor. As a result, $PICP$ can be approximated by $PICP_{soft}$,

$$PICP_{soft} := \frac{1}{n}\sum_{i=1}^{n} S(\gamma(y_i - \hat{y}_{l_i})) \cdot S(\gamma(\hat{y}_{u_i} - y_i)) \tag{6}$$

For QD-Soft, only the efficiency of predictions for which the PI captures the target are considered in the loss function,

$$MPIW_{capt} := \frac{1}{c}\sum_{i=1}^{n} (\hat{y}_{u_i} - \hat{y}_{l_i}) \cdot k_i \tag{7}$$

where $c = \sum_{i=1}^{n} k_i$. Based on the two terms of the High-Quality principles, $Loss_{QD-soft}$ was built to optimize both $PICP$ and $MPIW$, where λ is a control hyperparameter for balancing the importance of the two principles.

$$Loss_{QD-soft} = MPIW_{capt} + \lambda \frac{n}{\varepsilon(1-\varepsilon)} \max\left(0, (1-\varepsilon) - PICP_{soft}\right)^2 \tag{8}$$

QD-soft has achieved good results with MLP as the model structure, but it also has the following drawbacks:

- The convergence direction of gradient descent is highly unstable, the optimization process may be biased towards optimizing only $PICP$ or $MPIW$, and the derived $PICP$ may not align closely with the confidence level. In other words, it may result in a $PICP$ much larger than the pre-defined confidence level and therefore too large $MPIW$, or a $PICP$ much smaller than the $(1-\varepsilon)$.
- $Loss_{QD-soft}$ itself is fragile in the training process and sensitive to the learning rate and decay rate, and it exhibits computational problems (divide by zero) (Pearce et al., 2018). This problem becomes especially severe when using larger model structures, such as ResNet.
- There are two built-in hyperparameters, λ and γ in $Loss_{QD-soft}$. Improper hyperparameter settings may result in failure to generate high-quality PIs. Therefore, it incurs further computational costs to search for good hyperparameter values.

3.2. Inductive conformal predictors (ICP)

Let $\mathbf{z}_1, \dots, \mathbf{z}_n$ represent a set of independent and identically distributed instances, where each instance $\mathbf{z}_i$ consists of input variables $\mathbf{x}_i$ and a target label y_i. For $1 \le k < l < n$, use 1 to k to index a training set, $(k+1)$ to l index a calibration set, and $(l+1)$ to n to index a test set. The nonconformity measure (NCM) of a pair $(\mathbf{x}, y)$ depends on itself and instances in the training set, which can be represented by a function $A(\mathbf{x}, y)$:

$$A(\mathbf{x}, y) = \mathcal{A}(\mathbf{z}_1, \dots, \mathbf{z}_k, (\mathbf{x}, y)) \tag{9}$$

such that the ordering of the calibration examples does not affect the function value. Let the NCM of ith instance be denoted as $\alpha_i = A(\mathbf{x}_i, y_i)$. Typically, we consider the NCM as based on a machine learning algorithm that is able to generate NCMs from the underlying training set.

The ICP prediction set for a new unlabeled example $\mathbf{x}$ at a confidence level of $(1-\varepsilon)$ is

$$\Gamma^{\varepsilon}(\mathbf{x}) = \left\{ y \in \mathbb{R} : \sum_{j=k+1}^{l} H\left(A(\mathbf{x}, y) - A(\mathbf{x}_j, y_j)\right) \right. \\ \left. +1 \le \varepsilon(l-k+1) \right\} \tag{10}$$

where H is the Heaviside step function. Assuming that all instances in $\{\mathbf{z}_{k+1}, \dots, \mathbf{z}_n\}$ are exchangeable, it can be shown that the generated prediction set satisfies

$$Pr(y_i \in \Gamma^{\varepsilon}(\mathbf{x}_i)) \ge 1 - \varepsilon \tag{11}$$

for all $i \in \{l+1, \dots, n\}$ (Papadopoulos et al., 2002). This result guarantees the validity of ICP since it states that the probability that the true

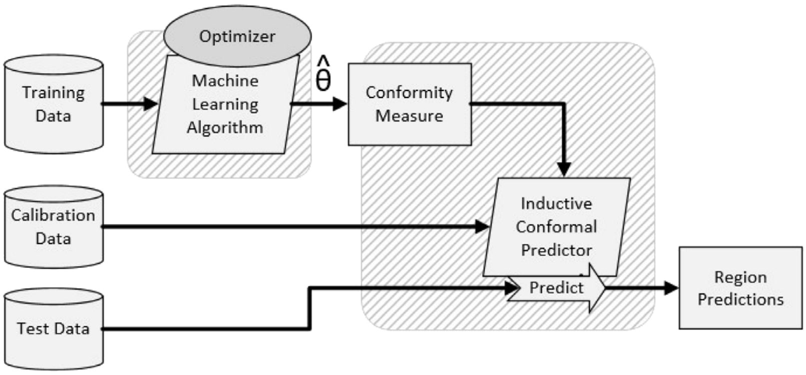


Fig. 1. Standard ICP framework.

label is in the prediction set is greater or equal to CL. The standard ICP framework is illustrated with training, calibration, and test sets in Fig. 1.

In this paper, the nonconformity measure we use is the normalized NCM,

$$\mathcal{A}(\mathbf{z}_1, \dots, \mathbf{z}_k, (\mathbf{x}, y)) = \frac{|y - m(\eta; \mathbf{x})|}{\sigma(\theta; \mathbf{x})} \tag{12}$$

where, in general, m and σ are two models with parameter vectors η and θ respectively. In general, m corresponds to a model of the point estimate of the target label and σ corresponds to a model of the uncertainty in that point estimate, and should take a positive value. Typically, σ can be a model of the absolute value of the residual of m. Parameter vectors η and θ need to be estimated. In this paper, NNs are used, in which case η and θ are the set of weights in the NNs.

Combining Eqs. (10) and (12), the prediction set becomes a PI with upper and lower bounds,

$$\Gamma^{\epsilon}(\mathbf{x}) = \left[\hat{y}_{l_i}, \hat{y}_{u_i}\right] \tag{13}$$

where

$$\hat{y}_{l_i} = m(\eta; \mathbf{x}_i) - q\sigma(\theta; \mathbf{x}_i) \tag{14}$$

$$\hat{y}_{u_i} = m(\eta; \mathbf{x}_i) + q\sigma(\theta; \mathbf{x}_i) \tag{15}$$

and q is the $(1-\epsilon)th$ quantile of NCMs in the calibration set, $\alpha_{k+1}, \dots \alpha_l$. Then the $MPIW$ of the test set under ICP is

$$MPIW_{NCM} = \frac{2q}{n-l} \sum_{i=l+1}^{n} \sigma(\theta; \mathbf{x}_i) \tag{16}$$

Since the range of function σ is positive real numbers, it is suitable to express it as the exponent of a function s with the range being all real numbers, i.e.

$$\sigma(\theta; \mathbf{x}_i) = \exp s(\theta; \mathbf{x}_i) \tag{17}$$

and this is the approach taken in this study.

The normalized NCM has proved effective in various models, including regression NNs (Johansson et al., 2014; Papadopoulos & Haralambous, 2011; Papadopoulos et al., 2002, 2011).

3.3. Surrogate Conformal Prediction Optimization (SCPO)

By utilizing the approach of QD-Soft and combining it with ICP, a form of PI generator can be developed which is approximately valid. This is done by approximating the exact validity requirement in the loss function by including the square loss of deviation of $PCIP$ from CL, whilst including the inefficiency term $MPIW$ across all examples. But as with, QD-Soft, to use gradient descent optimization, $PCIP$ cannot be used directly, hence the same soft approximation is used. This gives the loss function,

$$Loss_{SCPO} = PICP_{soft} + \lambda\ MIPW_{NCM} \tag{18}$$

where Eqs. (14) and (15) are used to construct the PIs. Minimizing the loss, with respect to η and θ will give an approximation to ICP, but is not guaranteed valid, hence the approach is called *Surrogate* Conformal Prediction Optimization (SCPO) (Bellotti, 2020). The output parameters from SCPO can then be passed to a proper ICP, with an independent calibration set, which guarantees validity.

SCPO was implemented with a simple underlying linear model and was shown to be successful and maintained validity (Bellotti, 2020). However, it shares problems with QD-soft: the gradient descent can be unstable leading to higher inefficient PIs, and it is sensitive to values of hyperparameters λ and γ. Generally, lower values of λ and higher values of γ give a closer approximation to ICP, but on the other hand, this range of values can lead to poor performance. Occasionally the best values lead to a SCPO which is a poor approximation to ICP which leads to weaker performance than just using the traditional ICP approach.

4. Directly Optimized Inductive Conformal Regression

DOICR is presented as an alternative algorithm that can directly optimize the ICP by minimizing $MPIW$ whilst controlling for validity. The general approach of DOICR is to embed ICP within the loss function of the DNN and then use gradient descent, as usual, to explore the space of the loss function. Hence, when using the normalized NCM, conceptually this involves searching across the range of all ICPs given by parameters η and θ given in Eq. (12). This is possible because an ICP can be formed without reference to an underlying machine learning algorithm as it is traditionally deployed and as illustrated in Fig. 1. To see this, referring to Section 3.2, the training set can be made empty and the NCM is formed without any training. Taking the normalized NCM, used for regression, it is sufficient to specify any values for η and θ, and an ICP can be run. If the values are arbitrary, the ICP will be a poor predictor and this will be reflected in inefficient predictions (i.e. wide PIs). However, Eq. (11) will nevertheless guarantee validity (Vovk et al., 2005). In consequence, η and θ form a space of (infinite) ICPs, all of which are valid, from which gradient descent can search to minimize $MPIW$. These ICPs that are used in the search are referred to as *embedded* ICPs. This approach to direct optimization of ICP through minimizing predictive inefficiency has already been used by Stutz et al. (2022) for classification problems with their ConfTr method and has been shown to outperform baseline ICP as well as a version of SCPO for classification.

To ensure the validity of the embedded ICPs, it is necessary that the training data is deployed correctly. Using the notation of Section 3.2, there is no need for a "training" set, as discussed above, but the "calibration" and "test" sets need to be independent. Therefore, the training data provided for optimization is randomly split into a proper training set $D1$ and an independent embedded calibration set $D2$. The embedded ICPs make predictions on $D1$ to compute $MPIW$ used as the value of the loss, hence $D1$ takes the role of the "test" set in the ordinary specification of ICP given in Section 3.2. Hence, following Eq. (16), the loss function is

$$Loss_{ICP-embedded} = \frac{2q}{|D1|} \sum_{i \in D1} \sigma(\theta, \mathbf{x}_i) \tag{19}$$

where q is the $(1-\epsilon)$th quantile of the NCMs of $D2$ and $|D1|$ is the number of examples in $D1$. For this loss function to be used in gradient descent, it is necessary that it is continuous. Decomposing the loss, the term in the sum is dependent on σ being a continuous function. If it is the output from a NN, this will be the case. The $|D1|$ term is a constant. However, q is the empirical quantile of the NCMs computed across the calibration set $D2$, so $q = \alpha_i$ for some ith example in $D2$ which is rank ordered in ascending order at position $\lceil \epsilon |D2| \rceil$. There are two ways that q can change, relative to η and θ:

1. The example i does not change, but α_i changes with η and θ, according to Eq. (12). If m and σ are both continuous functions, then the NCM and hence q are continuous. Since m and σ are output from DNN then this is indeed the case.
2. Or, example i will change to some other example in the calibration set $D2$, say, some $j \neq i$, so that with a change of η and θ, $q = \alpha_j$. This can only happen if the two examples switch ranking (i.e. initially $\alpha_i < \alpha_j$, then with a change of η and θ, this changes to $\alpha_j < \alpha_i$, or vice versa). Since α_i is a continuous function, as established above, this implies that there is some point at which i and j have the same NCM value, which is the point that they switch rankings and q changes, hence q is continuous at this switch point, although not smooth.

Hence, since all terms in Eq. (19) are continuous, the loss function is continuous and may be used as part of gradient descent. The loss function is not smooth, but it is common for modern optimizers to handle continuous, non-smooth loss functions, using techniques such as subgradient methods (e.g. see Nesterov, 2009; Shor, 1985) and coordinate descent (Tseng & Yun, 2009). The computation of this loss value is given in Algorithm 1.

Algorithm 1 Compute $Loss_{ICP-embedded}$

```
Require: $D1$ (proper training data set), and $D2$ (embedded calibration set),
Require: $N(w)$ (neural network with weights vector $w$),
Require: $(1-\epsilon)$ confidence level.
Return the vector of output values $\mathbf{m}_1, \mathbf{s}_1$ using forward propagation through $N(w)$ with $D1$.
Return the vector of output values $\mathbf{m}_2, \mathbf{s}_2$ using forward propagation through $N(w)$ with $D2$.
$\alpha = \frac{|\mathbf{y}_2 - \mathbf{m}_2|}{\exp(\mathbf{s}_2)}$ where $\mathbf{y}_2$ is the vector of the labels in $D2$.   ▷ $\alpha$ is the vector of nonconformity measures for $D2$.
$q = (1-\epsilon)$th quantile of the vector $\alpha$.
Return $Loss_{ICP-embedded} = 2q \sum \exp(\mathbf{s}_1)/|D1|$.
```

For this study, a single DNN is used to output the values of both m and σ functions. This follows the style of QD-soft, but differs from use in traditional ICP where separate models are required for m and σ, run sequentially, since σ is intended to model the uncertainty in m. Arguably, allowing a single DNN for both functions leads to less computation cost and also more flexibility in formulating the model structure for the normalized NCM. Therefore, since m and σ use the same DNN structure, they share the same parameters, $\eta = \theta$ which are the weights in the NN. We will refer to these shared parameters as θ. Fig. 2 illustrates the DNN used with DOICR.

As with other machine learning algorithms, we may expect DOICR to overfit, and this would be evident in the downwardly biased $MPIW$. Therefore, as usual, after training, we use independent hold-out calibration and test sets to run a test ICP using parameters $\hat{\theta}$ estimated by DOICR.

Although the embedded ICPs are valid, there is random variation between them which will mean that some empirically deviate from validity by random chance, based on training sample size. With a large search space of ICPs, the optimizer can exploit this feature to home in on the ICPs to decrease $MPIW$ by lowering $PICP$ below CL. As an example, with the Bias data set, which we describe later in Section 5, and setting $CL = 0.9$, we find that DOICR will deliver an embedded ICP with very low $PICP = 0.595$ to achieve a low $MPIW = 0.717$. However, once the same parameter settings are used in the test ICP, validity is restored ($PICP = 0.899$), as expected, but at the expense of higher $MPIW = 1.814$ and this is worse performance than the traditional ICP (test $PICP = 0.883$ and $MPIW = 1.120$). The change in training $PICP$ and $MPIW$ through the epochs of the gradient descent process is shown in Fig. 3 which demonstrates that $PICP$ is pushed below the CL whilst $MPIW$ continues to improve. The graph suggests that with further epochs, $PICP$ could be pushed even lower. This problem can be viewed as an overfitting problem with the selection of a training set $D1$ and embedded calibration set $D2$ across the training process. To remedy this problem, the training set is shuffled each epoch, and a new $D1$ and $D2$ are selected each time. This prevents the optimizer from following a path seeking the ICP with the lowest $PICP$. This is also the approach used by Stutz et al. (2022) in their experiments with classification. Using this approach, for Bias with $CL = 0.9$, results on the training data are $PICP = 0.8931$ and $MPIW = 0.7931$, leading to test results of $PICP = 0.8951$ and $MPIW = 1.022$ which is an improvement on traditional ICP (test $PICP = 0.883$ and $MPIW = 1.120$).

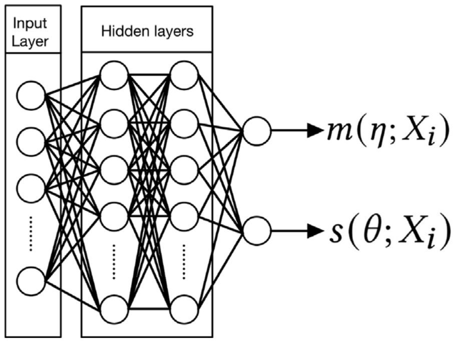


Fig. 2. Neural network structure used with DOICR.

DOICR is implemented using PyTorch for which the underlying neural network structure needs to be defined and only the loss function needs to be passed. Gradients do not need to be derived analytically, since PyTorch uses automatic differentiation to evaluate the loss function and implement back-propagation (Baydin et al., 2017). Even though the loss function involves an iterative step to compute q, the autograd package in Python is able to handle the differentiation of code blocks (Paszke et al., 2017). The DOICR algorithm is given in Algorithm 2 and the full framework for DOICR is illustrated in Fig. 4.

Algorithm 2 Directly Optimized Inductive Regression (DOICR)

```
Require: $T$ (training data set) and $r$ (percentage of data to be used as the embedded calibration set),
Require: $N$ (neural network with initial weights $w_0$) and $t_{epochs}$ (total number of epochs).
$w \leftarrow w_0$.
for each $i = 1, 2, \ldots t_{epochs}$ do
    Divide $T$ randomly into $D1$ and $D2$ with percentage $(100-r)\%$ and $r\%$ respectively.
    Perform back propagation (autograd) with $Loss_{ICP-embedded}$ and update weights $w$.
end for
return weights $w$.
```

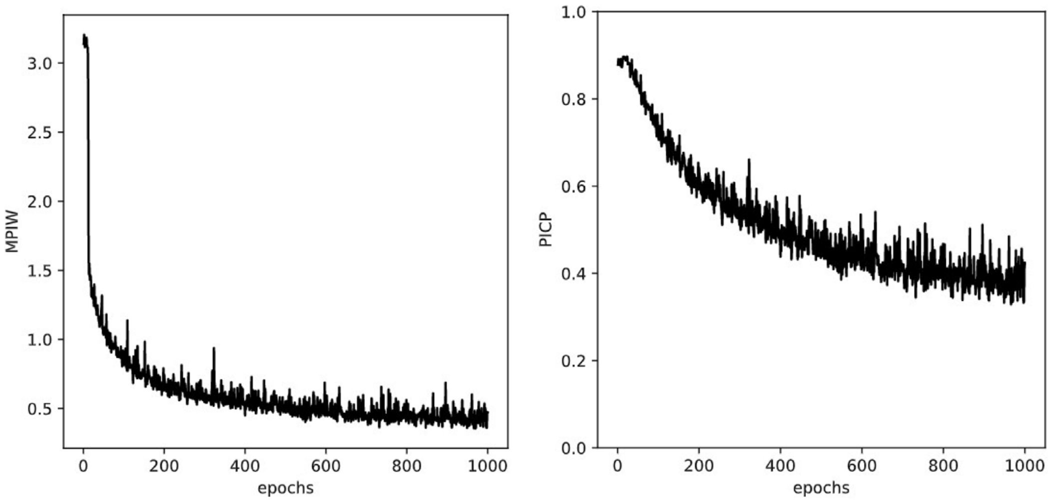


Fig. 3. Training $MPIW$ and $PICP$ obtained by embedded ICP with fixed embedded training and calibration sets (i.e. without random shuffle each epoch); for Bias data set and $CL = 0.9$.

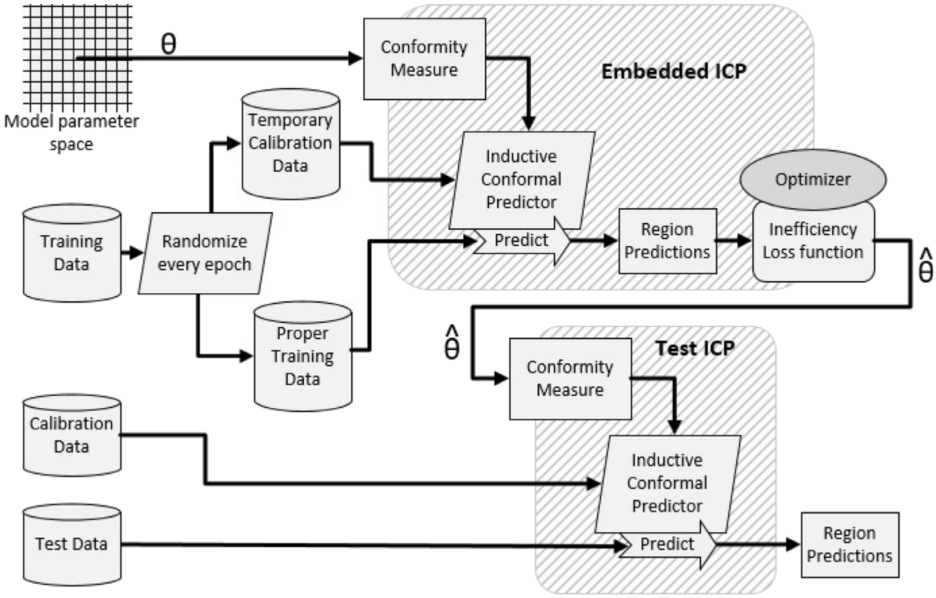


Fig. 4. General DOICR framework.

5. Experimental design and results

To demonstrate that our method can generate high-quality PIs, it is compared with the three mentioned baseline methods (i.e. QD-soft, ICP, and SCPO) on six publicly available datasets, including five tabular data sets and one image data sets. The public datasets we selected are tailored for regression problems across diverse application areas. The descriptions of these data sets are given in Table 1. To achieve good performance using NN, we only selected data sets with over 2000 examples. In particular the Ames and KC data sets are for house price prediction in which it is typical to require prediction intervals (see e.g. www.zoopla.co.uk), and RSNA is chosen as an example of an image data set.

For the RSNA data set, X-ray images are given in different sizes. For this study, they are all converted to RGB images with 224 by 224 pixels. The KC dataset is available on Kaggle website[1]

5.1. Experimental settings

In this study, for all experiments, a broad range of plausible confidence levels (CL) were explored: 0.8, 0.9, 0.95, and 0.99.

For the tabular data sets, a standard multi-layer perceptron (MLP) neural network structure is used, whereas a convolutional neural network (CNN) structure is used for the RSNA image data. Details of implementation are given in the next subsections.

5.1.1. Settings for hyperparameters inside loss functions

Of the four algorithms used, only the SCPO and QD-soft have hyperparameters inside their loss functions. However, grid search was not performed directly for these hyperparameters, since the λ and γ themselves affect the value of the loss (e.g. the $Loss_{QD-soft}$ for $\lambda = 100$ is definitely larger than that for $\lambda = 1$, it is impossible to use the value of $Loss_{QD-soft}$ as the criterion for selecting the best model). For QD-soft, multiple variations of λ and γ were explored and it was found that they have a great influence on the experimental results, and most of the combinations of values yielded results that are not in accordance with the High-Quality principles. Most of the reasonable combinations

[1] https://www.kaggle.com/harlfoxem/housesalesprediction (as of 23 May 2023).

Table 1
Description of all datasets. **n** = number of examples and **v** = number of predictor variables; for image datasets, the input data is the image, so **v** is shown as NA.

Name	Description	Target variable	n	v
KC	House sales in King County, USA (available on Kaggle website)	Sale price	21 613	24
Bias	Bias correction on temperature prediction (Cho et al., 2020)	Minimum temperature	7752	24
Ames	Housing data in Iowa, USA (Cock and Dean, 2011)	Sale price	2928	9
Super	Superconductor data (Hamidieh, 2018)	Critical temperature	21 263	81
GPU	GPU performance data (Nugteren and Codreanu, 2015)	Average performance time	241 600	14
RSNA	X-ray images and corresponding bone ages (Halabi et al., 2018)	Bone age	16 211	NA

Table 2
Hyperparameters for MLP and their corresponding search ranges.

Hyperparameter	Search range
Learning rate	0.0001, 0.001, 0.01, 0.1
Weight decay	0, 0.0001, 0.001
Batch size	16, 32, 64, 128

are close to $\lambda = 0.01$ and $\gamma = 160$ which are the values used by Pearce et al. (2021) in their implementation. For SCPO, we chose the same λ and γ as recommended by Bellotti (2020) .

5.1.2. Settings for MLP

For all four methods, we fixed the MLP structure as two hidden layers with 20 neurons in each layer, since initial exploration suggested this provides good performance across the multiple benchmark algorithms: base ICP, SCPO and QD-Soft. Models were trained with 1000 epochs. Across all these cases and DOICR, the MLP converges within the 1000 epochs, in the sense that generally loss is not lower over the last 100 epochs. For experiments on tabular data sets, grid search was used for all methods to find the best MLP hyperparameters. These are listed in Table 2 with candidate values for the search. This gives a total of 48 models to run during the grid search. The combination of hyperparameters that achieve the lowest corresponding loss on an independent validation set is used as the final model to be tested using an independent test set, and also an independent calibration set for ICP, SCPO, and DOICR. We make more data available for training and split the tabular data sets into partitions as shown in Table 3. Notice that since QD-Soft does not need a calibration set, that part is allocated to training. These proportions give reasonable sizes for each partition, except for Ames for which the low sample size makes conducting the experiment more challenging.

5.1.3. Settings for CNN

Several pre-existing backbone CNNs were used in experiments for learning the image data, which are EfficientNet (Tan & Le, 2019), ResNet (He et al., 2015), and Inception V4 (Szegedy et al., 2017) in PyTorch. In the QD-soft and DOICR experiments, the final layers of the backbone CNNs are configured with 2 output neurons. In contrast, during the ICP and SCPO experiments, each backbone CNN has a single output unit, and two backbone CNNs, one for point prediction and another for residual prediction, are trained. Hyperparameter searching was not performed, since it takes considerable time to train CNNs using image data, and it is impractical to perform the grid search for these experiments. As a result, we used the default values provided in Pytorch for the dropout rate and learning rate. As for the weight decay for QD-soft, using the default weight decay value in Pytorch will typically result in errors during training, due to infinities in the computed gradients. According to Pearce et al. (2021) , QD-soft will be vulnerable if a large decay rate is used in the training process. As a consequence, a tenth of the default value for the decay rate (0.001) is used in this study. Since grid search is not required, no validation set is required and data divisions are shown in Table 3. Since QD-Soft does not require a calibration set, that portion of data is added to the training.

Due to hardware limitations, the GPU we used could not process too many images at one time, and the GPU memory capacity is exceeded when the batch size is larger than 150. Meanwhile, after investigation, we found that a batch size lower than 100 would lead to errors (due to infinite gradient values) during training CNN with $Loss_{QD-soft}$, so we chose 128 as the batch size for all four approaches. The optimizer used is AdamW (Loshchilov & Hutter, 2019) since it has a faster convergence speed than Stochastic gradient descent for CNNs, able to converge within 50 epochs, which can greatly improve experimental efficiency. Unlike Adam (Kingma & Ba, 2014), AdamW directly adds the gradient of the regularization term to the backpropagation formula, eliminating the need to manually add the regularization term to the loss. Therefore, in our experiments, it is more computationally efficient than Adam.

5.2. Experimental results

The four methods were run for the six data sets and results on the independent test set are presented in Figs. 5 and 6 for MLP and CNN respectively. These figures plot CL and PICP against MPIW (inefficiency). Deviations of PICP from CL are shown by vertical lines. Tables 4 to 5 show the same full results in table form. The target we are focusing on is MPIW, and the best-performing methods will be marked in red in the tables. However, results where $PICP$ and confidence level differ significantly also need to be monitored. When $(1-\varepsilon) - PICP > 0.02$, the box indicating $PICP$ will be filled with yellow in the tables.

Overall, for almost all experiments DOICR achieved the smallest $MPIW$, and hence the lowest predictive inefficiency, whilst the corresponding $PICP$ is not far from the CL, demonstrating validity. Often the improvement in performance with DOICR is large. Two occasions when DOICR does not achieve the smallest $MPIW$ are for the Bias data set with CL = 0.8 and Ames with CL = 0.95, when QD-soft is marginally better, but at the expense of lower $PICP$ as deviation from validity. Another occasion is for ResNet with CL = 0.8 when a standard ICP is competitive, but the difference in performance is small. For all other CLs, DOICR is clearly performing better.

Generally, ICP and SCPO perform relatively poorly in this study, especially at higher CL. QD-soft is more competitive, but does not give a guarantee of validity, and often is seen to deviate greatly from validity (i.e. $PICP$ is far from CL). The results and process of our experiment illustrate the superiority of the DOICR in comparison to other algorithms:

- When contrasted with QD-soft and SCPO, DOICR demonstrates stable validity, avoiding the pitfalls of very low PICP leading to

Table 3
The division of the data sets under the 4 different methods.

Data set	Method	Training set	Validation set	Calibration set	Test set
Tabular	ICP, SCPO and DOICR	40%	20%	20%	20%
	QD-soft	60%	20%	–	20%
RSNA (image)	ICP, SCPO and DOICR	50%	–	25%	25%
	QD-soft	75%	–	–	25%

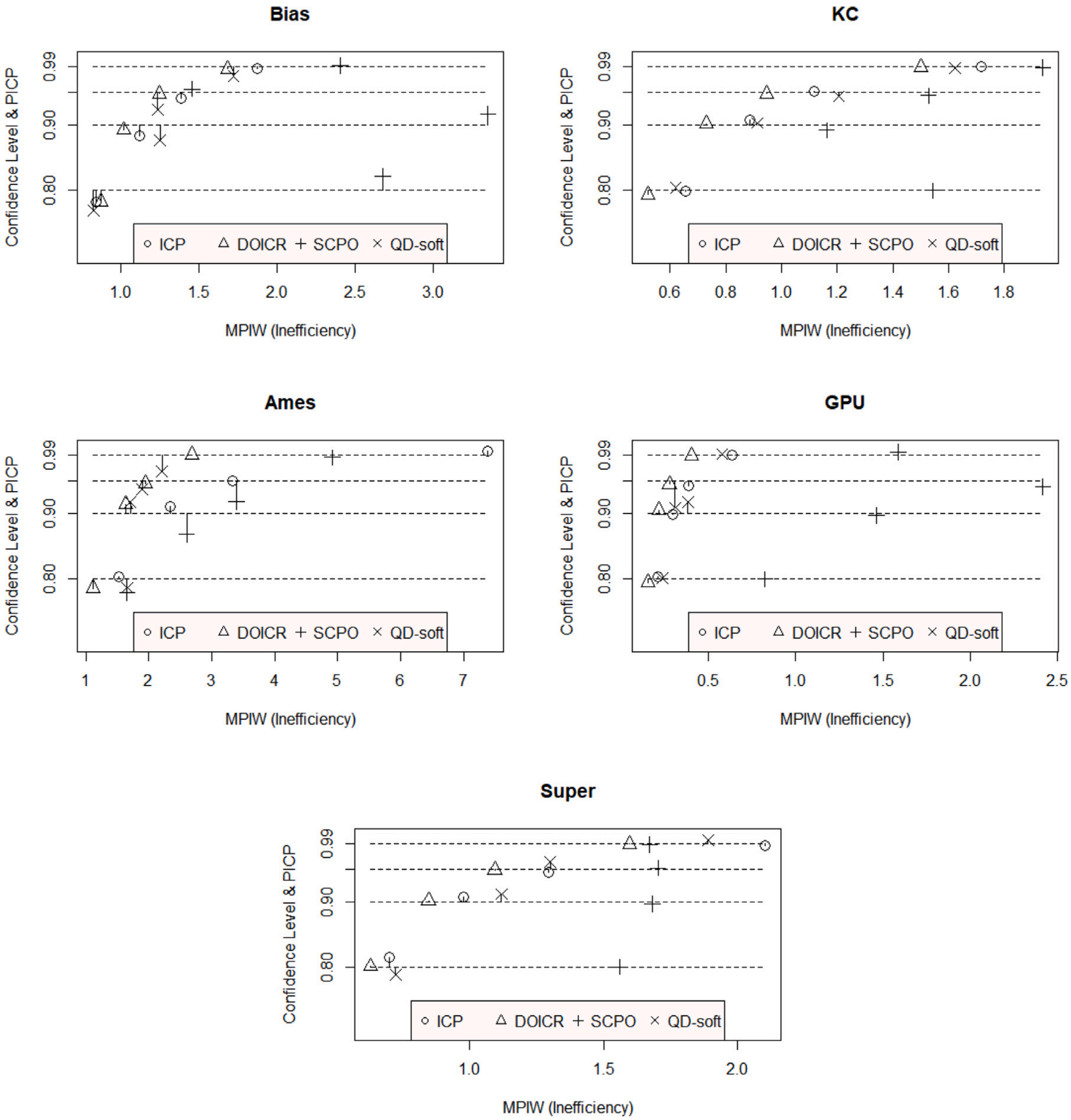


Fig. 5. Results on test sets for MLP on the 5 tabular data sets. The target confidence level (CL) is shown as horizontal dashed lines. Each experiment is shown as a point: PICP is shown as vertical deviation from the target CL and MIPW (predictive inefficiency) is shown on the horizontal axis.

minimal MPIW, or exceedingly high PICP resulting in substantial MPIW. This issue is often encountered with PI generators that employ ($MPIW + \lambda PICP$) framework as their loss function structure. It suggests that DOICR flawlessly inherits ICP's validity assurance.

- In relation to ICP, DOICR provides a more efficient training period. Traditional ICP training demands the sequential training of two models, whereas DOICR necessitates just one model with an equivalent number of parameters, effectively halving the training duration.
- The stability of DOICR's training proves to be superior, displaying low sensitivity to hyperparameters such as the weight decay rate. Consequently, it is less susceptible to computational issues.
- With DOICR, there is no need for additional hyperparameter searches within the loss function, significantly reducing workload. In contrast, algorithms like QD-soft and SCPO, which utilize the dual High-Quality principles as their optimization objective, may require varied optimal hyperparameters for each CL, dataset, and model, thus necessitating a considerable amount of time for λ and γ exploration.

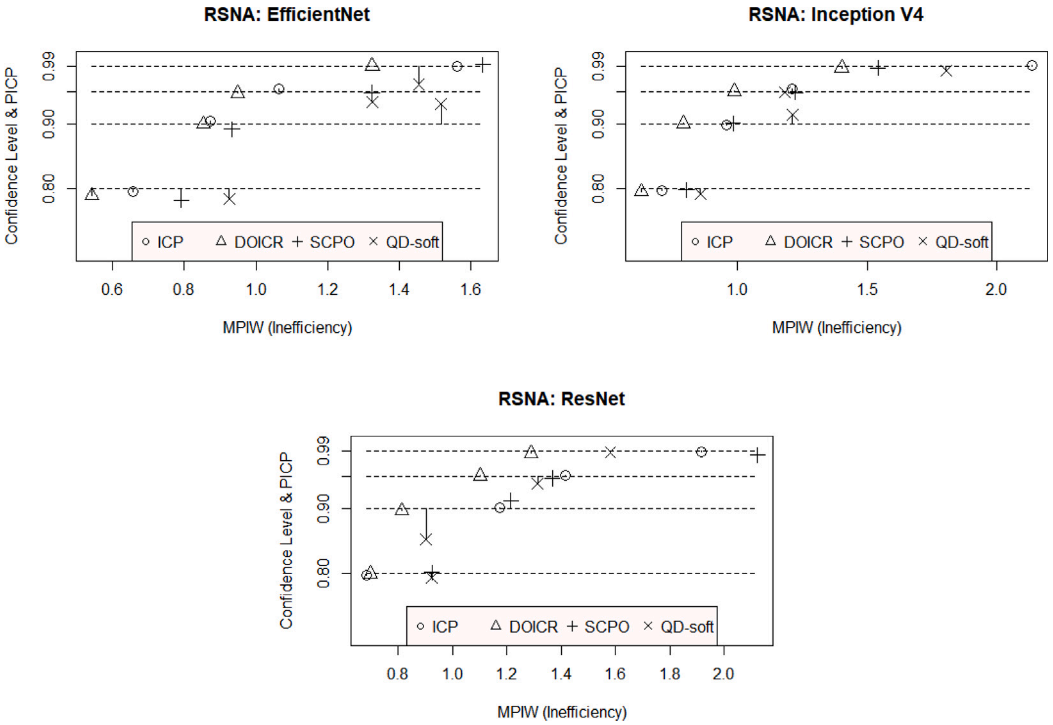


Fig. 6. Results on test sets for CNN on data set RSNA for the 3 backbones. The target confidence level (CL) is shown as horizontal dashed lines. Each experiment is shown as a point: PICP is shown as vertical deviation from the target CL and MIPW (predictive inefficiency) is shown on the horizontal axis.

6. Conclusion and future work

For many applications in the regression setting, there is a need to produce PIs, rather than point predictions. Additionally, the more that machine learning is being used in real-world critical settings, there is an increasing interest in reliable machine learning. For PIs, this essentially requires predictive validity. That is, if the user is expecting a particular confidence level, then the probability that the true label is in the PI meets the confidence level.

In this paper, we take advantage of the validity of ICP and use an algorithm, DOICR, that only needs to minimize $MPIW$ by applying gradient descent to explore the space of possible ICPs. We compared DOICR with other previous algorithms such as traditional ICP, built on an underlying machine learning algorithm and QD-soft, on six public data sets. It is found that DOICR not only inherits the validity of ICP, but also has an excellent performance in reducing the width of PIs ($MPIW$). Not only that, DOICR can also be easily combined with various state-of-the-art Deep Learning approaches such as CNN, and achieve far better performance than the baselines. The approaches we focus on offer the advantage of seamless integration with minimal modifications to current state-of-the-art DNNs, with negligible impact on the training time compared to a conventional point estimation model. Nevertheless, existing methods are not without challenges: they do not guarantee sufficient validity and demand a considerable amount of time for additional hyperparameter searches within the loss function. To address these two issues, we propose the DOICR. This technique not only resolves the aforementioned challenges but also excels in reducing the $MPIW$ while ensuring validity and enhancing usability.

Several aspects could lead to further investigation.

- Firstly, as with ICP, DOICR suffers from the dilemma of wasting data, because they both need to use part of the database to build calibration sets, which will greatly reduce the number of training examples. Further research is needed to determine the best use of data for DOICR, especially in high-dimensional deep learning settings.
- Secondly, most of the PIs are evaluated in terms of marginal validity; i.e. $PICP$ is measuring aggregate coverage across the population. However, there is a growing concern that validity may not be evenly spread amongst different sub-populations; that is, it may not be conditionally valid, in some sense (Vovk, 2013). Indeed, by setting the goal to maximize predictive efficiency, it may encourage the optimizer to assign poor (conditional) validity across some sub-populations to meet the overall $MIPW$ target. This could happen if a subpopulation is inherently harder to predict or there is a lack of data or poor data quality within these subgroups (so the optimizer is able to sacrifice validity in these subpopulations to meet better $MIPW$ overall). The problem of subpopulation bias in artificial intelligence systems is a general problem that needs addressing (see e.g. the study by Buolamwini and Gebru (2018)), but bias in conditional validity is a specific problem for PI generators and ICP that needs special consideration. For ICP classification, the development of the loss function to allow for some conditional validity such as class conditioning, has been proposed (Stutz et al., 2022). Further investigation and remedies should be considered to deal with this possibility, in the context of DOICR with general frameworks to handle conditional validity, such as guided adjustments to the NCM (Bellotti, 2021a).
- Thirdly, since DOICR is successful when using a CNN model structure, it would also be interesting to apply it to other complex state-of-the-art algorithms such as Temporal Convolutional Networks (Lea et al., 2017) or Transformer (Vaswani et al., 2017) or other complex DNNs.
- Fourthly, further work applying the method to specific problem domains should be further explored. For instance, multiagent systems could also benefit from the implementation of this method. We could consider dynamic control for multiagent systems with time-varying parameters (Cao et al., 2022; Lin et al., 2022) to provide reliable tracking and estimation of system states via NN with effective bounds using PIs, ICP and DOICR optimization. Or, it might provide significant improvements in the control and management of microgrids based on existing work with nonzero-sum game-based voltage recovery consensus optimal control for

Table 4
Results showing PICP and MPIW on test sets for MLP.

Data set	Confidence level	Method	PICP	MPIW
Bias	0.80	ICP	0.781	0.842
		DOICR	0.782	0.873
		SCPO	0.821	2.680
		QD-soft	0.768	0.824
	0.90	ICP	0.883	1.120
		DOICR	0.892	1.019
		SCPO	0.917	3.353
		QD-soft	0.876	1.252
	0.95	ICP	0.941	1.389
		DOICR	0.948	1.247
		SCPO	0.955	1.456
		QD-soft	0.924	1.236
	0.99	ICP	0.986	1.874
		DOICR	0.986	1.685
		SCPO	0.991	2.407
		QD-soft	0.975	1.723
KC	0.80	ICP	0.798	0.656
		DOICR	0.793	0.521
		SCPO	0.799	1.545
		QD-soft	0.803	0.620
	0.90	ICP	0.908	0.887
		DOICR	0.902	0.730
		SCPO	0.892	1.162
		QD-soft	0.903	0.912
	0.95	ICP	0.952	1.116
		DOICR	0.948	0.946
		SCPO	0.945	1.529
		QD-soft	0.944	1.206
	0.99	ICP	0.989	1.717
		DOICR	0.989	1.501
		SCPO	0.988	1.939
		QD-soft	0.987	1.623
Ames	0.80	ICP	0.803	1.524
		DOICR	0.785	1.116
		SCPO	0.778	1.653
		QD-soft	0.786	1.159
	0.90	ICP	0.911	2.346
		DOICR	0.914	1.630
		SCPO	0.868	2.598
		QD-soft	0.918	1.703
	0.95	ICP	0.950	3.326
		DOICR	0.946	1.947
		SCPO	0.918	3.392
		QD-soft	0.937	1.895
	0.99	ICP	0.995	7.385
		DOICR	0.990	2.682
		SCPO	0.986	4.916
		QD-soft	0.964	2.211
GPU	0.80	ICP	0.803	0.211
		DOICR	0.794	0.158
		SCPO	0.799	0.825
		QD-soft	0.800	0.240
	0.90	ICP	0.899	0.296
		DOICR	0.905	0.220
		SCPO	0.897	1.463
		QD-soft	0.917	0.384
	0.95	ICP	0.943	0.387
		DOICR	0.945	0.279
		SCPO	0.941	2.414
		QD-soft	0.909	0.311
	0.99	ICP	0.990	0.640
		DOICR	0.989	0.407
		SCPO	0.994	1.588
		QD-soft	0.991	0.581
Super	0.80	ICP	0.815	0.704
		DOICR	0.801	0.633
		SCPO	0.800	1.560
		QD-soft	0.788	0.726
	0.90	ICP	0.908	0.978
		DOICR	0.902	0.848
		SCPO	0.897	1.682
		QD-soft	0.912	1.120
	0.95	ICP	0.945	1.294
		DOICR	0.949	1.095
		SCPO	0.952	1.703
		QD-soft	0.961	1.301
	0.99	ICP	0.986	2.103
		DOICR	0.989	1.598
		SCPO	0.988	1.673
		QD-soft	0.995	1.890

(continued on next page)

Table 4 (continued).

Table 5
Results showing PICP and MPIW on test sets for CNN.

Backbone NN	Confidence level	Method	PICP	MPIW
EfficientNet	0.80	ICP	0.795	0.656
		DOICR	0.789	0.543
		SCPO	0.782	0.792
		QD-soft	0.784	0.925
	0.90	ICP	0.905	0.872
		DOICR	0.900	0.853
		SCPO	0.892	0.933
		QD-soft	0.931	1.516
	0.95	ICP	0.955	1.064
		DOICR	0.948	0.949
		SCPO	0.949	1.323
		QD-soft	0.934	1.324
	0.99	ICP	0.989	1.560
		DOICR	0.990	1.324
		SCPO	0.992	1.632
		QD-soft	0.961	1.454
Inception V4	0.80	ICP	0.796	0.707
		DOICR	0.795	0.629
		SCPO	0.798	0.802
		QD-soft	0.791	0.856
	0.90	ICP	0.898	0.956
		DOICR	0.901	0.791
		SCPO	0.901	0.983
		QD-soft	0.914	1.211
	0.95	ICP	0.954	1.212
		DOICR	0.951	0.987
		SCPO	0.949	1.223
		QD-soft	0.949	1.182
	0.99	ICP	0.991	2.137
		DOICR	0.987	1.402
		SCPO	0.987	1.542
		QD-soft	0.983	1.804
ResNet	0.80	**ICP**	0.797	0.684
		DOICR	0.799	0.697
		SCPO	0.801	0.924
		QD-soft	0.793	0.922
	0.90	ICP	0.901	1.175
		DOICR	0.897	0.813
		SCPO	0.912	1.215
		QD-soft	0.852	0.902
	0.95	ICP	0.952	1.417
		DOICR	0.951	1.102
		SCPO	0.947	1.367
		QD-soft	0.939	1.314
	0.99	ICP	0.988	1.917
		DOICR	0.986	1.289
		SCPO	0.983	2.123
		QD-soft	0.987	1.581

nonlinear microgrids (Liu et al., 2022), enabling reliable estimation of electricity generation by microgrids systems, within PI bounds. Or, for emotion detection problems where two outcomes, valance and arousal, are predicted (Tellamekala et al., 2022), the use of DOICR could be developed further to provide valid PIs in 2-dimensions. These potential application examples underscore the versatility and broad applicability of the proposed method, and certainly warrants further research in these areas.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

All data is from publicly accessible sources, but it is not ours.

Appendix. Glossary of Abbreviations

CL	Confidence Level, denoted as $(1-\varepsilon)$
CNN	Convolutional Neural Network
DNN	Deep Neural Network
DOICR	Directly Optimized Inductive Conformal Regression
ICP	Inductive Conformal Predictor
NCM	Nonconformity Measure
NN	Neural Network
MLP	Multilayer Perceptron
MPIW	Mean Prediction Interval Width
PI	Prediction Interval
PICP	Prediction Interval Coverage Probability
QD-soft	Quality-Driven-soft
SCPO	Surrogate Conformal Predictor Optimization

References

Bagnall, A., & Davis, L. (2014). Predictive modelling of bone age through classification and regression of bone shapes. arXiv preprint arXiv:1406.4781.

Baydin, A. G., Pearlmutter, B. A., Radul, A., & Siskind, J. M. (2017). Automatic differentiation in machine learning: a survey. *Journal of Machine Learning Research*, *18*, 153:1–153:43.

Bellotti, A. (2020). Constructing normalized nonconformity measures based on maximizing predictive efficiency. In *Proceedings of machine learning research, conformal and probabilistic prediction and applications, Vol. 128* (pp. 1–20).

Bellotti, A. (2021a). Approximation to object conditional validity with inductive conformal predictors. In *Proceedings of machine learning research, conformal and probabilistic prediction and applications, Vol. 152* (pp. 1–20).

Bellotti, A. (2021b). Optimized conformal classification using gradient descent approximation. arxiv.org/abs/2105.11255.

Buolamwini, J., & Gebru, T. (2018). Gender shades: Intersectional accuracy disparities in commercial gender classification. In S. A. Friedler, & C. Wilson (Eds.), *Proceedings of machine learning research*: *Vol. 81, Proceedings of the 1st conference on fairness, accountability and transparency* (pp. 77–91). PMLR, URL: https://proceedings.mlr.press/v81/buolamwini18a.html.

Cao, L., Pan, Y., Liang, H., & Huang, T. (2022). Observer-based dynamic event-triggered control for multiagent systems with time-varying delay. *IEEE Transactions on Cybernetics*.

Cho, D., Yoo, C., Im, J., & Cha, D. (2020). *Comparative assessment of various machine learning-based bias correction methods for numerical weather prediction model forecasts of extreme air temperatures in urban areas*. John Wiley & Sons, Ltd.

Cock, D., & Dean (2011). Ames, Iowa: Alternative to the Boston housing data as an end of semester regression project. *Journal of Statistics Education*, *19*(3), 8.

Deruyttere, T., Milewski, V., & Moens, M.-F. (2021). Giving commands to a self-driving car: How to deal with uncertain situations? *Engineering Applications of Artificial Intelligence*, *103*, 104–257.

Geifman, Y., & El-Yaniv, R. (2017). Selective classification for deep neural networks. *Advances in Neural Information Processing Systems*, *30*.

Halabi, S. S., Prevedello, L. M., Kalpathy-Cramer, J., Mamonov, A. B., Bilbily, A., Cicero, M., Pan, I., Pereira, L. A., Sousa, R. T., & Abdala, N. (2018). The RSNA pediatric bone age machine learning challenge. *Radiology*.

Hamidieh, K. (2018). A data-driven statistical model for predicting the critical temperature of a superconductor. *Computational Materials Science*, *154*, 346–354.

Hansen, K. B., & Borch, C. (2021). The absorption and multiplication of uncertainty in machine-learning-driven finance. *The British Journal of Sociology*, *72*(4), 1015–1029.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep residual learning for image recognition. CoRR, abs/1512.03385 arXiv:1512.03385 URL: http://arxiv.org/abs/1512.03385.

Heskes, T. (1996). Practical confidence and prediction intervals. In *Proceedings of the 9th international conference on neural information processing systems* (pp. 176–182). Cambridge, MA, USA: MIT Press.

Johansson, U., Boström, H., Löfström, T., & Linusson, H. (2014). Regression conformal prediction with random forests. *Machine Learning*, *97*, 155–176.

Kingma, D., & Ba, J. (2014). Adam: A method for stochastic optimization. *Computer Science*.

Krzywinski, Martin, Altman, & Naomi (2013). Points of significance: Power and sample size. *Nature Methods*.

Lai, Y., Shi, Y., Han, Y., Shao, Y., Qi, M., & Li, B. (2022). Exploring uncertainty in regression neural networks for construction of prediction intervals. *Neurocomputing*, *481*, 249–257.

Lea, C., Flynn, M. D., Vidal, R., Reiter, A., & Hager, G. D. (2017). Temporal convolutional networks for action segmentation and detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 156–165).

Li, M., & Wang, D. (2021). 2-D stochastic configuration networks for image data analytics. *IEEE Transactions on Cybernetics*, *51*(1), 359–372.

Lin, G., Li, H., Ma, H., & Zhou, Q. (2022). Distributed containment control for human-in-the-loop MASs with unknown time-varying parameters. *IEEE Transactions on Circuits and Systems. I. Regular Papers*, *69*(12), 5300–5311.

Liu, G., Sun, Q., Wang, R., & Hu, X. (2022). Nonzero-sum game-based voltage recovery consensus optimal control for nonlinear microgrids system. *IEEE Transactions on Neural Networks and Learning Systems*.

Loshchilov, I., & Hutter, F. (2019). Decoupled weight decay regularization. In *International conference on learning representations*.

Lu, J., & Ding, J. (2019). Mixed-distribution-based robust stochastic configuration networks for prediction interval construction. *IEEE Transactions on Industrial Informatics*, *16*(8), 5099–5109.

Lu, J., Ding, J., Dai, X., & Chai, T. (2020). Ensemble stochastic configuration networks for estimating prediction intervals: A simultaneous robust training algorithm and its application. *IEEE Transactions on Neural Networks and Learning Systems*, *31*(12), 5426–5440. http://dx.doi.org/10.1109/TNNLS.2020.2967816.

MacKay, D. J. C. (1992). A practical Bayesian framework for backpropagation networks. *Neural Computation*, *4*(3), 448–472. http://dx.doi.org/10.1162/neco.1992.4.3.448.

Nesterov, Y. (2009). Primal-dual subgradient methods for convex problems. *Mathematical Programming*, *120*, 221—259.

Nugteren, C., & Codreanu, V. (2015). CLTune: A generic auto-tuner for OpenCL kernels. In *Embedded multicore/many-core systems-on-chip (MCSoC), 2015 IEEE 9th international symposium on* (pp. 195–202).

Papadopoulos, H., & Haralambous, H. (2011). Reliable prediction intervals with regression neural networks. *Neural Networks*, *24*(8), 842–851.

Papadopoulos, H., Proedrou, K., Vovk, V., & Gammerman, A. (2002). *Lecture notes in computer science*, *Inductive Confidence Machines for Regression*. Springer.

Papadopoulos, H., Vovk, V., & Gammerman, A. (2011). Regression conformal prediction with nearest neighbours. *Journal of Artificial Intelligence Research*, *40*(4), 815–840.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Edward Yang, Z. D., Lin, Z., Desmaison, A., Antiga, L., & Lerer, A. (2017). Automatic differentiation in PyTorch. In *NIPS 2017 autodiff workshop: the future of gradient-based machine learning software and techniques, Long Beach, CA, USA, December 9, 2017*.

Pearce, T., Brintrup, A., Zaki, M., & Neely, A. (2018). High-quality prediction intervals for deep learning: A distribution-free, ensembled approach. In J. Dy, & A. Krause (Eds.), *Proceedings of machine learning research*: *Vol. 80, Proceedings of the 35th international conference on machine learning* (pp. 4075–4084).

Pearce, T., Brintrup, A., & Zhu, J. (2021). Understanding softmax confidence and uncertainty. arXiv preprint arXiv:2106.04972.

Shafer, G., & Vovk, V. (2008). A tutorial on conformal prediction. *Journal of Machine Learning Research*, *9*, 371–421.

Shor, N. (1985). *Springer series in computational mathematics*, *Minimization methods for non-differentiable functions*.

Stutz, D., Krishnamurthy, Dvijotham, Cemgil, A. T., & Doucet, A. (2022). Learning optimal conformal classifiers. In *International conference on learning representations*.

Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. A. (2017). Inception-v4, Inception-ResNet and the impact of residual connections on learning. In S. Singh, & S. Markovitch (Eds.), *Proceedings of the thirty-first AAAI conference on artificial intelligence, February 4-9, 2017, San Francisco, California, USA* (pp. 4278–4284). AAAI Press, URL: http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14806.

Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. CoRR, abs/1905.11946 arXiv:1905.11946 URL: http://arxiv.org/abs/1905.11946.

Tellamekala, M. K., Giesbrecht, T., & Valstar, M. (2022). Modelling stochastic context of audio-visual expressive behaviour with affective processes. *IEEE Transactions on Affective Computing*, 1. http://dx.doi.org/10.1109/TAFFC.2022.3157141.

Toccaceli, P. (2022). Introduction to conformal predictors. *Pattern Recognition*, *124*, Article 108507. http://dx.doi.org/10.1016/j.patcog.2021.108507, URL: https://www.sciencedirect.com/science/article/pii/S003132032100683X.

Tseng, P., & Yun, S. (2009). A coordinate gradient descent method for nonsmooth separable minimization. *Mathematical Programming*, *117*, 387—423.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. arXiv.org/abs/1706.03762.

Vovk, V. (2013). Conditional validity of inductive conformal predictors. *Machine Learning*, *92*, 349–376.

Vovk, V., Fedorova, V., Nouretdinov, I., & Gammerman, A. (2016). Criteria of efficiency for conformal prediction. In *COPA 2016: Proceedings of the 5th international symposium on conformal and probabilistic prediction with applications, Vol. 9653* (pp. 23–29).

Vovk, V., Gammerman, A., & Shafer, G. (2005). *Algorithmic learning in a random world*. Springer.

Wang, D., & Li, M. (2017). Stochastic configuration networks: Fundamentals and algorithms. *IEEE Transactions on Cybernetics*, *47*(10), 3466–3479.

Zhou, S. K., Greenspan, H., Davatzikos, C., Duncan, J. S., Van Ginneken, B., Madabhushi, A., Prince, J. L., Rueckert, D., & Summers, R. M. (2021). A review of deep learning in medical imaging: Imaging traits, technology trends, case studies with progress highlights, and future promises. *Proceedings of the IEEE*, *109*(5), 820–838.