Improving Uncertainty Quantification in Regression Problems through Conformal Training

Johannes Vallikivi

MPhil in Machine Learning and Machine Intelligence
Department of Engineering
University of Cambridge

This dissertation is submitted for the degree of
Master of Philosophy

Christ's College August 2023

Declaration

I, Johannes Vallikivi of Christ's College, being a candidate for the MPhil in Machine Learning and Machine Intelligence, hereby declare that this report and the work described in it are my own work, unaided except as may be specified below, and that the report does not contain material that has already been used to any substantial extent for a comparable purpose.

This thesis contains approximately 14335 words, excluding declarations, bibliography, photographs and diagrams, but including tables, footnotes, figure captions and appendices.

Johannes Vallikivi
August 2023

Acknowledgements

I would like to begin by expressing my gratitude to my supervisor, Mihaela van der Schaar. Thank you for your guidance and the insightful conversations we've had during this project. Special thanks are also due to my co-supervisors, Alan Jeffares and Nabeel Seedat. Your consistent feedback throughout this project has helped me stay on course and ask the right questions. I truly appreciate your support.

I extend my thanks to the Education and Youth Board of the Ministry of Education and Research in Estonia for generously funding and enabling my studies at the University of Cambridge through the Kristjan Jaak National Scholarship Programme. Your support has been invaluable to my academic pursuits.

Lastly, I must express my heartfelt appreciation to my partner, Johanna, for the unwavering support throughout this year, and to my family, who have always been there for me when I needed them. Your love and encouragement have made all the difference.

Abstract

Conformal Prediction (CP) is a principled framework in machine learning for uncertainty quantification. Through the process of calibration or *conformalisation*, CP provides the statistical guarantee of *marginal coverage* on any model's outputs. However, conformalisation is usually applied as a post-processing step after model training. While this process gives the coverage guarantee, it can cause the outputs of the model to be adjusted in unintended ways. If the model were to be *aware* of the conformalisation that follows its training, it could learn to provide outputs which, post-conformalisation, have both the guarantee of coverage *and* are well aligned with the original goals of the model. There has been recent interest into integrating awareness of conformalisation into the training regime. One such approach, called *conformal training*, has been successfully applied to classification tasks. While there have been some examples of similar methods for regression, a thorough and general approach has yet to be described. The goal of this thesis is to provide a general framework to applying conformal training to regression tasks. Called Conformal Training for Regression (CTR), our framework can be applied to any regression model for which the goal is to provide conformal uncertainty estimations. With a specific lens on *conditional coverage*, arguably the most important property in conformal prediction, we present three distinct models based on CTR and demonstrate that with the correct approach, results can be achieved that, for most real datasets, are similar or significantly improve upon state-of-the-art conformal prediction methods.

Table of contents

List of figures

List of tables

Nomenclature

Roman Symbols

$\mathcal{D}$	Dataset
$\mathcal{I}$	Indexing set
$\mathbb{N}$	Space of natural numbers
$\mathbb{R}$	Space of real numbers
$\mathcal{S}$	Score space
$\mathcal{X}$	Feature space
$\mathcal{Y}$	Target space
$\mathcal{Z}$	Cartesian product of feature and target spaces

Greek Symbols

α	Miscoverage rate

Subscripts

cal	Calibration portion of respective set
test	Test portion of respective set
train	Training portion of respective set

Other Symbols

$[N]$	Set of values $\{1, 2, \ldots, N\}$
$\mathcal{N}(\mu, \sigma^2)$	Normal distribution with mean μ and variance σ^2

Acronyms / Abbreviations

AQ-CTR	Adaptive-Quantile CTR
CHR	Conformal Histogram Regression
CP	Conformal Prediction
CQR	Conformalised Quantile Regression
CTR	Conformal Training for Regression
LACP	Locally Adaptive Conformal Prediction
LOOC	Leave-one-out (mini-batch) calibration
MAD	Mean Absolute Deviation
Q-CTR	Quantile CTR
QR	Quantile Regression
SC	Split (mini-batch) calibration
WM-CTR	Width-Minimising CTR
WSC	Worst Slab Coverage

Chapter 1

Introduction

The use of machine learning models is becoming increasingly widespread and their impact on our daily lives more apparent. The potential to simplify arduous parts of our daily lives, optimise healthcare, and automate industry is clear. However, there are downsides to using these models – they are often not interpretable, nor are they always reliable. For this reason, even though they clearly provide value, they do so inconsistently, making the introduction of machine learning models into high-risk domains and infrastructure extremely difficult. The development of reliable, trustworthy, and transparent machine learning models is vital.

With a variety of different approaches (Abdar et al. (2021)), uncertainty quantification (UQ) is an important research direction in machine learning which strives to make models more reliable. In a general sense, its goal is to have models that provide confidence estimations alongside their predictions. By doing so, we get a better understanding of the risk associated with making any decisions based on a prediction. One principled approach for UQ is the conformal prediction (CP) framework (Vovk et al. (2005)). At its core is a statistical calibration method for predictive models that can give finite-sample guarantees on the uncertainty that is predicted. With a user-specified level of allowed risk, conformal calibration or *conformalisation* adjusts the model's predictions into predictive sets or intervals which, on average, contain the true test labels or targets at least as often as the risk level requires.

Machine learning models tend to overfit and thus become overconfident in their predictions in ways that do not align with the underlying data distribution – conformal prediction is an excellent choice to correct for these issues. CP does not make assumptions about the underlying data distribution nor the predictive model that is used. The CP framework has been successfully applied in many real-life scenarios such as robotics (Luo et al. (2022)), drug discovery (Eklund et al. (2015)) and more widely in the medical sciences (Papadopoulos et al. (2009), Alnemer et al. (2016), Pereira et al. (2020)).

With this heightened research interest has come a push to increase the efficiency, accuracy, and reliability of conformal predictors. One new direction, called *conformal training* (Stutz et al. (2021)) has arisen from the need to integrate conformal prediction into the model training phase. Normal CP consists of conformalisation *after* the base model has already been fit. Bridging this gap by making the model aware of the conformalisation that follows its training could significantly reduce inefficiencies in prediction. Conformal training and other *conformal-aware* methods have shown some success in classification (Bellotti (2021), Stutz et al. (2021), Einbinder et al. (2022)) and regression (Bellotti (2020), Lei and Bellotti (2023)).

While there have been conformal training methods introduced for regression, these approaches either optimise for properties that are at odds with model reliability or lack the ability to accommodate task-specific requirements. This thesis provides a general and widely applicable framework for conformal training for regression. We have a specific lens on model reliability through conditional validity, a property for UQ models that faithfully communicate the inherent uncertainty in the data for any input.

We propose models based on this framework which have competitive performance and reliability, not only compared to prior and concurrent work in conformal training for regression, but also compared to state-of-the-art conformal regression methods such as Conformalised Quantile Regression (Romano et al. (2019)) and Conformal Histogram Regression (Sesia and Romano (2021)). Before we begin our background chapter, we provide a list of contributions followed by a thesis outline.

1.1 Contributions

The contributions of this thesis are as follows:

1. In Chapter 2, we provide a comprehensive overview of conformal prediction for regression including recent works in conformal training. Making it a contribution to existing literature, we believe there exists a gap for an in-depth, modern introduction to CP for regression.

2. In Chapter 3, we define a general framework for end-to-end conformal training for regression (CTR). This allows for any task-specific loss function to be used. It also has a more memory- and data-efficient calibration approach compared to that of conformal training for classification by Stutz et al. (2021).

3. We propose three different models, Width-Minimising CTR (Section 3.3), Quantile CTR (Section 3.4) and Adaptive-Quantile CTR (Section 3.5), the last of which uses a novel adjustment of the interval score loss function (Gneiting and Raftery (2007)) to produce skew-adaptive predictions. On a synthetic dataset and multiple real datasets, we present several important results.

   (a) With Width-Minimising CTR, we demonstrate the adverse effect of directly optimising for predictive interval width. Mean interval width is a commonly used metric in CP literature, but it can be at odds with conditional coverage. We present a clear case for why it is the case.

   (b) We demonstrate that Quantile CTR and Adaptive-Quantile CTR provide predictions that are similar to, and in many cases exceed the performance of state-of-the-art conformal regressors.

1.2 Thesis outline

Aside from the introduction, the thesis is structured into four chapters. Chapter 2 contains the background necessary for introducing our general approach to conformal training for regression. We begin by making a comprehensive overview of conformal prediction, including detailed examples for different models and an in-depth modern proof for full conformal prediction inspired by the original from Vovk et al. (2005).

In Chapter 3, we introduce our Conformal Training for Regression (CTR) framework. We also describe in detail the three main models which we derive from it.

Chapter 4 details our experiments. We show results for the CTR-based models including several other baseline models on a synthetic dataset. This is followed by results from experiments on multiple real datasets. We provide a detailed analysis and finish the chapter with a discussion.

Chapter 5 concludes this thesis. We discuss our approach in the context of conformal prediction and suggest future avenues for further research.

Chapter 2

Background

In this chapter, we give an overview of conformal prediction (CP) and conformal training. We begin by showing and proving the marginal coverage guarantee that conformal prediction can provide, and then, with the help of concrete examples, work our way up to the state-of-the-art conformalised quantile regression and conformal histogram regression models. Next, we provide the necessary background for evaluating and improving conformal predictors and highlight the importance of conditional validity for reliable CP. We finish this background chapter with an overview of existing conformal prediction methods which use conformal training. While CP can be successfully applied to classification tasks, we will concentrate on and provide examples from regression.

2.1 Basics of Conformal Prediction

Having highly reliable prediction models that estimate uncertainty is important in high-risk domains. For example, deploying a neural network (NN) model that predicts average blood pressure changes as a result of a specific treatment is subject to being a large source of risk in any drug development pipeline. Even if the model outputs an interval of possible values instead of a single predictive value, there are no guarantees for the reliability of these estimates. This matter is worsened in light of the fact that complex NN models tend to overfit to training data and are thus overconfident in their predictions. The conformal prediction framework allows us to take a model's predictions and convert them into prediction sets that have statistical guarantees on their coverage of the prediction target.

Conformal prediction methods aim to make the use of inference models more reliable by quantifying the uncertainty in model predictions. Although CP is also defined in the classification setting, we will consider a regression model to illustrate how CP works. Consider an m-dimensional input-space $\mathcal{X} \equiv \mathbb{R}^m$ and scalar output space $\mathcal{Y} \equiv \mathbb{R}$. Let us also denote the space of input-output pairs as $\mathcal{Z} \equiv \mathcal{X} \times \mathcal{Y}$. Suppose we have a dataset $\mathcal{D}_{\text{train}} = \{(\mathbf{x}_i, y_i)\}_{i \in \{1,\dots,n\}}$ with n feature vectors $\mathbf{x}_i \in \mathcal{X}$ with respective targets $y_i \in \mathcal{Y}$. Given a test input $\mathbf{x}_{n+1} \in \mathcal{X}$ with label $y_{n+1} \in \mathcal{Y}$, and a coverage level $1 - \alpha \in [0, 1]$, where α is known as the *miscoverage rate*, we wish to find a *conformal* subset $\hat{C}(\mathbf{x}_{n+1}) \subseteq \mathcal{Y}$ such that

$$\mathbb{P}\Big[y_{n+1} \in \hat{C}(\mathbf{x}_{n+1})\Big] \geq 1 - \alpha. \tag{2.1}$$

Conformal prediction defines a procedure to find $\hat{C}$ where this property of *marginal coverage* holds under the assumption of *exchangeability* which states that the joint probability of a sequence of random variables is invariant to permutations of the sequence. This is a weaker property compared to random variables being independent and identically distributed (i.i.d.).

As a side note, given this problem statement, one may be tempted to just set $\hat{C}(\mathbf{x}_{n+1}) = \mathcal{Y}$. Whilst it would not be useful method for quantifying uncertainty, this would clearly have the required property of coverage. As we will see later, conformal prediction methods are not equally powerful and there are several properties of conformal predictors that illustrate the differences between them, one of which is the size of $\hat{C}$, called *width* in the (common) case that $\hat{C}$ is an interval (see Section 2.2).

2.1.1 Full Conformal Prediction

The foundational approach to conformal prediction is *full* or *transductive* conformal prediction (Vovk et al. (2005, 2009)). While this approach in its unmodified form is computationally infeasible for many regression tasks, it serves as an important cornerstone for the more practical approaches that follow it. Given the previously stated problem, for every possible $\hat{y} \in \mathcal{Y}$ we train a model $\hat{f} : \mathcal{X} \to \mathcal{Y}$ on $\mathcal{D}_{\hat{y}} \equiv \mathcal{D}_{\text{train}} \cup \{(\mathbf{x}_{m+1}, \hat{y})\}$. For notational purposes let us explicitly show the model as being conditioned on the dataset that it has been trained on by $\hat{f}(\mathbf{x}; \mathcal{D}_{\hat{y}})$ for any input feature vector $\mathbf{x}$. The only requirement is that $\hat{f}$ is *symmetric*, or, in other words, $\hat{f}(X; \mathcal{D}_{\hat{y}}) = f(X; \pi(\mathcal{D}_{\hat{y}}))$ where π is a permutation operator on the dataset.

We then define a *non-conformity score* function $s : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ which takes in a data point $\mathbf{x}$ and the target label y and calculates how well the prediction $\hat{f}(\mathbf{x}; \mathcal{D}_{\hat{y}})$ approximates y. The simplest candidate definition for such a function is the *absolute residual*

$$s(\mathbf{x}, y; \mathcal{D}_{\hat{y}}) = |y - \hat{f}(\mathbf{x}; \mathcal{D}_{\hat{y}})|.$$

Using the score function s, let us define the sequence $\mathbf{s}^{(\hat{y})} \in \mathcal{S}^{n+1} \equiv \mathbb{R}^{n+1}$ where the element $s_i^{(\hat{y})}$ of $\mathbf{s}^{(\hat{y})}$ is given by $s(\mathbf{x}_i, y_i; \mathcal{D}_{\hat{y}})$ when $i \leq n$ and $s(\mathbf{x}_i, \hat{y}; \mathcal{D}_{\hat{y}})$ when $i = n+1$, and where $\mathcal{S}$ is the score-space. Also, for any $k \in (0, 1]$, we define the k-quantile $q_k(V)$ for a multiset or sequence V of scalar values such that $q_k(V)$ is the $\lceil k|V|\rceil$-th smallest element of V. Note that for multisets, this value can have a multiplicity larger than 1 in V.

Let us define the subset $\hat{C}(\mathbf{x}_{n+1})$ by:

$$\hat{C}(\mathbf{x}_{n+1}) = \{y : s_{n+1}^{(\hat{y})} \leq q_{1-\alpha}(\mathbf{s}^{(\hat{y})})\}.$$

This subset has the required coverage properties. This result is stated as a theorem and proven next. The proof is a reformulation of the original proofs from Vovk et al. (2005) to include detail and clarification relevant to this work.

Theorem 2.1.1 (Full Conformal Prediction Coverage)**.** *For any exchangeable distribution over $\mathcal{Z}^{n+1}$ with density g, and any symmetric model $\hat{f}$ with $\hat{C}$ defined as above for some score function s we have:*

$$\mathbb{P}_{((\mathbf{x}_1,y_1),\ldots,(\mathbf{x}_{n+1},y_{n+1}))\sim g}\Big[y_{n+1} \in \hat{C}(\mathbf{x}_{n+1})\Big] \geq 1 - \alpha.$$

Proof. Let us show that

$$\mathbb{P}_{((\mathbf{x}_1,y_1),\ldots,(\mathbf{x}_{n+1},y_{n+1}))\sim g}\Big[s_{n+1}^{(y_{n+1})} \leq q_{1-\alpha}(\mathbf{s}^{(y_{n+1})})\Big] \geq 1 - \alpha \tag{2.2}$$

with the main result deriving from this automatically by the definition of $\hat{C}$. Instead of a joint probability distribution g over the dataset, we can consider the probability distribution canonically induced by g over score sequences with the joint probability density $g_s : \mathcal{S}^{n+1} \to [0, 1]$. This gives us that we can equivalently to Eq. 2.2 prove

$$\mathbb{P}_{\mathbf{s}\sim g_s}\Big[s_{n+1} \leq q_{1-\alpha}(\mathbf{s})\Big] \geq 1 - \alpha \tag{2.3}$$

where we remove the superscripts for brevity. We have that since $\hat{f}$ is symmetric and g is exchangeable, then permuting the elements in $\mathbf{s}$ does not change the probability of $\mathbf{s}$. Hence g_s is also exchangeable.

Now let us disentangle the notions of value and order for any sampled sequence of scores $\mathbf{s}$. First, let us define the set $\mathcal{M}$ all possible multisets of cardinality $n + 1$ such that for any multiset $S \in \mathcal{M}$ we have that all elements $s \in S$ are also in the score space $\mathcal{S}$. In other words, $\mathcal{M}$ contains all possible multisets S with $n+1$ (potentially repeating) elements from $\mathcal{S}$.

Next, we consider the transformation of $S \in \mathcal{M}$ into an ordered sequence. Let us define the set Π_S of all *unique* permutations π of the elements of S, such that we let $\pi(S)$ represent the respective sequence in the given order. We have that $|\Pi_S|$ is given by multinomial coefficients, with the special case, for which we reserve $\Pi \equiv \Pi_S$, when S has no duplicate elements and $|\Pi_S| = (n+1)!$. Finally, due to exchangeability, we sample the permutations from the uniform distribution over the set of unique permutations. We denote the uniform distributions $u \equiv U(\Pi)$ and $u_S \equiv U(\Pi_S)$.

This means that for an arbitrary $\pi \in \Pi_S$, we can define the probability density $h : \mathcal{M} \to [0,1]$ by

$$h(S) \equiv \frac{g_s(\pi(S))}{u_S(\pi)} = g_s(\pi(S))|\Pi_S|$$

for any $S \in \mathcal{M}$.

This definition allows us to write $\mathbb{P}_{\mathbf{s}\sim g_s}\Big[s_{n+1} \leq q_{1-\alpha}(\mathbf{s})\Big]$ as the marginalisation of $\mathbb{P}_{\pi\sim u_S}\Big[\pi(S)_{n+1} \leq q_{1-\alpha}(\pi(S)), S\Big]$ over all possible values of $S \in \mathcal{M}$ where $\pi(S)_{n+1}$ is simply the final element of the sequence $\pi(S)$. This effectively means that we separate the sampling of the sequence $\mathbf{s}$ into first sampling a multiset S followed by sampling a permutation π. Our marginalisation can be written as

$$\mathbb{P}_{\mathbf{s}\sim g_s}\Big[s_{n+1} \leq q_{1-\alpha}(\mathbf{s})\Big] = \int_{S\in\mathcal{M}} \mathbb{P}_{\pi\sim u_S}\Big[\pi(S)_{n+1} \leq q_{1-\alpha}(\pi(S)) \,|\, S\Big] h(S) dS.$$

We see in the right-hand side that

$$\begin{aligned}
\mathbb{P}_{\pi\sim u_S}\Big[\pi(S)_{n+1} \leq q_{1-\alpha}(\pi(S)) \,|\, S\Big] &= \mathbb{P}_{\pi\sim u}\Big[\pi(S)_{n+1} \leq q_{1-\alpha}(\pi(S)) \,|\, S\Big] \\
&= \sum_{\pi\in S_{m+1}} \mathbb{P}\Big[\pi(S)_{n+1} \leq q_{1-\alpha}(\pi(S)) \,|\, S, \pi\Big] u(\pi) \\
&= \frac{1}{(n+1)!} \sum_{\pi\in S_{m+1}} \mathbb{P}\Big[\pi(S)_{n+1} \leq q_{1-\alpha}(\pi(S)) \,|\, S, \pi\Big] \\
&\geq \frac{n!}{(n+1)!} \lceil (1-\alpha)(n+1) \rceil \\
&= \frac{\lceil (1-\alpha)(n+1) \rceil}{(n+1)} \geq 1-\alpha.
\end{aligned} \tag{2.4}$$

In the above equation, the first equality arises from the fact that $\mathbb{P}(\pi(S))$ for a specific value of π is invariant to whether $\pi \sim u$ or $\pi \sim u_S$. The second equality stems from the fact that we can simply count all of the permutations. The third equality arises from the exchangeability property, i.e., with the uniform distribution, all permutations are equiprobable. For the first inequality we see that $\mathbb{P}\Big[\pi(S)_{n+1} \leq q_{1-\alpha}(\pi(S)) \,|\, S, \pi\Big]$ collapses into an indicator function, since S, π determine whether the inequality holds. Thus we see that this sum counts the number of permutations π where $\pi(S)_{n+1}$ is less than or equal to the $\lceil(1-\alpha)(n+1)\rceil$-th smallest element of S. This amounts to at least $n! \cdot \lceil(1-\alpha)(n+1)\rceil$ cases when accounting for the possibility of values s in S with $m_S(s) > 1$, giving the final inequality[1]. Thus, we have that

$$\begin{aligned}\mathbb{P}_{\mathbf{s}\sim g_s}\Big[s_{n+1} \leq q_{1-\alpha}(\mathbf{s})\Big] &\geq \int_{S\in\mathcal{M}} (1-\alpha)h(S)dS \\ &= 1-\alpha\end{aligned}$$

which gives our desired result. □

It is important to stress that this approach is *distribution-free*. This means that we do not need to know the distribution which generated the data, as long as exchangeability holds. This highlights the fact that conformal prediction is a powerful tool that can be applied in a wide array of domains. In the case of full conformal prediction however, we have the caveat that the model $\hat{y}$ needs to be trained over all $y \in \mathcal{Y}$ which, in the usual case of $\mathcal{Y} = \mathbb{R}$, is infeasible. Whilst there are approximation-based approaches to full conformal prediction Shafer and Vovk (2008), there is a more computationally appealing approach used in practise known as *split conformal prediction*.

2.1.2 Split Conformal Prediction

First introduced by Papadopoulos et al. (2002) and also known as *inductive* conformal prediction, split conformal prediction is a computationally more efficient approach than full conformal prediction. Split CP does not require training multiple models for every possible test label by *splitting* the training dataset into a training set $\mathcal{D}_{\text{train}}$ to fit a single model $\hat{f}$ and a calibration set $\mathcal{D}_{\text{cal}}$ to conformalise the predictions of f. Whilst all of the original training data cannot be seen by the training process, only a single model has to be fitted compared to the numerous models required for full conformal prediction.

For split conformal prediction, we have a training set $\mathcal{D}_{\text{train}}$ of size m, a calibration set $\mathcal{D}_{\text{cal}}$ of size n, a non-conformity score function s and an error rate α. Also, we have a test dataset $\mathcal{D}_{\text{test}} \subset \mathcal{Z}^m$ of m input-output pairs. We first fit a model $\hat{f}$ on $\mathcal{D}_{\text{train}}$

[1] Note that if S never has duplicate values, i.e. g_s is continuous, then the first inequality in Eq. 2.4 becomes an equality and we also get a final upper bound of $\frac{1}{n+1} + 1 - \alpha$.

Algorithm 1: Split Conformal Prediction

```
Input: $\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{cal}}$, $\alpha$, $s$ (score function), fit (model training algorithm), $\mathbf{x}$
       (test input)
Result: $\hat{C}$
$n \leftarrow |\mathcal{D}_{\text{cal}}|$;
Instantiate scalar sequence $\mathbf{s}$ of length $n$;
$\hat{f} \leftarrow \texttt{fit}(\mathcal{D}_{\text{train}})$;
for $(\mathbf{x}_i, y_i) \in \mathcal{D}_{cal}$ do
    $s_i \leftarrow s(\mathbf{x}_i, y_i; \hat{f})$;
end
$r \leftarrow q_{\lceil((n+1)(1-\alpha)\rceil/n}(\mathbf{s})$;
$\hat{C} \leftarrow \{y : s(\mathbf{x}, y; \hat{f}) \leq r\}$;
```

and then calculate the scores $s_i = s(\mathbf{x}_i, y_i; \hat{f})$ over all $(\mathbf{x}_i, y_i) \in \mathcal{D}_{\text{cal}}$. We then calculate the $\lceil((n+1)(1-\alpha)\rceil$-th smallest score r and compute the conformal subset $\hat{C}$ per test input $\mathbf{x}$ with target $y \in \mathcal{Y}$ such that $s(\mathbf{x}, y; \hat{f})$ is not greater than r. If s is defined to be the absolute residual between y and $\hat{f}(\mathbf{x})$ for some input-label pair $(\mathbf{x}, y)$, then $\hat{C}$ becomes an interval around x in the general case when $\mathcal{Y} = \mathbb{R}$.

The pseudo-code for split CP can be seen in Algorithm 1. To contrast with full conformal prediction, other than the different dataset setup, the quantile which we calculate is slightly different, however. This is simply a correction to account that now the test point is not in the dataset that we calibrate with. Also, we drop the requirement that $\hat{f}$ is symmetric – this arises since proving coverage requires enumerating the permutations of elements in $\mathcal{D}_{\text{cal}}$, which, compared to $\mathcal{D}_{\text{train}}$, do not affect the fitted model $\hat{f}$.

Theorem 2.1.2 (Split Conformal Prediction Coverage)**.** *Consider an exchangeable distribution with density g over $\mathcal{Z}^{m+n+1}$ of sequences of input-label pairs with the first m pairs being the $\mathcal{D}_{train}$, the following n pairs being $\mathcal{D}_{cal}$ and the final element being the test input-label pair. For any model $\hat{f}$ and non-conformity score function s with $\hat{C}$ defined as above, we have:*

$$\mathbb{P}_{((\mathbf{x}_1,y_1),\ldots,(\mathbf{x}_{m+n+1},y_{m+n+1}))\sim g}\Big[y_{m+n+1} \in \hat{C}(\mathbf{x}_{m+n+1})\Big] \geq 1-\alpha.$$

The proof for this result is very similar to the full conformal prediction case. As mentioned before, the only significant difference is that the proof requires enumerating permutations over the calibration set instead of $\mathcal{D}_y$.

Example. We now demonstrate split conformal prediction using a very simple synthetic example with $\mathcal{X} = [0,1]$, $\mathcal{Y} = \mathbb{R}$ and $\alpha = 0.15$. We generate data using the function $f(x) \sim \mathbb{1}_{x \le \frac{1}{2}} \mathcal{N}(\cos(5x), \frac{1}{4}) + \mathbb{1}_{x > \frac{1}{2}} \mathcal{N}(1, \frac{1}{100})$ where $\mathbb{1}$ is the indicator variable and $x \in \mathcal{X}$. We sample 100 training input-label pairs and 30 calibration input-label pairs. We use an algorithm that fits a polynomial of degree 4 to the data, resulting in $\hat{f}$. The score function is simply the absolute error. We perform conformalisation using Algorithm 1 and show the results in Figure 2.1.

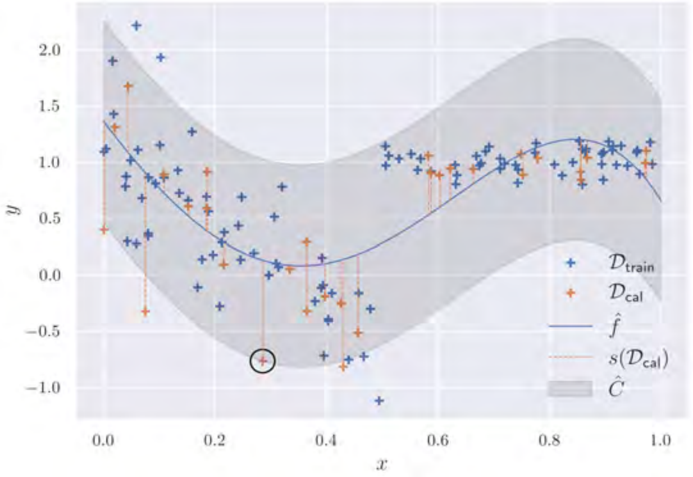

Fig. 2.1 Split conformal prediction on a synthetic dataset. Conformal band ($\alpha = 0.15$) is shaded and the calibration datapoint respective to the quantile r of the calibration scores is circled. The scores are illustrated by the magnitude of the vertical dashed lines.

As can be seen, split conformal prediction adds an equally wide band around the polynomial model $\hat{f}$. Since $\lceil (n+1)(1-\alpha) \rceil = \lceil 31 \cdot 0.85 \rceil = 27$, the circled datapoint denotes the 27th calibration datapoint closest to the respective model predictions – note the three calibration datapoints outside the band.

One simple analysis that we can perform for this example is to measure the coverage of the conformal band for unseen datapoints. For a large amount (10^6) of test points, we calculate that the conformal band covers around 92% of the test points. Even though the underlying model is poorly fitted, the conformalisation process still gives prediction intervals with adequate coverage. This illustrates the power of conformal prediction, and any kind of model can be made more trustworthy and dependable. We note that we expand on the evaluation of conformal prediction methods in Section 2.2.

On the other hand, for the case of simple split CP, this example highlights the fact that the constant-width conformal band can be ineffective for heteroskedastic data.

We next turn extensions of basic split conformal prediction methods which can output varying-width conformal bands.

2.1.3 Locally Adaptive Conformal Prediction

In the previous sections we saw that the non-conformity score function s has no special restrictions placed upon it. The absolute residual $|y - \hat{f}(\mathbf{x})|$ is a convenient form for the score function since it allows the conformal sets be defined as intervals defined as

$$\hat{C}(\mathbf{x}) = [\hat{f}(\mathbf{x}) - r, \hat{f}(\mathbf{x}) + r],$$

where r is the desired quantile value. In order to become free of the apparent restriction of the constant-width conformal bands, the score function could instead be normalised such that

$$s(\mathbf{x}, y) = \frac{|y - \hat{f}(\mathbf{x})|}{\sigma(\mathbf{x})}, \tag{2.5}$$

where σ is a function which predicts the expected error, such as the average absolute residual of $\hat{f}$ at a point x (Papadopoulos et al. (2008), Johansson et al. (2015)). This function is usually fitted separately either on the training set, which can be subject to overfitting behaviour, or another held out dataset which we can denote as $\mathcal{D}_{\text{res}}$. In this case, noticing that $\sigma(\mathbf{x})s(\mathbf{x}, y)$ is the original absolute residual, we can define the conformal sets as intervals of the form

$$\hat{C}(\mathbf{x}) = [\hat{f}(\mathbf{x}) - \sigma(\mathbf{x})r, \hat{f}(\mathbf{x}) + \sigma(\mathbf{x})r],$$

where r is the desired quantile value of the normalised scores. We refer to this approach by Locally Adaptive Conformal Prediction (LACP).

One intuitive way to visualise this process is by first imagining that $\sigma(x)$ defines a scaling of the output space at each x. Then in this scaled output space, fixed-width conformalisation is done. Finally, the scaling procedure is inverted to yield varying-width conformal intervals.

Example. We use the same dataset as from the previous section. In addition to the previous procedure, we fit a polynomial of degree 4 on the absolute residuals of the training set given the model $\hat{f}$. We then calculate the conformal intervals as shown below to yield the conformal band. The results are shown in Figure 2.2.

As we can see here, in a large part of $\mathcal{X}$, the conformal band's width is smaller than in Figure 2.1. The predictive intervals have clearly adapted to the varying levels of noise even though the underlying model is extremely simple. One downside to using

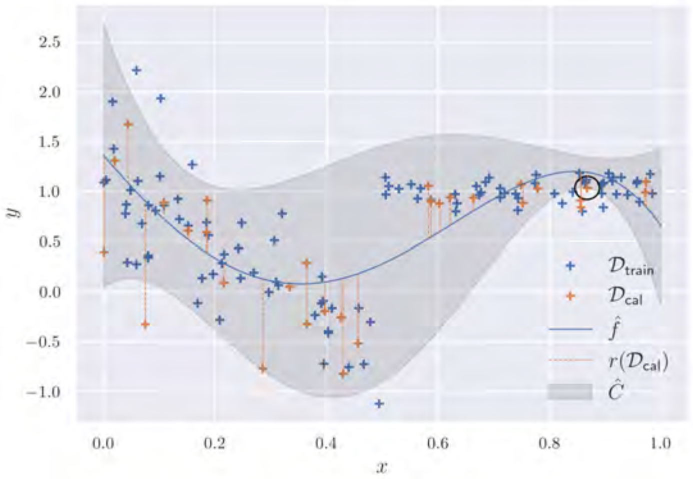

Fig. 2.2 Locally adaptive split conformal prediction on the synthetic dataset from Figure 2.1. Conformal band ($\alpha = 0.15$) is shaded and the calibration datapoint respective to the quantile r of the calibration scores is circled. The residuals are illustrated by the magnitude of the vertical dashed lines.

this approach is that there are two disjoint training runs: one for the mean prediction and the other for the prediction for the normaliser of the score function. The normaliser can only do so much if the mean predictions are not faithful to the actual data. We now turn to a much more advanced method for conformal regression that does not have this downside among others and produces significantly improved adaptive results.

2.1.4 Conformalised Quantile Regression

In this section we describe one of the most popular state-of-the-art approaches for conformal regression. Conformalised Quantile Regression (CQR) (Romano et al. (2019)) shows significant improvements to the LACP procedure described above. Whilst this procedure is also locally adaptive, we keep it distinct from LACP.

The main improvement that CQR introduces is that of using a quantile regression model (Koenker and Bassett Jr (1978)) as its base model instead of a point-predictor. A quantile regression (QR) model usually outputs quantile predictions $\hat{q}_\alpha(\mathbf{x})$ for different levels α. The target $q_\alpha(\mathbf{x}) \in \mathcal{Y}$ is defined as the value below which lies α of the conditional probability mass of the true target distribution $p(y|\mathbf{x})$. QR models are usually trained via the pinball loss (Steinwart and Christmann (2011)) for each level α

$$l_\alpha(y, \hat{y}) = \mathbb{1}_{y-\hat{y}>0}\alpha(y - \hat{y}) + \mathbb{1}_{y-\hat{y}\leq 0}(1 - \alpha)(\hat{y} - y), \tag{2.6}$$

where y is the true quantile value $q_\alpha(\mathbf{x})$ and $\hat{y}$ is the prediction $\hat{q}_\alpha(\mathbf{x})$.

Conformalised Quantile Regression uses, by default, as its base model a QR model which predicts two quantiles, $q_{\alpha_{\text{lo}}}$ and $q_{\alpha_{\text{hi}}}$. The conformalisation procedure follows split conformal prediction with a held-out calibration set. The non-conformity score is different to the point-predictor model and is defined as

$$s(\mathbf{x}, y) = \max(\hat{q}_{\alpha_{\text{lo}}}(\mathbf{x}) - y, y - \hat{q}_{\alpha_{\text{hi}}}(\mathbf{x})). \tag{2.7}$$

As before, the conformalisation procedure in this case involves finding the desired quantile r such that at least $1 - \alpha$ of the calibration scores are below r. Similarly, the conformalised prediction interval for some test point x are thus defined to be $\hat{C}(\mathbf{x}) = \{y \mid s(\mathbf{x}, y) \leq r\}$. Considering the score function s, the interval form of the prediction set can simply be derived to be

$$\hat{C}(\mathbf{x}) = [\hat{q}_{\alpha_{\text{lo}}}(\mathbf{x}) - r, \hat{q}_{\alpha_{\text{hi}}}(\mathbf{x}) + r].$$

The reason as to why CQR performs much better than locally adaptive point-prediction conformalisation arises due to the following. For one, the base model in CQR is trained to estimate uncertainty from the get-go. This contrasts with the point-predictor which tries to estimate the mean only and where the uncertainty estimate comes from an additional model trained after the base model. What is more, the mean estimate cannot capture any asymmetry present in the conditional probability distribution $p(y|\mathbf{x})$. The authors of CQR further highlight the fact that the uncertainty estimate for the point-predictor is usually computed by training on the same training set as the point predictor, yielding biased estimates of the uncertainty.

Example. We repeat our previous example, this time using conformalised quantile regression. We fit two polynomials of degree 4 to get predictions $\hat{q}_{0.075}$ and $\hat{q}_{0.925}$ for the two quantiles with the same miscoverage rate $\alpha = 0.15$ as before. We perform conformalisation using Equation 2.7 as the score function. The results are shown in Figure 2.3.

Although, due to the simple base predictors, it is not immediately clear that this approach for conformal regression is better, one difference that is clear between LACP and CQR in this example is the number of training points outside the conformal band – LACP seems to be overconfident at around $x = 0.9$. Using more complex data alongside more expressive models like random forests and neural networks indeed shows that CQR tends to be significantly more faithful to the underlying data distribution; we refer the reader to the original work (Romano et al. (2019)) for a comparison of the

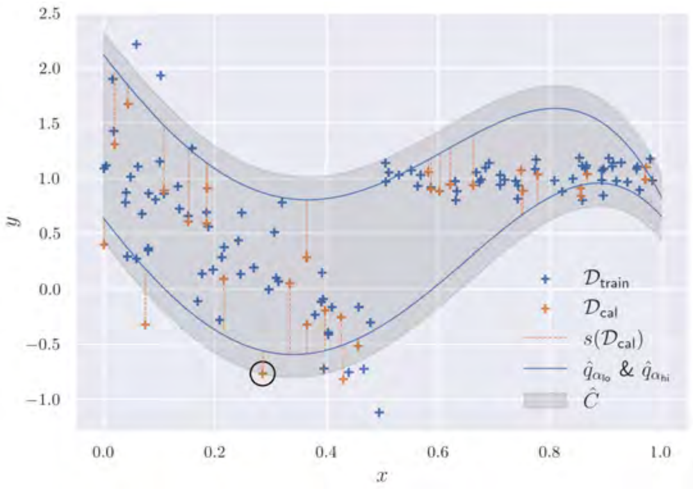

Fig. 2.3 Conformalised quantile regression on the synthetic dataset from Figure 2.1. Conformal band ($\alpha = 0.15$) is shaded and the calibration datapoint respective to the quantile r of the calibration scores is circled. The lower and upper quantile predictions are shown in blue. The scores are illustrated by the magnitude of the vertical dashed lines.

two approaches on real datasets. Also, we note that there are several variants of the quantile score function – for more, see Sesia and Candès (2020). We now turn to the last model in this section, which shows improvements over CQR with both methods being considered state-of-the-art in conformal prediction for regression.

2.1.5 Conformal Histogram Regression

In this section, we introduce the Conformal Histogram Regression (CHR) model by Sesia and Romano (2021). Often our target distributions are asymmetric or *skewed*. We might wish to have predictive intervals which have valid coverage whilst being as tight as possible. Hence, if our data is skewed, giving centred predictive intervals, such as estimating $q_{0.05}$ and $q_{0.95}$ may not yield as tight intervals as predicting, say $q_{0.0}$ and $q_{0.9}$. CHR is a solution to this problem. We will provide a simplified outline of the method and refer the reader to the main paper for more.

Conformal Histogram Regression relies on the underlying model outputting a conditional histogram of the target variable, a discretisation of the conditional probability distribution $p(y|\mathbf{x})$. More specifically, for a given ordered grid $B \equiv (b_i)_{i\in\{0,\dots,N\}} \in \mathcal{Y}^{N+1}$ with histogram size N that divides the relevant $\mathcal{Y}$ into parts, we define the conditional histogram for an input $\mathbf{x} \in \mathcal{X}$ as $\Pi(\mathbf{x}) \equiv \{\pi(\mathbf{x}, i) = \mathbb{P}[Y \in (b_{i-1}, b_i]]\}_{i\in\{1,\dots,N\}}$ with the random variable Y taking values from $\mathcal{Y}$. We refer to the model's probability estimation of a bin at index i as $\hat{\pi}(\mathbf{x}, i)$ which is an estimate of the probability that the true target variable y is between b_{i-1} and b_i. We refer to the whole set of bin estimates, i.e., the respective histogram as $\hat{\Pi}(\mathbf{x})$.

CHR defines a calibration method for models which give estimates in this way. The calibration consists of 1) converting for each $(\mathbf{x}, y) \in \mathcal{D}_{\text{cal}}$ the conditional histogram $\hat{\Pi}(\mathbf{x})$ into a nested sequence of intervals of increasing size, 2) finding the index of the interval in this sequence for each $\mathbf{x}$ which covers the respective y – this yields the scores over the calibration set – and 3) finding the appropriate quantile r_{cal} of these scores as shown in previous sections. This index r_{cal} is then used for inference on a test point $\mathbf{x} \in \mathcal{D}_{\text{test}}$ which consists of 1) converting the respective $\hat{\Pi}(\mathbf{x})$ into a nested sequence of intervals and 2) choosing the interval at index r_{cal}. This interval is then used as the prediction. We now describe calibration and inference in more detail.

For the model to output calibrated predictions at some target miscoverage rate α, these histogram predictions need a score function. Namely, the authors of CHR define how to generate a sequence $S(\mathbf{x}) \equiv \left((i_j^{\text{lo}}(\mathbf{x}), i_j^{\text{hi}}(\mathbf{x}),)\right)_{j \in [K]}$ of some predetermined number K of nested index-pairs from $[N]^2$ such that $i_j^{\text{lo}}(\mathbf{x}) \leq i_j^{\text{hi}}(\mathbf{x})$, $i_j^{\text{lo}}(\mathbf{x}) \geq i_{j+1}^{\text{lo}}(\mathbf{x})$, and $i_j^{\text{hi}}(\mathbf{x}) \leq i_{j+1}^{\text{hi}}(\mathbf{x})$ for all $j \in [K]$. Each index pair $(i_j^{\text{lo}}(\mathbf{x}), i_j^{\text{hi}}(\mathbf{x}))$ refers to the interval $(b_{i^{\text{lo}}{}_j(\mathbf{x})-1}, b_{i_j^{\text{hi}}(\mathbf{x})}]$ in $\mathcal{Y}$. Thus, $S(\mathbf{x})$ defines a sequence of nested intervals based on the conditional histogram. The nested intervals are generated in such a way that that, for each $j \in [K]$ and a predetermined level of coverage α_j, we have $\sum_{i=i_j^{\text{lo}}(\mathbf{x})}^{i_j^{\text{hi}}(\mathbf{x})} \hat{\pi}(\mathbf{x}, i) \geq \alpha_j$.

Having made this sequence, the nonconformity score that is calculated is simple. For some $(\mathbf{x}, y) \in \mathcal{D}_{\text{cal}}$, we have $s(\mathbf{x}, y)$ is simply the smallest $j \in [K]$ such that $b_{i_j^{\text{lo}}(\mathbf{x})-1} < y \leq b_{i_j^{\text{hi}}(\mathbf{x})}$. Having calculated the appropriate quantile $r_{\text{cal}} \in [K]$ of the scores, this value is then used for all test set predictions $\mathbf{x} \in \mathcal{D}_{\text{test}}$ by selecting the r^{th} conditional nested interval $(b_{i_r^{\text{lo}}(\mathbf{x})-1}, b_{i_r^{\text{hi}}(\mathbf{x})}]$ as the calibrated prediction.

As the authors of CHR show, this approach results in predictive intervals that are smaller thanks to its adaptivity to data skewness through the generation of nested intervals. Also, due to having access to the conditional histogram, the calibration sequence is more unbiased towards different inputs $\mathbf{x}$ and any heteroskedasticity present in the data. This yields improved conditional coverage – a very important property describing predictor reliability that we will discuss in more depth in the next section.

Finally, we make a note on how the prediction of the conditional histogram can be learnt in the first place. The authors of CHR demonstrate how to train a quantile regression model on a grid of target quantile levels q_α (with the default of $\alpha \in (0.01, 0.02, ..., 0.99)$) using the pinball loss. They then convert the predictions $\hat{q}_\alpha$ into the respective histogram of size N simply by considering the distances between adjacent quantile predictions. We note that our description of CHR is a simplified version of the original with the authors making several adjustments to improve the stability and accuracy of the approach.

So far we have discussed the presented models in qualitative terms. Even though we have visualised predictions for some of these models using simple examples, we wish to understand and compare their performance on more complex datasets. In the next section we turn to describing some of the most important approaches for evaluating and improving the performance of conformal prediction methods.

2.2 Evaluating and Improving Conformal Predictors

Several metrics can be used to compare models that have undergone conformalisation. In this section we introduce some of these beginning with a short overview of what makes a good conformal predictor. We finish this section with a short overview of Mondrian conformal prediction.

To understand whether a conformal predictor is a good one, it is important to go beyond the notion of *marginal validity* – the possession of marginal coverage. Since this is given for any base model that has undergone conformalisation, we must also look at other measures for evaluation.

One of the most widely used metrics that accompany coverage in model comparisons is average prediction set size, or, in the case of regression, mean interval width. Many works on conformal prediction use this metric (e.g., Romano et al. (2019), Sesia and Candès (2020), Lei et al. (2018)). However, small mean interval width for a model with valid coverage does not necessarily mean it is better. In fact, it is possible to have a marginally valid model which is extremely misleading in terms of the uncertainty it predicts whilst having better mean interval width than other models. We will illustrate this phenomenon in Chapter 4.

From the prior examples we showed that a good conformal predictor is one which faithfully communicates the uncertainty present in the data. Instead of just the marginal coverage guarantee

$$\mathbb{P}\Big[y \in \hat{C}(\mathbf{x})\Big] \geq 1 - \alpha,$$

a stronger guarantee would be that for all points $(\mathbf{x}, y) \in \mathcal{Z}$ we have

$$\mathbb{P}[\, y \in \hat{C}(\mathbf{x}) \mid \mathbf{x}\,] \geq 1 - \alpha.$$

Called *conditional validity*, this initially does not look too different from marginal validity. However, conditional validity is a significantly better property for a model to have than small mean interval width – it gives the guarantee that the model communicates its uncertainty consistently in an unbiased way.

Conditional coverage is not simple to measure, nor does there exist a calibration process that guarantees it; conditional validity cannot be achieved through conformalisation for finite data in the general case (Vovk (2012), Lei and Wasserman (2014), Foygel Barber et al. (2021)). As an intuition, the reason boils down to the fact that there will be regions of the input space where samples $\mathbf{x} \in \mathcal{X}$ have non-zero probability $p(\mathbf{x})$, but from where a sample has not been drawn – therefore the conditional target distribution $p(y|\mathbf{x})$ for those samples cannot be known and thus predicted with any guarantee of coverage unless assumptions regarding smoothness are made.

For this reason, properties which interpolate between marginal and conditional coverage, such as local conditional validity (Lei and Wasserman (2014)) or restricted conditional validity (Foygel Barber et al. (2021)), are instead measured or optimised. These measures effectively partition the input space into subgroups and measure coverage in each subgroup – if a model has a coverage close to the target coverage within each subgroup, then it can be said it has improved conditional coverage. There are several approaches to partition the input space. One of them batches together inputs from $\mathcal{X}$ depending on the respective predictive interval width (e.g., Feldman et al. (2021)). Another way is to use the K-means algorithm on the dataset inputs (e.g., Izbicki et al. (2019), Alaa et al. (2023)). Of course, both approaches require choosing a number k of subgroups beforehand – any metrics that are measured depend on this – hence aggregating results across multiple different values for k yields more reliable results.

One of the most popular metrics for approximating conditional coverage is worst-slab coverage (Cauchois et al. (2021), with examples from Romano et al. (2020), Sesia and Romano (2021), Feldman et al. (2021)). For a partitioning of the input space, it gives the worst coverage across all subgroups, i.e., slabs. This is an elegant measure indicating the lower bound on the model reliability.

Before we move on to Conformal Training, we finish this section by mentioning Mondrian Conformal Prediction (Vovk et al. (2003)) due to its ties to improving conditional coverage through partitioning the data before conformalisation. This approach partitions the calibration set $\mathcal{D}_{\text{cal}}$ into disjoint subsets or subgroups using a predetermined splitting approach, i.e., a *Mondrian taxonomy.* Calibration is then made for each group separately. Inference consists of finding the respective subgroup in the taxonomy a test input $\mathbf{x}$ belongs to and applying the corrective quantile of the scores calculated during calibration for that specific subgroup.

This approach has yielded multiple methods such as class-conditional Mondrian classification where the Mondrian taxonomy is based on class predictions (Shi et al. (2013)) and Mondrian conformal regressors which use the previously introduced LACP method and establish the Mondrian taxonomy on the predicted uncertainty levels $\sigma(\mathbf{x})$ (Boström and Johansson (2020)).

2.3 Conformal Training Methods

One limitation of the conformalisation procedure is that it is performed after model training. This means that the base model lacks *conformal-awareness* – it is unaware of conformalisation and of its effect on the final conformalised outputs. A way to bring this awareness to the model is by *simulating* conformalisation during training. There are several potential advantages of doing this. For one, this can help the base model adjust its predictions in ways that yields improved results post-conformalisation.

Another potential advantage of simulating conformalisation during training is the increased freedom to choose a task-specific loss when using conformal prediction. This flexibility is made possible since the conformalised outputs of a conformal-aware model can be more faithful to the original loss function than those of a conformal-unaware model. For example, consider a task with a specific loss which aims to reduce model bias on certain minority subgroups of the data. If the model lacks conformal-awareness, then the conformalisation process that follows its training may accentuate certain biases which are intended to be avoided. However, if conformalisation were to be simulated during training, the model could adjust its outputs in a way such that when we finally conformalise after training, model bias is not worsened.

We now describe existing classification and regression approaches which integrate conformalisation or conformal-awareness into model training. As our proposed method also utilises this approach, we perform a deeper comparison of our and related methods in Section 3.1.

2.3.1 Classification

The notion of training models in a conformal-aware fashion is a new topic in conformal prediction literature. In the classification setting, Stutz et al. (2021) introduce *conformal training* which they denote “ConfTr”. This is a process whereby conformalisation is “simulated” on a classifier $\hat{f}$ such as a neural network. This is done in a differentiable way on mini-batches during training.

Since the authors of ConfTr were first to propose such an approach, we refer to this and other similar methods throughout this thesis by the name *conformal training*. We will now describe a top-level overview of the approach. We refer the reader to the original work for more details; moreover, for an introduction to conformal prediction for classification, we refer the reader to Angelopoulos and Bates (2021).

At the core of conformal training is what we denote *mini-batch calibration*. This is done during training by first splitting the mini-batch at hand into two parts, a calibration subset $B_{\text{cal}} \subset \mathcal{Z}$ and a predictive subset $B_{\text{pred}} \subset \mathcal{Z}$. We denote $B_{\text{pred}}^{(\mathcal{X})}$ and $B_{\text{pred}}^{(\mathcal{Y})}$ as the features and respectively the targets of B_{pred}. We do the same to yield $B_{\text{cal}}^{(\mathcal{X})}$ and $B_{\text{cal}}^{(\mathcal{Y})}$ from B_{cal}. An inductive conformalisation procedure is done by first calculating the set of scores S_{cal} for all predictions $\hat{f}(\mathbf{x})$ for $\mathbf{x} \in B_{\text{cal}}^{(\mathcal{X})}$.

This is followed by taking the relevant quantile r_{cal}, i.e., the $\lceil(|S_{\text{cal}}|+1)(1-\alpha)\rceil$-smallest element of S_{cal}[2]. Then, the predictions $\hat{f}(B_{\text{pred}})$ are adjusted using r_{cal}, yielding mini-batch calibrated predictions. These adjusted predictions are then used to calculate the loss and update model parameters. It must be noted that calibration on a calibration set $\mathcal{D}_{\text{cal}}$ is still needed to be done after training, because the model effectively trains on $B_{\text{cal}} \subset \mathcal{D}_{\text{train}}$ and can thus overfit to them.

The authors of ConfTr describe multiple avenues for a loss function – with conformal training, the original loss function that otherwise would be used on $\hat{f}$ is effectively freed up. One approach they take is minimising the size of the locally conformalised predictive sets. This allows for the base model to make outputs such that, after conformalisation on a calibration set $\mathcal{D}_{\text{cal}}$, would result in as small as possible predictive sets whilst retaining the marginal coverage guarantees. However, as we note in later sections, this is a dangerous loss function to use on its own since it can damage conditional validity.

[2] Note that the ConfTr method employs a differentiable sorting approach (for example, see Blondel et al. (2020)) which yields a smoothed r_{cal}. This allows the score quantile estimation to be more stable whilst also allowing the loss gradient to flow through multiple scores instead of one.

2.3.2 Regression

In the regression setting, Lei and Bellotti (2023) train a point-predictor $\hat{f}$ and an uncertainty estimator $\hat{\sigma}$ together as a single neural network model by employing conformal training. They denote their approach Directly Optimised Inductive Conformal Regression (DOICR). As in Stutz et al. (2021), they also split the mini-batch into two, B_{cal} and B_{pred} and perform mini-batch calibration as described before.

More specifically, after splitting the mini-batch, they follow by a forward pass on both subsets to yield point-predictions and uncertainty estimations. They then calculate the scores

$$S_{\text{cal}} = \left\{ \frac{y - \hat{f}(\mathbf{x})}{\exp(\hat{\sigma}(\mathbf{x}))} \mid (\mathbf{x}, y) \in B_{\text{cal}} \right\}$$

and calculate its respective quantile r_{cal} – the $\lceil(|S_{\text{cal}}| + 1)(1 - \alpha)\rceil$-smallest element of S_{cal}.

The forward pass is then completed by calibrating $\hat{f}(\mathbf{x})$ using r for all $\mathbf{x} \in B_{\text{pred}}^{(\mathcal{X})}$. The predictive interval for an input $\mathbf{x}$ from $B_{\text{pred}}^{(\mathcal{X})}$ is, similarly to Equation 2.5,

$$[\hat{f}(\mathbf{x}) - r_{\text{cal}} \exp(\hat{\sigma}(\mathbf{x})), \hat{f}(\mathbf{x}) + r_{\text{cal}} \exp(\hat{\sigma}(\mathbf{x}))].$$

The loss for this approach is computed to minimise interval widths, equivalent to $2r_{\text{cal}} \exp(\hat{\sigma}(\mathbf{x}))$ for an input $\mathbf{x}$. This results in the general form of the loss on the mini-batch of

$$\frac{2r_{\text{cal}}}{|B_{\text{pred}}|} \sum_{\mathbf{x} \in B_{\text{pred}}^{(\mathcal{X})}} \exp(\hat{\sigma}(\mathbf{x})).$$

However, while this loss indeed improves predictive efficiency, this approach can cause a false sense of security – it tends to reduce conditional validity whilst retaining marginal validity. In our thesis, we demonstrate that this phenomenon is prevalent across different datasets for models such as DOICR. We propose models which aim to reduce the exploitation of conditional coverage. This section concludes our background chapter covering conformal prediction and conformal training. We now turn to the next chapter of this thesis where we define and evaluate our general conformal training method for regression.

Chapter 3

A Conformal Training Framework for Regression

In this chapter, we introduce our general scheme for the conformal training of gradient-based regression models which we refer to as Conformal Training for Regression (CTR). We then follow by presenting three models based on CTR with different architectures and loss functions. The first variant is an example of an approach we suggest avoiding and acts as a counterexample – Width-Minimising Conformal Training for Regression (WM-CTR). The second and third variants are Quantile Conformal Training for Regression (Q-CTR) and Adaptive-Quantile Conformal Training for Regression (AQ-CTR), both of which we demonstrate in the following chapter to yield state-of-the-art results across multiple real datasets. Before we introduce our general CTR approach, we give an overview of prior and concurrent work.

3.1 Related Work

In section we provide an overview of related work with a summary shown in Table 3.1. We begin by comparing the two baseline methods we use that do not utilise conformal training - conformalised quantile regression (see Section 2.1.4) and conformal histogram regression (see Section 2.1.5) - to our own approach.

Conformalised quantile regression (CQR) (Romano et al. (2019)) is most similar to our Q-CTR model which optimises the same loss function as CQR and hence gives prediction intervals for to the centred quantile pair $(q_{\alpha/2}, q_{1-\alpha/2})$. Simply put, we can say Q-CTR is conformally trained CQR. Conformal histogram regression (CHR) (Sesia and Romano (2021)), compared to CQR, tries to output possibly non-centred predictive intervals with valid coverage and minimal width. The difference between

Table 3.1 Comparison of related prior and concurrent work on classification (Cls.) and regression. The compared attributes are (1) fully end-to-end conformal training, (2) improved conditional coverage, (3) prediction interval adaptivity to target distribution skewness.

	Method	(1)	(2)	(3)
Cls.	ConfTr – Stutz et al. (2021)	✓	✓	N/A
	Einbinder et al. (2022)	✓	✓	N/A
Regression	CQR – Romano et al. (2019)	✗	✗	✗
	CHR – Sesia and Romano (2021)	✗	✓	✓
	SCPO – Bellotti (2020)	✓	✗	✓
	DOICR – Lei and Bellotti (2023)	✓	✗	✓
	WM-CTR	✓	✗	✓
	Q-CTR	✓	✓	✗
	AQ-CTR	✓	✓	✓

the predictions of CHR and CQR is apparent in datasets with skewness present in the target variable. This method bears most similarity to our AQ-CTR model which uses our adaptive-quantile reformulation of the interval score loss function to adjust to potentially skewed target distributions. Compared to AQ-CTR, CHR does not use conformal training, has a complex calibration procedure, and needs the base model to predict many conditional quantiles to compute its final predictive interval.

There have been several works which consider incorporating conformalisation into the training of predictive models. These methods have been shown to work both for classification and the regression tasks. We refer the reader to Section 2.3 for a general introduction to these methods. The original work[1] on incorporating conformalisation into the training of classifiers was the introduction of the *ConfTr* framework (Stutz et al. (2021)). This work showed impressive results by incorporating mini-batch calibration into the training scheme.

For calibration, in contrast to our leave-one-out based calibration (LOOC) approach (which we believe is more efficient based on our results in Section 4.3.2, Stutz et al. (2021)), split each batch into two, a predictive set and a calibration set. We refer to this approach as split calibration (SC). Furthermore, they use a differentiable sorting approach (for an example work, see Blondel et al. (2020)) to smooth quantile estimation. We in our model do not use differentiable sorting or any other smoothing approach since during preliminary tests we did not notice any significant difference when using it

[1] We note that the work by Bellotti (2021) was done concurrently. However, while their method was conformal-aware, they did not incorporate calibration into the training scheme.

– we leave a conclusive analysis as future work. Finally, we note that the main method they present optimises for the cardinality of the predictive sets. They do however provide another variant which optimises for class-conditional coverage.

For the work on classification by Einbinder et al. (2022), they take a different approach and optimise not for the cardinality of the predictions sets but maximise the uniformity of the conditional score distribution. This prevents the model from exploiting conditional coverage. However, this approach, as is the case for Bellotti (2021), does not directly perform local conformalisation during training.

One of the earliest works on integrating conformal-awareness into regression was the Surrogate Conformal Predictor Optimisation (SCPO) model by Bellotti (2020). They use a linear model and do not use calibration during training. An extension to this for the deep learning setting is the Directly Optimised Inductive Conformal Regression (DOICR) model by Lei and Bellotti (2023). This work is most like our WM-CTR model since they also include calibration in their training scheme and minimise width. They have a base model with two outputs, one for the predictive mean and one to quantify uncertainty for score normalisation similarly to Equation 2.5. Although our base model for CTR also gives two outputs by default, we believe the mean and uncertainty prediction approach to be more unstable during training due to normalisation. Also, in contrast to our LOOC approach, they use SC (like Stutz et al. (2021)) and, most importantly, they do not aim to counteract the negative effect on conditional coverage from directly minimising predictive interval width.

Finally, we note the concurrent work by Colombo (2023). While they do not have an end-to-end model as they rely on a pretrained base model, they use all scores in the mini-batch for score adjustment. This bears a slight resemblance to LOOC. In their words, this allows for the model to concurrently fit multiple target coverage levels without the need for retraining. We believe that this has a similar effect of directing the conditional score function to be uniform as in Einbinder et al. (2022), thus improving conditional validity. However, by considering any dataset with some level of skewness, one can see that optimising a single predictive interval for conformalisation simultaneously over a wide grid of coverage levels can decrease efficiency for predictions for any specific target level. In contrast to this, we use only a single target coverage level for mini-batch calibration. Additionally, they use a class of score functions similar to the approach of DOICR and their approach is not fully end-to-end since it requires training the mean predictor in advance of training the uncertainty estimator.

3.2 Conformal Training for Regression

In this section, we present our conformal training for regression (CTR) framework. The general CTR-based model we propose outputs two bound estimates similar to the quantile estimates in a normal quantile regression model. We define the base model as a function from the function class $F = \{f_\theta \mid \theta \in \Theta\}$ where Θ is a continuous space of parameters which uniquely defines $f_\theta \in F$ with $f_\theta : \mathcal{X} \to \mathbb{R}^2$. For some input $\mathbf{x}_i$ from the dataset $\mathcal{D}$ and parameters θ, we define the lower and upper bound estimate as $(\hat{y}_i^{\text{lo}}, \hat{y}_i^{\text{hi}}) \equiv f_\theta(x_i)$. A simple definition of Θ, the search-space of parameters, could be the weight-space of a neural network.

During training, compared to standard conformalised quantile regression, we perform mini-batch calibration to adjust predictions within the model forward pass, and allow for any loss function to be used. We also perform local conformalisation after training on a split calibration set $\mathcal{D}_{\text{cal}}$ to get the marginal coverage guarantee on the test data $\mathcal{D}_{\text{test}}$. During inference on the test dataset, mini-batch calibration is not used and the predictions are adjusted using the appropriately calculated quantile of the calibration scores.

Both mini-batch calibration and post-training calibration are defined to use the same score function. We represent it as $s(\hat{y}^{\text{lo}}, \hat{y}^{\text{hi}}, y)$, where y is the true target and $\hat{y}^{\text{lo}}, \hat{y}^{\text{hi}}$ are defined as above. CTR allows different score functions to be used, which possibly have more parameters. By default, the score function we use is the conformalised quantile regression score function from Equation 2.7 which we define here as

$$s(\hat{y}^{\text{lo}}, \hat{y}^{\text{hi}}, y) = \max(\hat{y}^{\text{lo}} - y, y - \hat{y}^{\text{hi}}). \tag{3.1}$$

We now describe mini-batch calibration in more detail. The forward pass of CTR during training is illustrated in Figure 3.1. The whole of the CTR method is provided in Algorithm 2. We note that we use the default score function (Equation 3.1) in the algorithm. For any other type of suitable score function, the correction using the respective score quantiles may have to be adapted accordingly.

Mini-Batch Calibration

Mini-batch calibration is the process whereby for each $\mathbf{x}_i$ in a mini-batch (with indices defined to be from $B \subset \mathcal{I}_{\text{train}}$), bound estimates $(\hat{y}_i^{\text{lo}}, \hat{y}_i^{\text{hi}}) = f_\theta(\mathbf{x}_i)$ are transformed into *mini-batch conformalised* estimates $\bar{y}_i^{\text{lo}}, \bar{y}_i^{\text{hi}}$. It consists of two main steps, leave-one-out score quantile calculation and score adjustment.

Algorithm 2: Conformal Training for Regression

```
Input: $\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{cal}}$, $\alpha$, $\mathbf{x}$ (test input), $f_\theta \in F$ with $\theta$ (randomly initialised
        parameters), $b$ (mini-batch size), loss (loss function)
Result: $\hat{C}$
for $t = 1, ..., N_{epochs}$ do
    Randomly partition $\mathcal{I}_{\text{train}}$ (the indexing set of $\mathcal{D}_{\text{train}}$) into family of
      mini-batch indices $\mathcal{B}$ with each set of indices $B \in \mathcal{B}$ of size $b$;
    for $B \in \mathcal{B}$ do
        $\Big((\hat{y}_i^{\text{lo}}, \hat{y}_i^{\text{hi}})\Big)_{i \in B} \leftarrow \Big(f_\theta(\mathbf{x}_i)\Big)_{i \in B}$ ;            /* Evaluate $f_\theta$ on batch */
        $(s_i)_{i \in B} \leftarrow \Big(\max(\hat{y}^{\text{lo}} - y, y - \hat{y}^{\text{hi}})\Big)_{i \in B}$ ;        /* Calculate scores */
        Set $r_1$ as $\lceil |B|(1-\alpha) \rceil$-th smallest in $(s_i)_{i \in B}$;
        Set $r_2$ as $(\lceil |B|(1-\alpha) \rceil + 1)$-th smallest in $(s_i)_{i \in B}$;
        Initialise empty list $\Big(r_{i,\alpha}^{(t)}\Big)_{i \in B}$;
        for $i \in B$ do
            /* Calculate leave-one-out score quantile:  */
            if $s_i > r_1$ then
                $r_{i,\alpha}^{(t)} \leftarrow r_1$;
            else
                $r_{i,\alpha}^{(t)} \leftarrow r_2$;
            end
        end
        /* Finish mini-batch calibration and calculate loss:  */
        $\Big((\bar{y}_i^{\text{lo}}, \bar{y}_i^{\text{hi}})\Big)_{i \in B} \leftarrow \Big((\hat{y}_i^{\text{lo}} - r_{i,\alpha}^{(t)}, \hat{y}_i^{\text{hi}} + r_{i,\alpha}^{(t)})\Big)_{i \in B}$
         $\mathcal{L}_B = \texttt{loss}\Big(\Big((\bar{y}_i^{\text{lo}}, \bar{y}_i^{\text{hi}})\Big)_{i \in B}, (y_i)_{i \in B}\Big)$;
        Update the model parameters $\theta$ using the gradients of $\mathcal{L}_B$
    end
    $\{(\hat{y}_i^{\text{lo}}, \hat{y}_i^{\text{hi}})\}_{i \in \mathcal{D}_{\text{cal}}} \leftarrow \{f_\theta(\mathbf{x}_i)\}_{i \in \mathcal{D}_{\text{cal}}}$;
    $S_{\text{cal}} \leftarrow \{\max(\hat{y}_i^{\text{lo}} - y_i, y_i - \hat{y}_i^{\text{hi}})\}_{i \in \mathcal{D}_{\text{cal}}}$;
    Calculate the $\lceil (|\mathcal{D}_{\text{cal}}| + 1)(1-\alpha) \rceil$-th smallest score $r_{\text{cal}}$ of the scores $S_{\text{cal}}$;
    $(\hat{y}^{\text{lo}}, \hat{y}^{\text{hi}}) \leftarrow f_\theta(\mathbf{x})$;
    $\hat{C} \leftarrow [\hat{y}^{\text{lo}} - r_{\text{cal}}, \hat{y}^{\text{hi}} + r_{\text{cal}}]$;
end
```

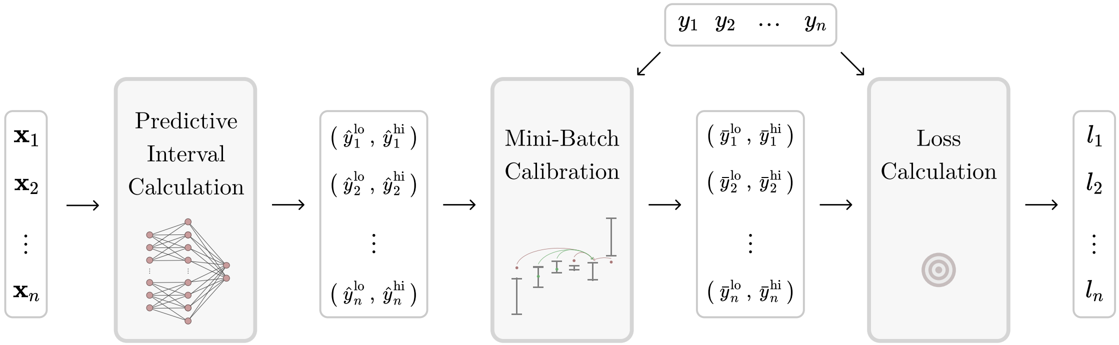
$\mathbf{x}_1$
$\mathbf{x}_2$
$\vdots$
$\mathbf{x}_n$
Predictive Interval Calculation
$(\hat{y}_1^{\text{lo}}, \hat{y}_1^{\text{hi}})$
$(\hat{y}_2^{\text{lo}}, \hat{y}_2^{\text{hi}})$
$\vdots$
$(\hat{y}_n^{\text{lo}}, \hat{y}_n^{\text{hi}})$
Mini-Batch Calibration
y_1 y_2 $\cdots$ y_n
$(\bar{y}_1^{\text{lo}}, \bar{y}_1^{\text{hi}})$
$(\bar{y}_2^{\text{lo}}, \bar{y}_2^{\text{hi}})$
$\vdots$
$(\bar{y}_n^{\text{lo}}, \bar{y}_n^{\text{hi}})$
Loss Calculation
l_1
l_2
$\vdots$
l_n

Fig. 3.1 Illustration of the forward pass of a mini-batch of the general CTR model during training. This consists of a mini-batch of inputs being input to a neural network to yield two bound estimates. These estimates are then locally adjusted through mini-batch calibration. The loss of choice is calculated on the calibrated estimates.

For some $i \in B$, basic score quantile calculation is done simply by calculating the target $\lceil |B|(1-\alpha) \rceil / (|B|-1)$ quantile $r_{i,\alpha}$ over the set of scores $S_{i,B} \equiv \{s(\hat{y}_j^{\text{lo}}, \hat{y}_j^{\text{hi}}, y_j) \mid j \in B,\ j \neq i\}$. This constitutes the *leave-one-out* score quantile calculation. When using the default score function (Equation 3.1), score adjustment is done using the quantile $r_{i,\alpha}$, yielding

$$(\bar{y}_i^{\text{lo}}, \bar{y}_i^{\text{hi}}) = (\hat{y}_i^{\text{lo}} - r_{i,\alpha}, \hat{y}_i^{\text{hi}} + r_{i,\alpha}).$$

Instead of directly using the base model's outputs, these locally calibrated predictions $\bar{y}_i^{\text{lo}}$, and $\bar{y}_i^{\text{hi}}$ are then used for calculating the loss on each mini-batch element. We stress that this form of the correction is the default case when Equation 3.1 is used as the score function. For any other score function, the correction can be derived by considering the predictive set $\hat{C}$ from Algorithm 1.

Conformalisation

After training, the base model has distilled information about the conformalisation procedure, including the form of the score function. We then perform conformalisation of the predictions of the base model using $\mathcal{D}_{\text{cal}}$. It is important to note that during the post-training conformalisation process 1) the same score function is used as during training and 2) that we do not perform any kind of mini-batch calibration. We now turn to defining three novel models which use CTR. We will show performance of these models on multiple experiments in the following chapter.

3.3 Width-Minimising CTR: A Counterexample

The first example of a model based on our CTR framework is Width-Minimising CTR (WM-CTR). This serves as a counterexample to the two main models of this thesis which are introduced directly after. For WM-CTR, we employ interval width as its loss within the CTR framework. We aim to show with this approach of directly optimising conformalised mean-interval width has adverse results specifically regarding conditional validity and the resulting model is not advisable for use in practise due to its unreliable nature.

Definition 3.3.1 (WM-CTR)**.** As a CTR-based model, Width-Minimising CTR uses the loss

$$l(\bar{y}_i^{\mathrm{lo}}, \bar{y}_i^{\mathrm{hi}}, y_i) = \bar{y}_i^{\mathrm{hi}} - \bar{y}_i^{\mathrm{lo}},$$

where $\bar{y}_i^{\mathrm{lo}}$ and $\bar{y}_i^{\mathrm{hi}}$ are mini-batch calibrated estimates for input $\mathbf{x}_i$ with target y_i. The score function is kept as the default (Equation 2.7).

This method can be considered in essence similar to DOICR (Lei and Bellotti (2023)) as it also outputs two predictions, performs mini-batch calibration, and has been trained to minimise mean interval width. However, the base model for WM-CTR outputs lower and upper bound estimates whilst DOICR outputs a mean and normaliser prediction. The latter in our experiments was more difficult to train due to numerical issues to do with the normaliser. Another difference between the two models we note is in the mini-batch calibration approach. WM-CTR uses leave-one-out calibration (LOOC) while DOICR uses split calibration (SC). We believe LOOC utilises the mini-batch more efficiently and we show preliminary results for this in Section 4.3.2. We note that for most intents and purposes, the main conclusions we make about WM-CTR also apply to DOICR and any other similar conformal prediction methods that solely optimise for mean interval width.

3.4 Quantile CTR

The first central model of this thesis that we propose is Quantile Conformal Training for Regression (Q-CTR). Using the CTR framework, it applies the pinball loss in the same way as for CQR. In other words, referring to the illustration in Figure 3.1, the loss calculation is simply the sum of the pinball losses for the lower and upper bound estimates. We suggest that this model can be regarded as an extension to the basic CQR model making its predictions more reliable and trustworthy.

Definition 3.4.1 (Q-CTR)**.** As a CTR-based model, for a target miscoverage rate α, Quantile CTR uses the loss

$$l(\bar{y}_i^{\text{lo}}, \bar{y}_i^{\text{hi}}, y_i) = l_{\alpha/2}(y, \bar{y}_i^{\text{lo}}) + l_{1-\alpha/2}(y, \bar{y}_i^{\text{hi}}),$$

where $\bar{y}_i^{\text{lo}}$ and $\bar{y}_i^{\text{hi}}$ are mini-batch calibrated estimates for input $\mathbf{x}_i$ with target y_i and $l_{\alpha/2}$ and $l_{1-\alpha/2}$ are pinball losses (Equation 2.6). The score function is kept as the default (Equation 2.7).

The inherent value we believe conformal training on the CQR model introduces is simple. It allows the base model to be *aware* of the conformalisation procedure which follows fitting, thereby allowing it to output predictions that have an improved quantile loss after conformalisation. We note that there are several issues with conformalisation of a quantile regression model using the most basic score function (Equation 2.7).

For one, since conformalisation using the quantile score function does a constant correction to the predicted intervals, we might end up with an uneven calibration that is biased to certain input space regions. Furthermore, shifting intervals equally at the lower and upper bounds may make for a suboptimal correction. While asymmetric conformalisation (see Romano et al. (2019)) and model variants such as CQR-m and CQR-r (for more details, see Sesia and Candès (2020)) address some of these concerns, the results can vary from dataset to dataset with, to our knowledge, there being no clear winner.

Once the model is trained to be aware of the constant-width conformalisation procedure which we employ in CTR, it can correct its base predictions to accommodate for any shortcomings of the conformalisation procedure subject to the dataset that it is being trained on. For this reason, we believe this approach allows for more reliable and robust predictions compared to CQR.

3.5 Adaptive-Quantile CTR

As the second central model based on CTR, we propose Adaptive-Quantile Conformal Training for Regression (AQ-CTR). While dataset-dependent, there are a few potential downsides for optimising using the pinball loss. For one, it requires fixing quantiles to specific levels – say 0.05 and 0.95 for a target coverage of $\alpha = 0.9$. This means that if skewness is present in the conditional data distribution, the predictive intervals may be unnecessarily wide. If interval width is of importance, this may become a problem. Another issue is that the pinball loss for quantile regression is prone to driving sufficiently expressive models to overfit (for more see Chung et al. (2021)) – this may have an effect even on a conformal-aware method that utilises this loss.

The loss we use for AQ-CTR is the interval score function from Gneiting and Raftery (2007) which we have adjusted to allow for the model to provide adaptive quantile estimation. The original interval score function is given as

$$l_\alpha(\bar{y}^{\text{lo}}, \bar{y}^{\text{hi}}, y) = (\bar{y}^{\text{hi}} - \bar{y}^{\text{lo}}) + \frac{2}{\alpha}\Big(\mathbb{1}_{y<\bar{y}^{\text{lo}}}(\bar{y}^{\text{lo}} - y) + \mathbb{1}_{y>\bar{y}^{\text{hi}}}(y - \bar{y}^{\text{hi}})\Big), \tag{3.2}$$

where $\bar{y}^{\text{lo}}, \bar{y}^{\text{hi}}$ are lower and upper quantile predictions and y is the target. Introduced specifically as a loss by Chung et al. (2021), the authors prove that this loss function is minimised, on average, when the predictions are exactly the true quantiles $q_{\alpha/2}$ and $q_{1-\alpha/2}$. We note that this means the interval width component of the loss is perfectly balanced with the component which targets miscoverage.

This motivates an adjustment to Equation 3.2 which directs the model to predict intervals with correct coverage that are minimal whilst having freedom to choose any pair of target quantiles with valid coverage between them. The resulting loss we call the *adaptive-quantile interval score loss* is given by

$$l_\alpha(\bar{y}^{\text{lo}}, \bar{y}^{\text{hi}}, \beta, y) = (\bar{y}^{\text{hi}} - \bar{y}^{\text{lo}}) + \frac{2}{\alpha - \beta}\mathbb{1}_{y<\bar{y}^{\text{lo}}}(\bar{y}^{\text{lo}} - y) + \frac{2}{\alpha + \beta}\mathbb{1}_{y>\bar{y}^{\text{hi}}}(y - \bar{y}^{\text{hi}}) \tag{3.3}$$

where β is a quantile adjustment term with $|\beta| < \alpha$ which indicates that $\bar{y}^{\text{lo}}$ is an estimation of $q_{(\alpha-\beta)/2}$ and $\bar{y}^{\text{hi}}$ is an estimation of $q_{1-(\alpha+\beta)/2}$ – still targeting a coverage of $1 - \alpha$ between the two. This allows for the model to adapt to the skewness of the data and produce tighter predictions thanks to the additional degree of freedom. In our implementation, we make the value of β dependent on the input and learnable by adding a third output to the base neural network model, thus yielding the *adaptive quantile* predictions

$$(\hat{y}_i^{\text{lo}}, \hat{y}_i^{\text{hi}}, \beta_i) \equiv f_\theta(x_i).$$

We employ β_i solely during training for the loss calculation. As an additional detail, we restrict the output of β to $-\alpha + \epsilon \leq \beta \leq \alpha - \epsilon$ using a suitably scaled hyperbolic tangent activation function, where we set $\epsilon = 10^{-3}$ to prevent numerical instability during training. For clarity, we formally state the definition of AQ-CTR next.

Definition 3.5.1 (AQ-CTR)**.** As a CTR-based model, for a target miscoverage rate α, Adaptive-Quantile CTR applies the adaptive-quantile interval score loss (Equation 3.3) to the CTR framework and adjusts the base model for an input $\mathbf{x}$ to yield an additional third output β, the quantile adjustment factor. The score function is kept as the default (Equation 2.7).

Even though mean interval width is a component of this loss, this loss for any specified β is still at a minimum when $\bar{y}^{\mathrm{lo}} = q_{(\alpha-\beta)/2}$ and $\bar{y}^{\mathrm{hi}} = q_{1-(\alpha+\beta)/2}$. Hence this loss is *faithful* to the target level of coverage which is not the case for WM-CTR that uses the interval width as its only loss component. We state this as a theorem next and provide a proof in Appendix A.

Theorem 3.5.1. *The adaptive-quantile interval score loss (Equation 3.3) achieves a minimum whenever* $\bar{y}^{lo} = q_{(\alpha-\beta)/2}$ *and* $\bar{y}^{hi} = q_{1-(\alpha+\beta)/2}$ *and thus is faithful to a coverage level of* $1-\alpha$*.*

Finally, we note that this adaptive-quantile interval score loss could be used as a drop-in replacement for the loss function used in CQR to yield improved adaptivity to skewness. We leave analysing this model variant as a possible future research direction. We now turn to the experiment section and present a thorough comparison of the proposed models and related works using a synthetic dataset and numerous real datasets.

Chapter 4

Empirical Results on Synthetic and Real Datasets

In this chapter we provide results of the three proposed models on a synthetic dataset and multiple real datasets. As a comparison, we also provide results for CQR (Romano et al. (2019)), CHR (Sesia and Romano (2021)), both state-of-the-art conformal prediction models based on quantile regression. As a direct comparison to other conformal training methods, we give results for the DOICR method (Lei and Bellotti (2023)) which is closest to our WM-CTR model[1]. At the end of the chapter, we provide empirical evidence supporting our claim that our approach for mini-batch calibration is more memory- and data-efficient than previously proposed approaches. We finish this chapter with a discussion. We now give a short preview of the results.

In this chapter, we demonstrate the effectiveness of using conformal training. We show that the novel Q-CTR and AQ-CTR methods yield more reliable estimates than conformalised quantile regression (CQR); are significantly more robust than conformal training methods which solely optimise for interval width; and also have similar or improved results compared to conformal histogram regression (CHR) (Sesia and Romano (2021)). Also, thanks to its adaptivity to target distribution skewness, we show that AQ-CTR can provide tighter predictive intervals without sacrificing coverage in areas of high uncertainty – in many cases making it more reliable than the skew-adaptive CHR.

Furthermore, with our WM-CTR model, we demonstrate the pitfalls of directly optimising for average predictive set size. We show on synthetic and real datasets,

[1] We note we include results for DOICR on the synthetic dataset only – we ran into numerical instabilities due to the form of the score function during training of DOICR on the real datasets. The authors of the method also mention this difficulty in their work.

that a model fit using conformal training with a loss function that is solely mean interval width can be expected to have worse conditional coverage than the classical conformalised quantile regression case. Moreover, we show that this approach will provide misleading estimations of uncertainty, e.g., giving tight predictive intervals in regions of space where uncertainty should instead be high. Before we present our results, we first describe our experimental design. We include details on models, data preprocessing, the training scheme, and the metrics we use for assessing model performance.

4.1 Experimental design

Data preprocessing

We use the same data loading and preprocessing approach as the authors of conformalised quantile regression (Romano et al. (2019)). For each dataset, this includes randomly (for a given random generator seed) splitting $\mathcal{D}$ into $\mathcal{D}_{\text{train}}$ (40%), $\mathcal{D}_{\text{cal}}$ (40%) and $\mathcal{D}_{\text{test}}$ (20%). Further, we perform standardisation of inputs in all data splits given the inputs in $\mathcal{D}_{\text{train}}$. We also scale the targets of all splits by dividing them by the mean of the targets $\mathcal{D}_{\text{train}}$.

Hyperparameters and training scheme

For all experiments we set the miscoverage rate to $\alpha = 0.1$, or, in other words, we target a coverage of 90%. We use a neural network with two hidden 64 unit layers with ReLU activation functions and dropout applied to the outputs of both layers with 0.1 probability. The size of the output layer is 2 in all cases except for 3 in the case of AQ-CTR and 99 for CHR. We use a batch size of 128 and the Adam optimiser with a learning rate of $5 \cdot 10^{-4}$ and a weight decay of 10^{-6}. Also, since $\hat{y}^{\text{lo}}$ and $\hat{y}^{\text{hi}}$ can cross, a notable issue when training bound predictions with neural networks, we add a regularising loss of the form $10^2 \cdot \mathbb{1}_{\hat{y}^{\text{lo}} > \hat{y}^{\text{hi}}} \hat{y}^{\text{lo}} - \hat{y}^{\text{hi}}$ for the CTR models – while this does not improve accuracy, it helps avoid numerical issues during training.

Furthermore, we perform cross-validation before training to determine a suitable number of epochs for training. We randomly split $\mathcal{D}_{\text{train}}$ into a separate training and validation set, with a ratio of 4:1. Out of 2000 epochs, measure the best number of epochs in terms of a validation set metric which we then use for the real training run. For the metric, we use coverage and mean interval width for CQR (as the authors of the method specify), and the actual loss function which include mini-batch calibration for DOICR and all CTR-based models. For CHR, we use a single cross-validation round as in the original work, 1 000 bins for the histogram, and randomisation during prediction (for more, see Sesia and Romano (2021)).

Measuring Conditional Coverage

For the synthetic dataset, we can simply measure conditional coverage through analytically deriving oracle quantiles from the data generating process. However, for real datasets, measuring conditional coverage is more difficult, since we do not know the conditional probability density of the response variable given the input. For this reason we turn to binning-based approaches to provide estimates of conditional coverage. Simply put, we split the test data $\mathcal{D}_{\text{test}}$ into k bins and calculate coverage for each bin given a model.

More specifically, by defining the indexing set $\mathcal{I}_{\text{test}} \subset \mathbb{N}$ of $\mathcal{D}_{\text{test}}$ such that $\{(\mathbf{x}_i, y_i)\}_{i \in \mathcal{I}_{\text{test}}} = \mathcal{D}_{\text{test}}$, we represent the bins as a set $\mathcal{G}$ of index sets such that $\bigcup_{G \in \mathcal{G}} G = \mathcal{I}_{\text{test}}$, $G_i \cap G_j = \emptyset$ for any distinct $G_i, G_j \in G$, and $|\mathcal{G}| = k$. We define the empirical coverage function c as for an input set of bin $G \in \mathcal{G}$ as

$$c(G) = \frac{1}{|G|} \sum_{i \in G} \mathbb{1}_{\hat{y}^{\text{lo}} \leq y_i \leq \hat{y}^{\text{hi}}},$$

where $\hat{y}_i^{\text{lo}}$ and $\hat{y}_i^{\text{hi}}$ are the respective lower and upper bound predictions of the model given $\mathbf{x}_i$. This binning-based approach allows conditional coverage to be visualised in an approximate way, but also allows calculating measures of conditional coverage.

For our metrics, we use mean absolute deviation (m_{MAD}) from the target coverage (e.g., 0.9) over all bins. We also use the worst slab coverage (m_{WSC}) metric (Cauchois et al. (2021)) with gives the coverage level of the bin which covers the least. To be precise, m_{MAD} is defined as

$$m_{\text{MAD}} = \frac{1}{|\mathcal{G}|} \sum_{G \in \mathcal{G}} \Big| (1 - \alpha) - c(G) \Big|$$

with m_{WSC} given simply by $\min_{G \in \mathcal{G}} c(G)$. We report both values in percentages and provide means and standard deviations of the values across all seeded runs.

We use two different tactics for partitioning the data into $\mathcal{G}$, one is based on predictive interval width and the other on clustering the input space using K-means. The interval-width based partitioning takes the predictions of the baseline CQR method and sorts them in ascending order. Then it performs an equal split of the sorted sequence of respective data points into k bins. This partition is then used for all models. This has similarities with the work by Angelopoulos et al. (2020) where prediction sets are partitioned by their size. We use this binning method only for visualisation purposes (see the two top rows of Figure 4.2). The main method we use for binning is the clustering of input data points. We use a K-means clustering approach and partition the test inputs into k bins. All results given in the tables and the lower halves of Figure 4.2 and the figures from Appendix B are based on this approach.

To limit any bias, we perform our analysis over several values of k. Since our datasets are of different sizes we split our datasets into two groups, *small* and *large*, both of which has different set of values for k. This is such that we do not perform clustering into too many bins which would render the m_{MAD} and m_{WSC} metrics uninformative. For the smaller datasets (`concrete`, `community`, `star`), we use the number of bins $k \in \{2, 3, ..., 10\}$. For the larger datasets (see below for more details), we use the number of bins $k \in \{5, 10, 15, ..., 150\}$.

Overview of Datasets

We provide results across multiple real tabular datasets. This is the same collection of datasets used in the original CQR work. The datasets consist of the following: the concrete compressive strength dataset `concrete` (UC Irvine Machine Learning Repository (2007)); the communities and crime dataset `community` (UC Irvine Machine Learning Repository (2009)); the student teacher achievement ratio dataset `star` (Word et al. (1990)); the bike sharing dataset `bike` (UC Irvine Machine Learning Repository (2013a)); medical expenditure panel survey data `meps_19` and `meps_20` (Agency for Healthcare Research and Quality (2017)), and `meps_21` (Agency for Healthcare Research and Quality (2018)); the physicochemical properties of protein tertiary structure dataset `bio` (UC Irvine Machine Learning Repository (2013b)); the blog feedback dataset `blog_data` (UC Irvine Machine Learning Repository (2014)); and the Facebook comment volume datasets `facebook_1` and `facebook_2` (Singh et al. (2015); UC Irvine Machine Learning Repository (2016)). We provide an overview of the dataset targets and sizes in table 4.1.

4.2 Experiment on Synthetic Data

We begin with a synthetic dataset example to demonstrate the improved conditional validity of the predictions of Q-CTR and AQ-CTR. The example further illustrates

Table 4.1 Overview of real tabular regression datasets used. Includes a description of the target scalar variable and the size of the dataset after preprocessing.

Dataset name	Target description	Size
`concrete`	Concrete compressive strength	1 030
`community`	Violent crimes per capita	1 994
`star`	Student sum of reading and mathematics scores	2 161
`bike`	Number of active rental bike users at given hour	10 886
`meps_19`	Patient total expenditure	15 785
`meps_20`	Patient total expenditure	17 541
`meps_21`	Patient total expenditure	15 656
`bio`	Protein residue size	45 730
`blog_data`	Blog post comment volume	52 397
`facebook_1`	Post comment volume	40 948
`facebook_2`	Post comment volume	81 311

that directly optimising predictive efficiency yields unreliable and even misleading predictions in the case of WM-CTR and DOICR.

A datapoint in the heteroskedastic synthetic dataset we use is generated as follows. First, we sample a latent z from the uniform distribution as $z \sim U([0, 1])$. We then define the respective input $\mathbf{x} \in \mathbb{R}^4$ – which is arbitrary as long as it contains sufficient information about z – to the model as

$$\mathbf{x} = \begin{pmatrix} \sin 2z \\ z^2 \\ \sin 3z \\ z^3 \end{pmatrix}.$$

Finally, we sample the target $y \in \mathbb{R}$ with $y \sim \mathcal{N}\Big(0, (z + \frac{1}{2})^8\Big)$. Thanks to the type of noise we use, we can compare the predictions to an oracle and exactly calculate the conditional coverage level given a prediction for a certain $\mathbf{x}$. We generate 6000 data points for $\mathcal{D}$ and perform the split with the same ratio as described in the previous sections.

We show results in Figure 4.1. We highlight that the predictions of Q-CTR, AQ-CTR are closest to the oracle's predictions. CHR follows directly after. For CQR, we can see undercoverage in the high-noise region and slight overcoverage elsewhere. This could be attributable to the fact that the score function makes a constant correction to the predictions which could be too large for the low noise area and too small for the

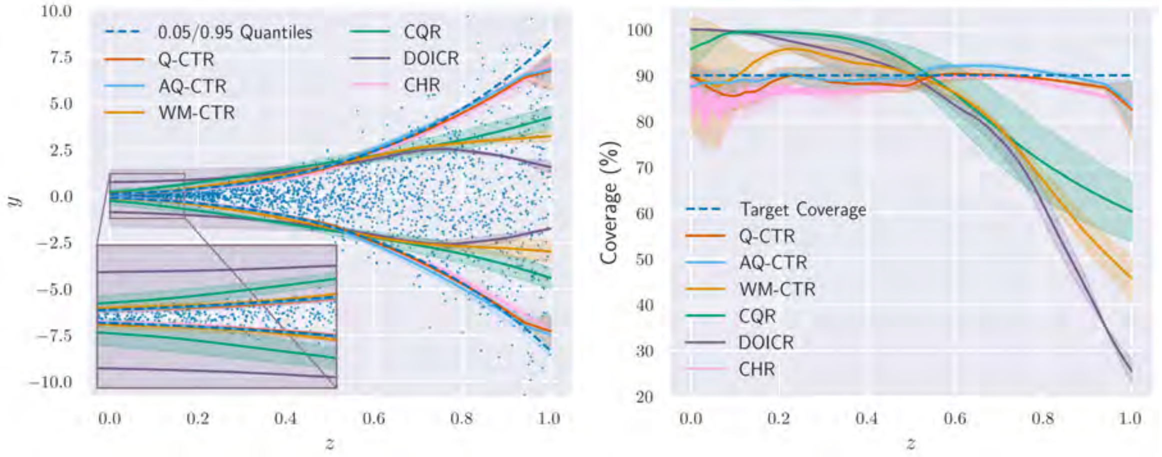

Fig. 4.1 Results for CQR, WM-CTR, DOICR, CHR, and CAQR with $\alpha = 0.1$ on a synthetic dataset consisting of targets sampled from zero-centred Gaussian noise with varying scale depending on latent variable z. Left: Lower and upper bound predictions of y given z. Optimal predictions for bounds (0.05/0.95 quantiles) are given in blue dashed lines. Samples from $p(y|z)$ are shown scattered in blue. Right: actual conditional coverage level given z for all models. Target 90% coverage is denoted with a blue dashed line.

high noise area. We also note that cross-validation could have an impact on the results for CQR, since the approach defined by the authors optimises for mean interval width.

Finally, WM-CTR and DOICR both significantly undercover in the high-noise region, with DOICR even more markedly so. Whilst it is true, the area between the lower and upper bound prediction in the left-hand-side of Figure 4.1 is smallest for WM-CTR and DOICR, this highlights the misleading nature of models which solely optimise for predictive efficiency. We reiterate a difference between the two models: whilst both optimise for mean interval width, WM-CTR uses a quantile prediction base model and DOICR estimates the conditional target mean and uncertainty (conformal normaliser)[2]. Since it does not impact the calibration procedure, both models are incentivised to inflate the largest tail end of the scores through minimising the respective predictive interval widths – note the drop in interval width for DOICR and the reduction of growth in interval width for WM-CTR from around $z = 0.8$ onwards. Noting their difference, we believe the two models' predictions vary slightly since, from an optimisation standpoint, it could be easier to perform this inflation by decreasing the values of the conformal normaliser (in the case of DOICR) than to bring two bound predictions close to one-another (in the case of WM-CTR).

[2] The other difference between DOICR and WM-CTR is in the mini-batch calibration. However, we believe it has no significant impact in this synthetic case. For more, see Section 4.3.2.

Table 4.2 Mean absolute deviation (%) of bin coverage from target of 90%. Averaged over 30 seeded runs with means and standard deviations reported. Lowest is best and marked in bold. Partitions into bins calculated by K-means on the input space. Multiple values for k for were used partitioning with $k \in \{2, ..., 10\}$ for the small datasets `concrete`, `community`, and `star` and $k \in \{5, 10, 15, ..., 150\}$ otherwise.

Dataset	Model				
	CQR	**WM-CTR**	CHR	**Q-CTR**	**AQ-CTR**
`concrete`	5.21 ±1.64	5.83 ±1.82	5.00 ±1.51	**4.90 ±1.04**	5.44 ±1.45
`community`	4.86 ±1.07	5.39 ±1.18	**4.06 ±1.24**	4.62 ±1.23	4.69 ±1.21
`star`	3.05 ±0.72	3.07 ±0.65	3.09 ±0.80	3.04 ±0.85	**2.96 ±0.69**
`bike`	6.98 ±0.60	6.79 ±0.50	7.04 ±0.74	6.29 ±0.50	**6.08 ±0.52**
`meps_19`	8.99 ±2.19	10.99 ±1.26	**6.21 ±0.61**	7.20 ±1.23	7.41 ±0.95
`meps_20`	8.23 ±1.78	10.53 ±1.19	**5.55 ±0.50**	6.55 ±0.97	6.95 ±0.92
`meps_21`	8.16 ±1.31	11.13 ±1.38	**5.99 ±0.51**	6.89 ±0.93	7.35 ±1.04
`bio`	5.55 ±0.90	6.22 ±1.11	5.47 ±0.84	5.31 ±1.24	**5.22 ±0.93**
`blog_data`	14.81 ±1.61	15.21 ±2.74	11.34 ±2.29	**9.96 ±2.41**	10.99 ±1.83
`facebook_1`	16.02 ±2.19	20.23 ±2.53	10.51 ±2.34	**8.04 ±1.45**	8.08 ±1.98
`facebook_2`	13.03 ±1.87	21.43 ±2.71	9.93 ±2.25	7.75 ±1.68	**7.10 ±1.74**

4.3 Experiments on Real Data

4.3.1 General Performance

We begin by showing extended results for three different real datasets in Figure 4.2. For reference, we also provide the extended figures in Appendix B for all 11 datasets. We provide summary statistics over different bin sizes for mean absolute deviation (MAD) of bin coverage and worst slab coverage (WSC) in Tables 4.2 and 4.3 respectively. For clarity, we note that the values for both tables for each dataset were averaged over multiple values of k with partitioning done on the test set using the K-means algorithm. We now describe our results in more detail.

Coverage

As can be seen in Figure 4.2, Q-CTR, AQ-CTR and CHR are consistently the top performers for the three datasets in terms of their coverage profile. Namely, they have a reduced drop in coverage levels for bins where prediction widths are the largest. While the trend of coverage decreasing as widths become larger remains, we also see that these models have less overcoverage in low prediction width regions compared to WM-CTR and CQR. CHR was the best performer for all `meps` datasets. However, for the other

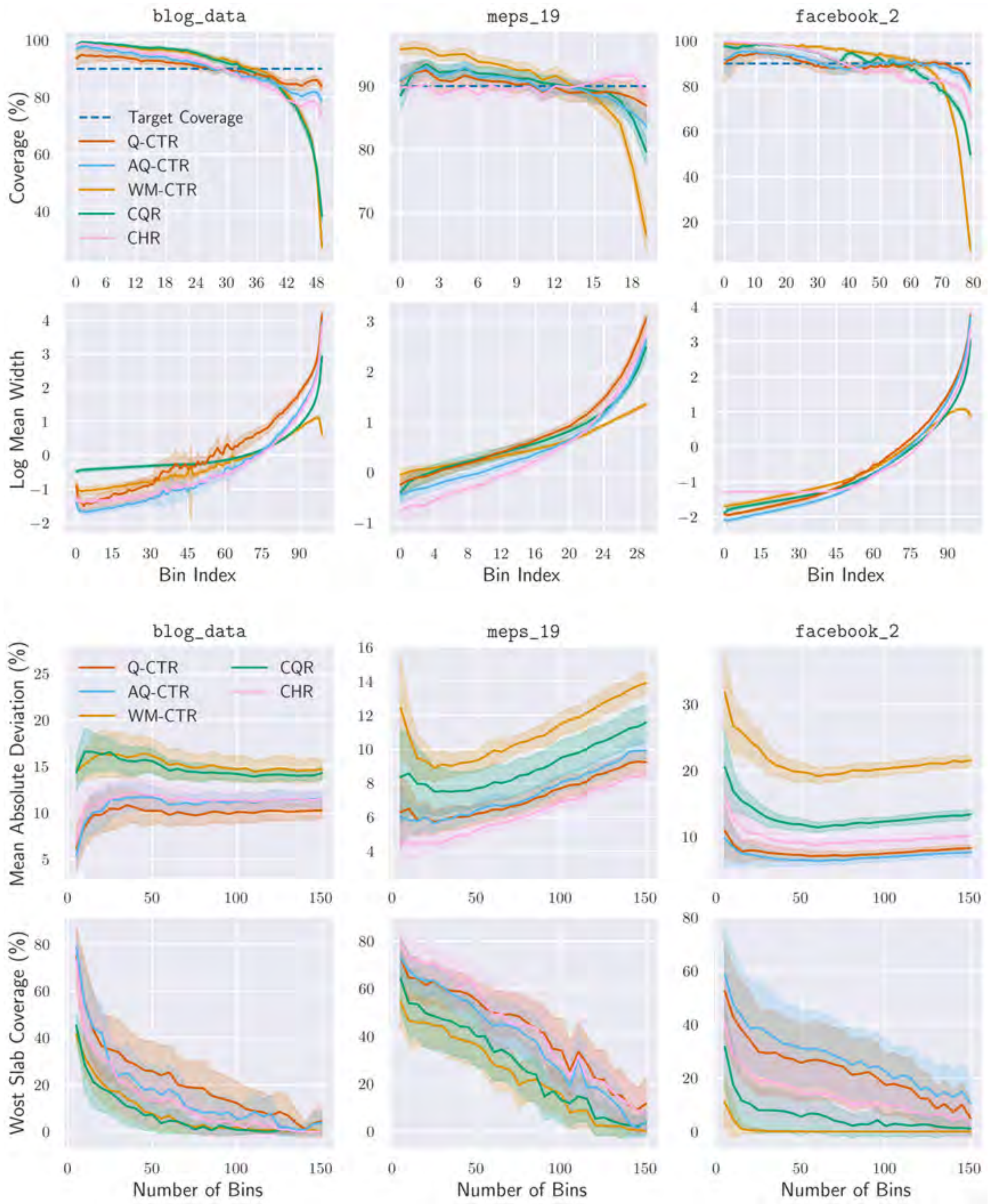

Fig. 4.2 Various metrics measuring conditional coverage and predictive interval width for blog_data, meps_19, and facebook_2 over 30 seeded runs with means and half standard deviations reported. Upper half: coverage and log mean width for bins chosen by the interval width-based method as described in Section 4.1. The respective number of bins was chosen relative to dataset size with 50 for blog_data, 20 for meps_19, and 80 for facebook_2. Lower half: mean absolute deviation and worst slab coverage across different values of k using K-means clustering.

Table 4.3 Worst slab (bin) coverage (%). Highest is best and marked in bold. Averaged over 30 seeded runs with means and standard deviations reported. Bins are calculated by K-means over the input space with k chosen as in Figure 4.2. Values in red mark significant (<10%) undercoverage.

Dataset	Model				
	CQR	**WM-CTR**	CHR	**Q-CTR**	**AQ-CTR**
`concrete`	79.34 ±7.20	76.49 ±7.85	80.55 ±6.17	**81.76 ±4.33**	79.27 ±7.88
`community`	**82.33 ±4.34**	80.70 ±3.78	82.07 ±4.88	81.66 ±4.40	81.56 ±4.68
`star`	85.28 ±2.42	84.59 ±2.51	84.81 ±2.67	84.75 ±2.76	**85.29 ±2.55**
`bike`	52.23 ±7.07	54.78 ±8.53	50.81 ±7.42	57.13 ±7.66	**59.54 ±8.00**
`meps_19`	27.29 ±16.06	23.27 ±10.46	**42.94 ±12.73**	41.92 ±13.70	37.71 ±9.18
`meps_20`	36.89 ±10.54	24.92 ±7.87	**51.75 ±10.34**	47.71 ±13.53	41.87 ±9.82
`meps_21`	35.76 ±8.67	21.39 ±9.70	**47.52 ±9.49**	45.51 ±13.15	38.81 ±11.23
`bio`	28.50 ±18.02	25.43 ±19.31	30.92 ±19.01	**41.40 ±25.09**	35.40 ±23.09
`blog_data`	6.87 ±5.54	7.78 ±5.28	12.98 ±10.82	**21.07 ±16.16**	16.73 ±11.60
`facebook_1`	5.48 ±6.14	1.48 ±2.08	10.88 ±11.79	18.35 ±14.67	**19.42 ±18.33**
`facebook_2`	5.85 ±10.71	0.60 ±1.26	13.48 ±19.15	22.82 ±23.56	**27.43 ±24.43**

datasets, the coverage curves for Q-CTR and AQ-CTR were better or matched CHR. Also we bring attention to the fact that Q-CTR had slightly improved undercoverage than AQ-CTR in the high-width regions. We believe this could be since AQ-CTR, with its additional degree of freedom, is more prone to overfitting. Finally, similarly to the synthetic data experiment, we see that WM-CTR had significant undercoverage in high-width regions.

Mean Interval Length

We report several findings on mean interval width from Figure 4.2 and the figures in Appendix B. First, Q-CTR has wider predictive intervals on average than AQ-CTR. Also, both models have an in increased width in the regions of highest predictive uncertainty – its effect is clearly seen by the improved coverage compared to other models.

We note the difference between CHR and AQ-CTR for `blog_data`. In regions of overcoverage for CHR, we see less overcoverage and tightened intervals for AQ-CTR. Similarly, in regions of undercoverage for CHR, we see less undercoverage and widened intervals. We see this also for the `facebook` datasets. This suggests that AQ-CTR, while also being skew-adaptive, is in many cases more reliable than CHR. However, we note the improved log mean width for CHR without significant loss of coverage across all bins for all meps datasets. Finally, we point to the clear drop in interval width at the last bins for WM-CTR – an excellent example of the downsides of directly optimising for mean interval width.

Mean Absolute Deviation and Worst Slab Coverage

As can be seen in Tables 4.2 and 4.3, results for Q-CTR and AQ-CTR are in large part superior to CQR and WM-CTR. In general, we see performance is similar or improved compared to CHR – whilst CHR performs slightly better on the meps datasets, Q-CTR and AQ-CTR have a clear advantage for the four largest datasets. Also, we note that CQR and WM-CTR had significantly worse results for the largest three datasets.

4.3.2 Effect of Calibration Type on Conformal Training

Before our discussion, we finish this experimental section with a short investigation into two different calibration approaches during conformal training: our leave-one-out calibration (LOOC) and the split calibration (SC) from Stutz et al. (2021) and Lei and Bellotti (2023). We perform this investigation to see whether the use of LOOC is justified compared to SC.

The difference between the two approaches arises from how the score quantile is calculated. For LOOC, for each element of a mini-batch, we calculate the corrective score quantile using all of the other elements of the batch. For SC, the mini-batch is split into two – a calibration and a prediction batch. A single corrective score quantile is computed on the calibration batch and applied to the prediction batch. The loss for SC is applied to only the adjusted prediction batch.

We justify our use of LOOC since it is more memory- and data-efficient without degradation in performance compared to SC. Due to the fact that LOOC uses approximately twice as many predictions for calibration, a more precise estimation of the score quantile can be made using LOOC compared to SC. This implies that given the same hyperparameters, LOOC may decrease its loss further and more quickly during training. We hypothesise that if batch size and the number of epochs was doubled for SC, then the loss curve would look similar in shape.

In order to validate this, we use the facebook_1 dataset and compare loss curves for the two approaches. We adjust the CTR method to work with split calibration, namely with a split ratio of 1:1. We denote the respective models by CTR-SC. Our original batch size is 128. Considering the case where the proportion of scores used for calibration was the same between SC and LOOC, we additionally run the SC variant with double the batch size and double the number of epochs – denoting it by CTR-SC(2x). We do this to validate whether the loss curve would look similar to that of LOOC. We note that we downsample this loss curve by a factor of two to have a fair comparison between CTR and CTR-SC(2x), since only half of the training data is used during an epoch for calibration with SC given a split ratio of 1:1.

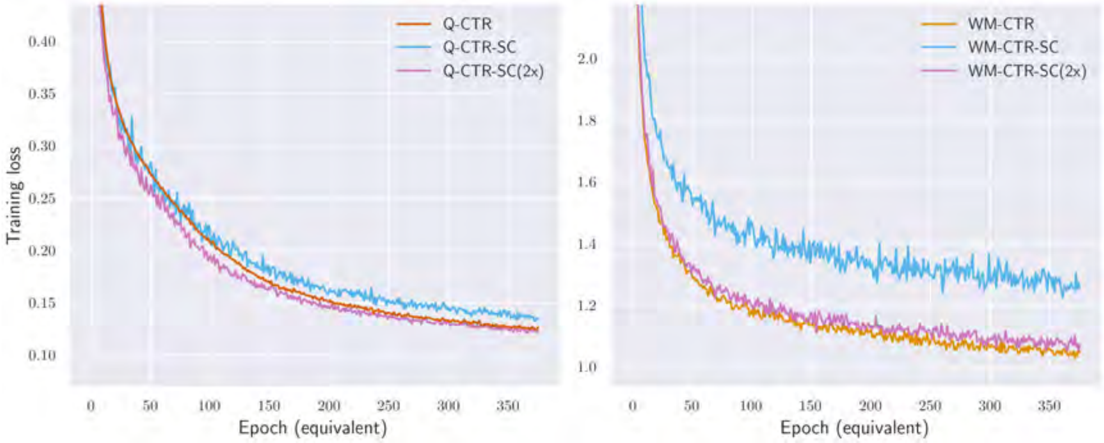

Fig. 4.3 Comparison of mean training loss curves over 10 different runs for the Q-CTR & Q-CTR-SC & Q-CTR-SC(2x) (left) and WM-CTR & WM-CTR-SC & WM-CTR-SC(2x) (right) models on the `facebook_1` dataset. By taking the mean of every consecutive non-overlapping pair of values, we downsample the loss curves by a factor of two for the CTR-SC(2x) variants.

We show the mean loss curves over 10 runs for Q-CTR, Q-CTR-SC, Q-CTR-SC(2x), WM-CTR, WM-CTR-SC, and WM-CTR-SC(2x) in Figure 4.3. As can be seen, the curves for CTR and the downsampled curves for CTR-SC(2x) curves are very similar. We believe this is sufficient validation for our hypothesis that the loss for CTR and CTR-SC decreases in a similar manner albeit it takes CTR-SC twice as long with a doubled batch size. We conclude that there is no significant difference between the two approaches in terms of *optimisation*. However, clearly justifying its use, LOOC is more memory- and data-efficient.

Finally, we note that leave-one-out calibration is slightly more computationally intensive: it requires 1) sorting the whole mini-batch compared to half of the mini-batch and 2) a comparison operation per mini-batch element to select the relevant quantile. As a further experiment, we run Q-CTR and Q-CTR-SC on the `facebook_1` dataset for 10 000 epochs and report an average time of 1.04 and 1.05 seconds per epoch for Q-CTR and Q-CTR-SC, respectively – a negligible time difference. The experiment was done on an Intel(R) Core(TM) i7-10700K CPU. We conclude that having made calibration more memory- and data-efficient using LOOC, there is no real impact on execution time.

4.4 Discussion

We conclude this chapter with a discussion of the results. We believe we have demonstrated several significant results supporting the use of conformal training in building reliable regression models for real use.

Post-training conformalisation is always biased in one way or another – including it into training allows the model to correct for this bias. Our results show that, especially for the larger datasets (with size greater than 10 000), both Q-CTR and AQ-CTR give predictions with significantly improved conditional coverage compared to the state-of-the-art CQR model. They also have similar or improved results compared to CHR. We also note that, based on Figure 4.2 and figures in Appendix B, Q-CTR seems to be more conservative and give wider predictive intervals in general. On the other hand, AQ-CTR gives tighter interval estimations than Q-CTR while having similar conditional coverage metrics. For this reason, we believe AQ-CTR is the most competitive model out of our proposed approaches.

Of course, CHR (Sesia and Romano (2021)) proves to be a very useful model, with it slightly outperforming other models on all of the `meps` datasets. We must note however, that CHR is a much more complex model than the CTR-based models, requiring a lengthy calibration procedure. It also requires the baseline quantile regression model to estimate a large set of quantiles. This gives CHR access to significantly more information for calibration and prediction, although it does make it computationally more intensive to train. Also, as a sidenote, since CHR uses a sequence of nested intervals calculated to minimize width, it can predict non-centred lower and upper bounds, something that AQ-CTR can do while CQR and Q-CTR cannot. We believe that this, accompanied by the robustness of the histogram-based approach, can give CHR an upper hand in some scenarios.

Furthermore, with WM-CTR we have clearly demonstrated that solely optimising for post-conformalisation predictive interval length will result in unreliable and misleading predictions. We believe this also applies for the classification setting in terms of optimising for predictive set cardinality. All-in-all this gives support to the claim that average predictive efficiency is a poor indicator of model reliability.

Chapter 5

Conclusion

This thesis explored the integration of conformalisation into the training process of conformal prediction for regression. In this final chapter, we conclude this work by listing our findings, discussing limitations, and proposing future directions for research.

5.1 Findings

We defined Conformal Training for Regression (CTR), a general approach to conformal training for regression which allows for any task-specific loss to directly optimise post-conformalised outputs. This freedom wouldn't be possible without the integration of mini-batch calibration within the training loop. Also, defining a loss function would usually require specifying a task-specific score function for conformalisation. However, as conformal training introduces information about the existing score function, finding an optimal score function is not necessarily required, making the CTR framework more robust. We believe these are important attributes of CTR and constitute the main reason to believe it has a positive impact in practise.

Based on the CTR framework, we defined three models, WM-CTR, Q-CTR, and AQ-CTR, which we analysed extensively on a synthetic dataset and multiple real datasets. With WM-CTR, we demonstrated the perils of optimising for predictive set size – even though mean width is reduced, conditional coverage is severely worsened, especially in regions of high uncertainty. With Q-CTR, we showed how introducing conformal awareness to the CQR model yields improved results after conformalisation. Finally, with AQ-CTR, we demonstrated a skew-adaptive approach which aims to find tight intervals while retaining relatively even coverage. For both Q-CTR and AQ-CTR, we showed that conditional coverage levels are similar or improve upon the CQR and CHR baselines.

As our conformal training framework used a different calibration approach than other previously defined end-to-end conformal training methods, we performed a comparative analysis. We demonstrated that our calibration is more memory- and data-efficient without any significant loss of execution time.

5.2 Limitations

Our CTR framework and conformal training in general has some limitations. First, it is more computationally expensive to train compared to the counterpart that does not use conformal training. This is mainly due to the sorting operation required for mini-batch calibration. However, if batch size is kept fixed, computational cost still scales linearly with dataset size. For this reason, we believe using this approach for large datasets remains feasible.

Another issue that is unexplored in this thesis is the problem of overfitting. While we use cross-validation to estimate an optimal training duration, we believe the discrepancy between the estimated score quantile during mini-batch calibration compared to the actual score quantile from the calibration set could be impactful and should therefore be investigated.

5.3 Future Work

We believe that the topic of conformal training for regression has great potential to play a role in developments in the field down the line. There are several avenues that open up with this thesis. For example, we used the default score function across all CTR-based models. The idea of adapting the score function by modelling additional values could be an interesting research direction. This may allow for the improvement of the conformalisation procedure used both during and after training. Another noteworthy direction is adapting the CTR method to work with Mondrian CP methods, or with models that give estimations of the full conditional distribution of the target, such as distributional conformal prediction (Chernozhukov et al. (2021)) or conditional histogram regression (Sesia and Romano (2021)).

References

Abdar, M., Pourpanah, F., Hussain, S., Rezazadegan, D., Liu, L., Ghavamzadeh, M., Fieguth, P., Cao, X., Khosravi, A., Acharya, U. R., Makarenkov, V., and Nahavandi, S. (2021). A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, 76:243–297.

Agency for Healthcare Research and Quality (2017). Medical expenditure panel survey hc-181: 2015 full year consolidated data. https://meps.ahrq.gov/mepsweb/data_stats/download_data_files_detail.jsp?cboPufNumber=HC-181. Accessed: April 2023.

Agency for Healthcare Research and Quality (2018). Medical expenditure panel survey hc-192: 2016 full year consolidated data. https://meps.ahrq.gov/mepsweb/data_stats/download_data_files_detail.jsp?cboPufNumber=HC-192. Accessed: April 2023.

Alaa, A. M., Hussain, Z., and Sontag, D. (2023). Conformalized unconditional quantile regression. In *International Conference on Artificial Intelligence and Statistics*, pages 10690–10702. PMLR.

Alnemer, L. M., Rajab, L., and Aljarah, I. (2016). Conformal prediction technique to predict breast cancer survivability. *Int J Adv Sci Technol*, 96:1–10.

Angelopoulos, A., Bates, S., Malik, J., and Jordan, M. I. (2020). Uncertainty sets for image classifiers using conformal prediction. *arXiv preprint arXiv:2009.14193*.

Angelopoulos, A. N. and Bates, S. (2021). A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *arXiv preprint arXiv:2107.07511*.

Bellotti, A. (2020). Constructing normalized nonconformity measures based on maximizing predictive efficiency. In *Conformal and Probabilistic Prediction and Applications*, pages 41–54. PMLR.

Bellotti, A. (2021). Optimized conformal classification using gradient descent approximation. *arXiv preprint arXiv:2105.11255*.

Blondel, M., Teboul, O., Berthet, Q., and Djolonga, J. (2020). Fast differentiable sorting and ranking. In *International Conference on Machine Learning*, pages 950–959. PMLR.

Boström, H. and Johansson, U. (2020). Mondrian conformal regressors. In Gammerman, A., Vovk, V., Luo, Z., Smirnov, E., and Cherubin, G., editors, *Proceedings of the Ninth Symposium on Conformal and Probabilistic Prediction and Applications*, volume 128 of *Proceedings of Machine Learning Research*, pages 114–133. PMLR.

Cauchois, M., Gupta, S., and Duchi, J. C. (2021). Knowing what you know: valid and validated confidence sets in multiclass and multilabel prediction. *The Journal of Machine Learning Research*, 22(1):3681–3722.

Chernozhukov, V., Wüthrich, K., and Zhu, Y. (2021). Distributional conformal prediction. *Proceedings of the National Academy of Sciences*, 118(48):e2107794118.

Chung, Y., Neiswanger, W., Char, I., and Schneider, J. (2021). Beyond pinball loss: Quantile methods for calibrated uncertainty quantification. *Advances in Neural Information Processing Systems*, 34:10971–10984.

Colombo, N. (2023). On training locally adaptive cp. *arXiv preprint arXiv:2306.04648*.

Einbinder, B.-S., Romano, Y., Sesia, M., and Zhou, Y. (2022). Training uncertainty-aware classifiers with conformalized deep learning. *Advances in Neural Information Processing Systems*, 35:22380–22395.

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

Feldman, S., Bates, S., and Romano, Y. (2021). Improving conditional coverage via orthogonal quantile regression. *Advances in neural information processing systems*, 34:2060–2071.

Foygel Barber, R., Candes, E. J., Ramdas, A., and Tibshirani, R. J. (2021). The limits of distribution-free conditional predictive inference. *Information and Inference: A Journal of the IMA*, 10(2):455–482.

Gneiting, T. and Raftery, A. E. (2007). Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378.

Izbicki, R., Shimizu, G. T., and Stern, R. B. (2019). Flexible distribution-free conditional predictive bands using density estimators. *arXiv preprint arXiv:1910.05575*.

Johansson, U., Sönströd, C., and Linusson, H. (2015). Efficient conformal regressors using bagged neural nets. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Koenker, R. and Bassett Jr, G. (1978). Regression quantiles. *Econometrica: journal of the Econometric Society*, pages 33–50.

Lei, H. and Bellotti, A. (2023). Reliable prediction intervals with directly optimized inductive conformal regression for deep learning. *arXiv preprint arXiv:2302.00872*.

Lei, J., G'Sell, M., Rinaldo, A., Tibshirani, R. J., and Wasserman, L. (2018). Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113(523):1094–1111.

Lei, J. and Wasserman, L. (2014). Distribution-free prediction bands for non-parametric regression. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 76(1):71–96.

Luo, R., Zhao, S., Kuck, J., Ivanovic, B., Savarese, S., Schmerling, E., and Pavone, M. (2022). Sample-efficient safety assurances using conformal prediction. In *International Workshop on the Algorithmic Foundations of Robotics*, pages 149–169. Springer.

Papadopoulos, H., Gammerman, A., and Vovk, V. (2008). Normalized nonconformity measures for regression conformal prediction. In *Proceedings of the IASTED International Conference on Artificial Intelligence and Applications (AIA 2008)*, pages 64–69.

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

Papadopoulos, H., Proedrou, K., Vovk, V., and Gammerman, A. (2002). Inductive confidence machines for regression. In *Machine Learning: ECML 2002: 13th European Conference on Machine Learning Helsinki, Finland, August 19–23, 2002 Proceedings 13*, pages 345–356. Springer.

Pereira, T., Cardoso, S., Guerreiro, M., Madeira, S. C., Initiative, A. D. N., et al. (2020). Targeting the uncertainty of predictions at patient-level using an ensemble of classifiers coupled with calibration methods, venn-abers, and conformal predictors: A case study in ad. *Journal of biomedical informatics*, 101:103350.

Romano, Y., Patterson, E., and Candes, E. (2019). Conformalized quantile regression. *Advances in neural information processing systems*, 32.

Romano, Y., Sesia, M., and Candes, E. (2020). Classification with valid and adaptive coverage. *Advances in Neural Information Processing Systems*, 33:3581–3591.

Sesia, M. and Candès, E. J. (2020). A comparison of some conformal quantile regression methods. *Stat*, 9(1):e261.

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

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

Shi, F., Ong, C. S., and Leckie, C. (2013). Applications of class-conditional conformal predictor in multi-class classification. In *2013 12th International Conference on Machine Learning and Applications*, volume 1, pages 235–239. IEEE.

Singh, K., Sandhu, R. K., and Kumar, D. (2015). Comment volume prediction using neural networks and decision trees. In *IEEE UKSim-AMSS 17th International Conference on Computer Modelling and Simulation, UKSim2015 (UKSim2015).*

Steinwart, I. and Christmann, A. (2011). Estimating conditional quantiles with the help of the pinball loss. *Bernoulli*, 17(1):211 – 225.

Stutz, D., Cemgil, A. T., Doucet, A., et al. (2021). Learning optimal conformal classifiers. *arXiv preprint arXiv:2110.09192*.

UC Irvine Machine Learning Repository (2007). Concrete compressive strength. http://archive.ics.uci.edu/dataset/165/concrete+compressive+strength. Accessed: April 2023.

UC Irvine Machine Learning Repository (2009). Communities and crime. http://archive.ics.uci.edu/dataset/183/communities+and+crime. Accessed: April 2023.

UC Irvine Machine Learning Repository (2013a). Bike sharing dataset. https://archive.ics.uci.edu/dataset/275/bike+sharing+dataset. Accessed: April 2023.

UC Irvine Machine Learning Repository (2013b). Physicochemical properties of protein tertiary structure. https://archive.ics.uci.edu/dataset/265/physicochemical+properties+of+protein+tertiary+structure. Accessed: April 2023.

UC Irvine Machine Learning Repository (2014). Blogfeedback. https://archive.ics.uci.edu/dataset/304/blogfeedback. Accessed: April 2023.

UC Irvine Machine Learning Repository (2016). Facebook comment volume dataset. https://archive.ics.uci.edu/dataset/363/facebook+comment+volume+dataset. Accessed: April 2023.

Vovk, V. (2012). Conditional validity of inductive conformal predictors. In *Asian conference on machine learning*, pages 475–490. PMLR.

Vovk, V., Gammerman, A., and Shafer, G. (2005). *Algorithmic learning in a random world*, volume 29. Springer.

Vovk, V., Lindsay, D., Nouretdinov, I., and Gammerman, A. (2003). Mondrian confidence machine. *Technical Report*.

Vovk, V., Nouretdinov, I., and Gammerman, A. (2009). On-line predictive linear regression. *The Annals of Statistics*, pages 1566–1590.

Word, E., Johnston, J., Bain, H. P., Fulton, B. D., Zaharias, J. B., Achilles, C. M., Lintz, M. N., Folger, J., and Breda, C. (1990). The state of tennessee's student/teacher achievement ratio (star) project. *Tennessee Board of Education*.

Appendix A

Adaptive-Quantile Interval Score Loss is Faithful to Target Coverage

In this Appendix, we prove Theorem 3.5.1 which states that the adaptive-quantile interval score loss achieves a minimum whenever $\bar{y}^{\text{lo}} = q_{(\alpha-\beta)/2}$ and $\bar{y}^{\text{hi}} = q_{1-(\alpha+\beta)/2}$ and thus is faithful to a coverage level of $1-\alpha$. We restate the adaptive-quantile interval score loss from Equation 3.3. For a specified target level of miscoverage α, the adaptive-quantile interval score loss is given by

$$l_\alpha(\bar{y}^{\text{lo}}, \bar{y}^{\text{hi}}, \beta, y) = (\bar{y}^{\text{hi}} - \bar{y}^{\text{lo}}) + \frac{2}{\alpha-\beta}\mathbb{1}_{y<\bar{y}^{\text{lo}}}(\bar{y}^{\text{lo}} - y) + \frac{2}{\alpha+\beta}\mathbb{1}_{y>\bar{y}^{\text{hi}}}(y - \bar{y}^{\text{hi}})$$

where β is a quantile adjustment term with $|\beta| < \alpha$, $\bar{y}^{\text{lo}}$ is an estimation of $q_{(\alpha-\beta)/2}$, $\bar{y}^{\text{hi}}$ is an estimation of $q_{1-(\alpha+\beta)/2}$, and y is the target variable.

We will now provide the proof which is an adjustment of the proof by Chung et al. (2021) for the same property of the basic interval score loss.

Proof. First, we consider the expected loss under the target distribution f with respective c.d.f. F_Y such that $y \sim f$:

$$\begin{aligned}
\mathbb{E}_{y\sim f}\Big[l_\alpha(\bar{y}^{\text{lo}}, \bar{y}^{\text{hi}}, \beta, y)\Big] &= \int_{-\infty}^{\infty} l_\alpha(\bar{y}^{\text{lo}}, \bar{y}^{\text{hi}}, \beta, y) f(y)dy \\
&= (\bar{y}^{\text{hi}} - \bar{y}^{\text{lo}}) \\
&\quad + \frac{2}{\alpha-\beta}\int_{-\infty}^{\bar{y}^{\text{lo}}} (\bar{y}^{\text{lo}} - y) f(y)dy \\
&\quad + \frac{2}{\alpha+\beta}\int_{\bar{y}^{\text{hi}}}^{\infty} (y - \bar{y}^{\text{hi}}) f(y)dy.
\end{aligned}$$

Next, we calculate the partial derivatives of the expectation with respect to both $\bar{y}^{\text{lo}}$ and $\bar{y}^{\text{hi}}$. We have

$$\begin{aligned}
\frac{\partial \mathbb{E}_{y\sim f}\Big[l_\alpha(\bar{y}^{\text{lo}}, \bar{y}^{\text{hi}}, \beta, y)\Big]}{\partial \bar{y}^{\text{lo}}} &= -1 + \frac{2}{\alpha-\beta}\int_{-\infty}^{\bar{y}^{\text{lo}}} f(y)dy = -1 + \frac{2}{\alpha-\beta}F_Y(\bar{y}^{\text{lo}}) \text{ and} \\
\frac{\partial \mathbb{E}_{y\sim f}\Big[l_\alpha(\bar{y}^{\text{lo}}, \bar{y}^{\text{hi}}, \beta, y)\Big]}{\partial \bar{y}^{\text{hi}}} &= \quad 1 - \frac{2}{\alpha+\beta}\int_{\bar{y}^{\text{hi}}}^{\infty} f(y)dy = \quad 1 - \frac{2}{\alpha+\beta}\Big(1 - F_Y(\bar{y}^{\text{hi}})\Big).
\end{aligned}$$

Finally, we set these partial derivatives to zero. We hence have that the adaptive-quantile interval score loss is minimised whenever $F_Y(\bar{y}^{\text{lo}}) = \frac{\alpha-\beta}{2}$ and $F_Y(\bar{y}^{\text{hi}}) = 1 - \frac{\alpha+\beta}{2}$, i.e. when $\bar{y}^{\text{lo}} = q_{(\alpha-\beta)/2}$ and $\bar{y}^{\text{hi}} = q_{1-(\alpha+\beta)/2}$ with $F_Y(\bar{y}^{\text{hi}}) - F_Y(\bar{y}^{\text{lo}}) = 1 - \frac{\alpha+\beta}{2} - \frac{\alpha-\beta}{2} = 1 - \alpha$ giving the target level of coverage. □

Appendix B

Extended Results for All Real Datasets

In this appendix, we show the results over all real datasets (see Table 4.1) of all the relevant models as in Figure 4.2. The results can be seen in Figures B.1, B.2, B.3, and B.4.

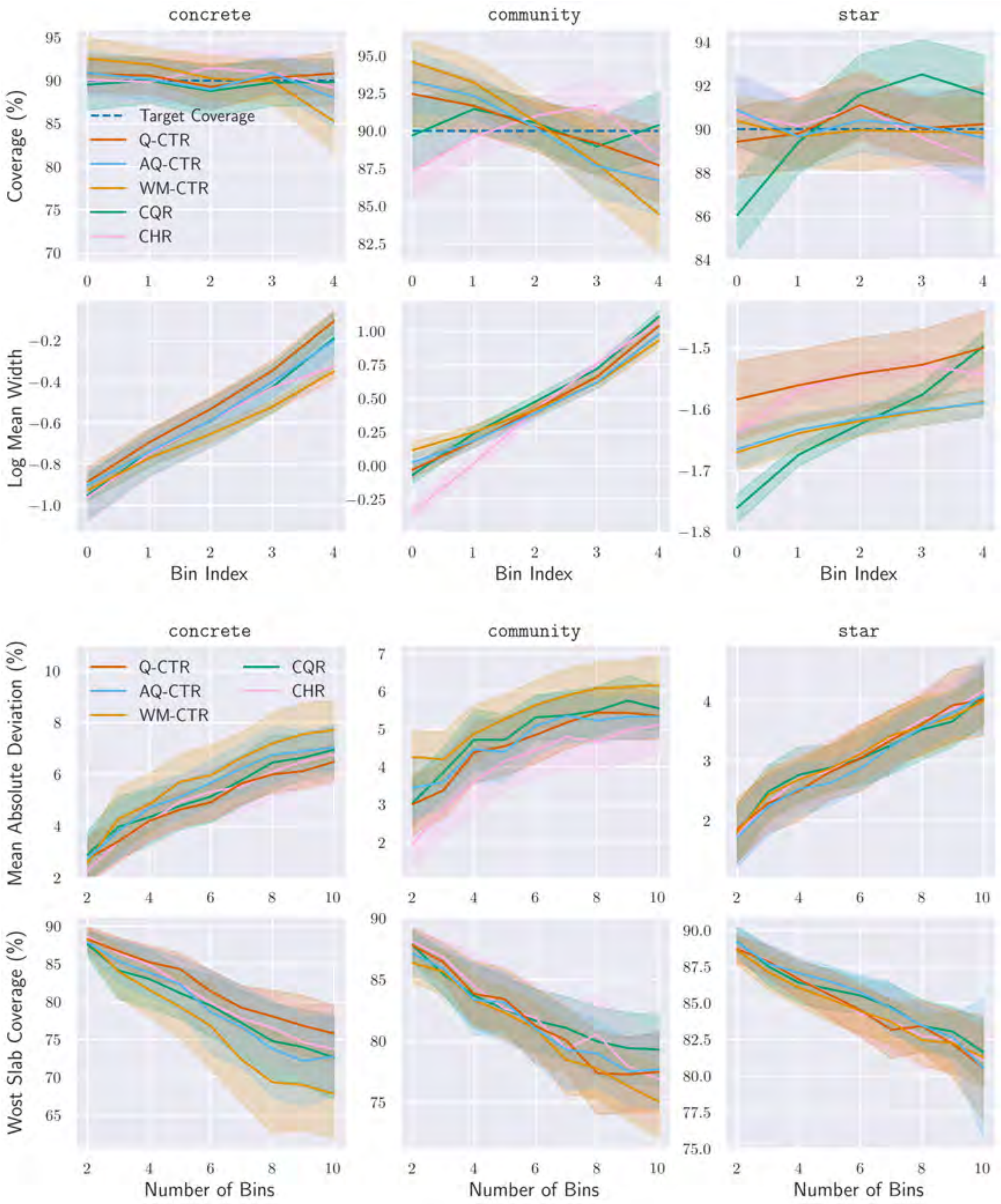

Fig. B.1 Various metrics measuring conditional coverage and predictive interval width for `concrete`, `community`, and `star` over 30 seeded runs with means and half standard deviations reported. See Figure 4.2 for more details.

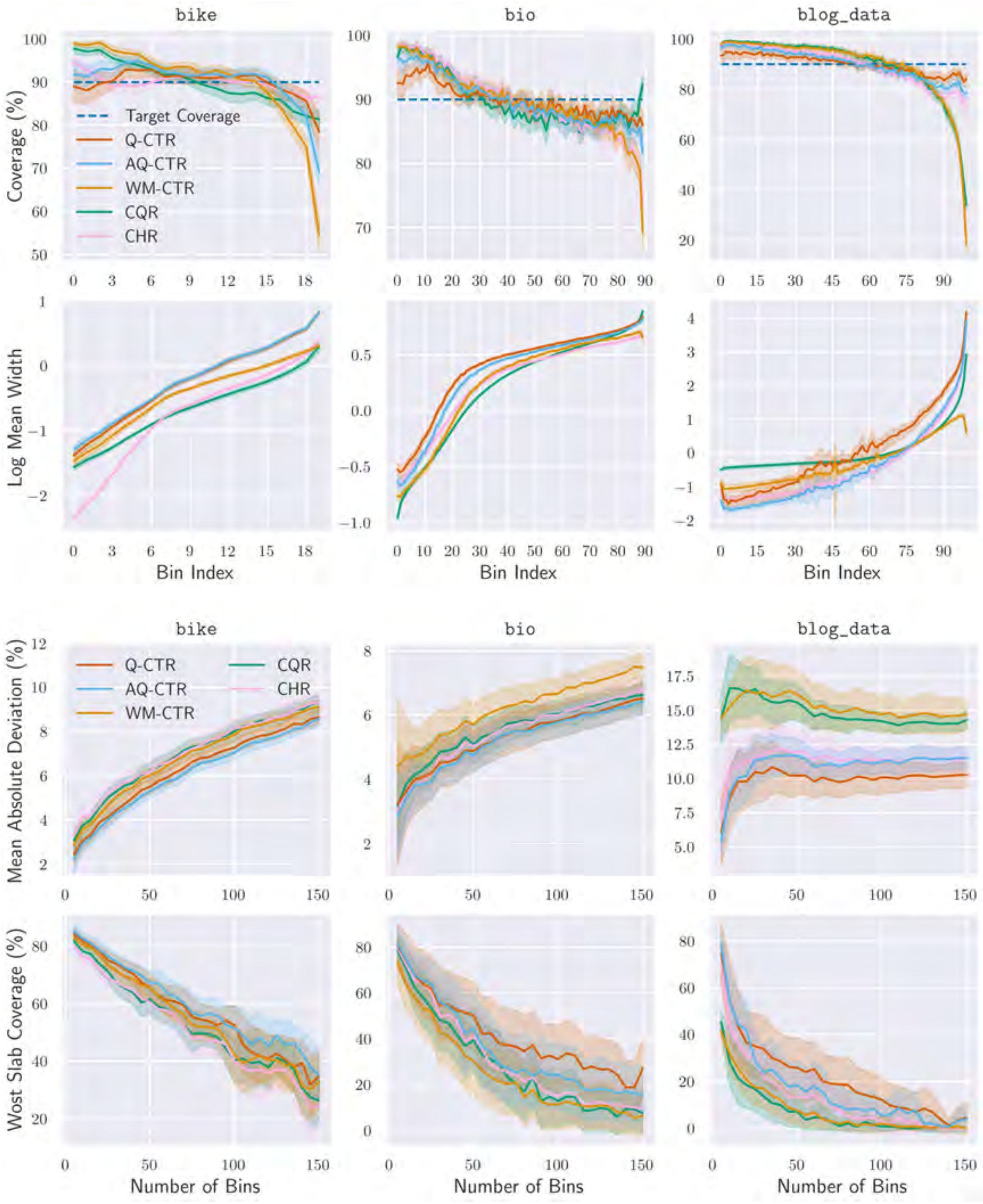

Fig. B.2 Various metrics measuring conditional coverage and predictive interval width for `bike`, `bio`, and `blog_data` over 30 seeded runs with means and half standard deviations reported. See Figure 4.2 for more details.

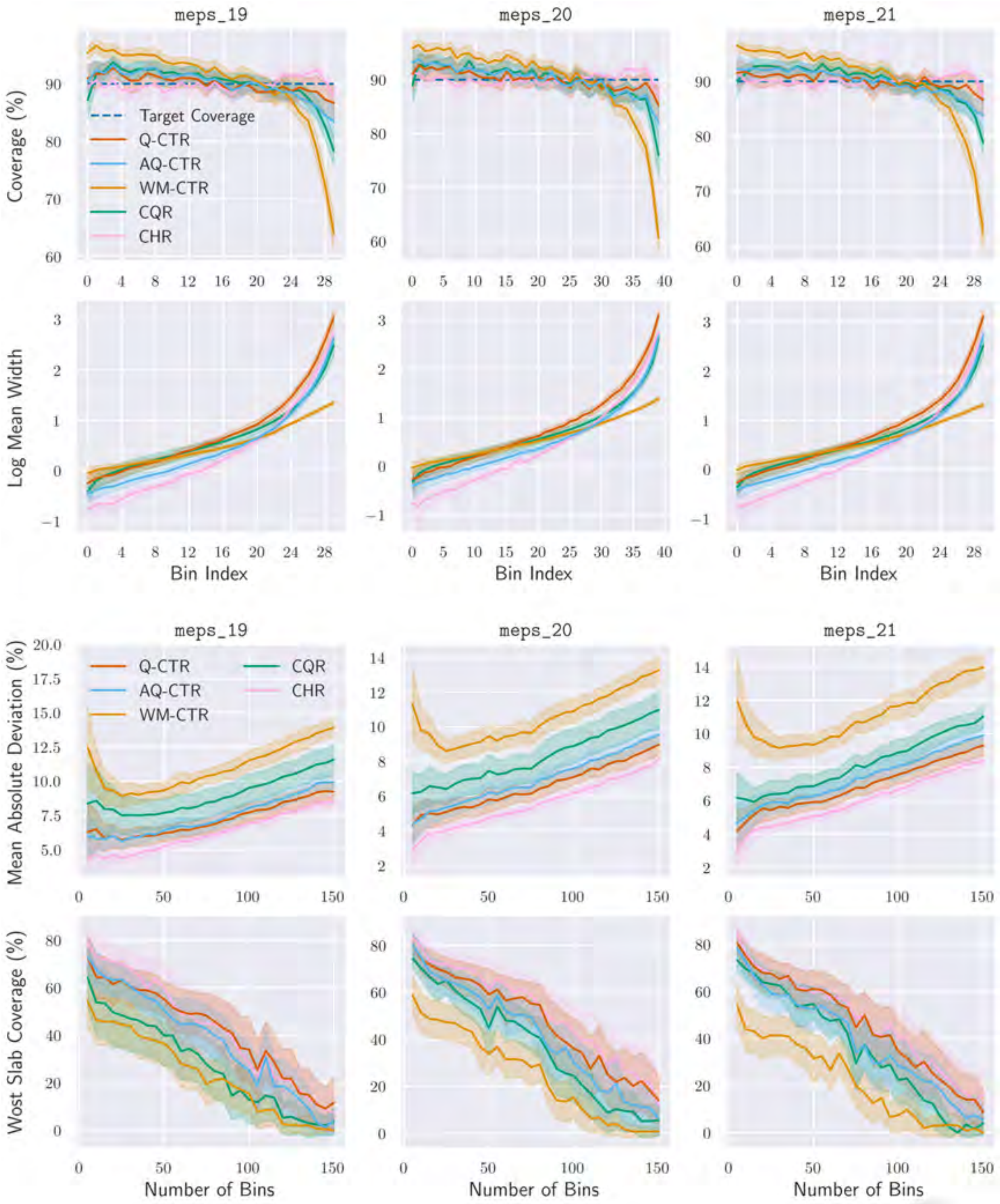

Fig. B.3 Various metrics measuring conditional coverage and predictive interval width for `meps_19`, `meps_20`, and `meps_21` over 30 seeded runs with means and half standard deviations reported. See Figure 4.2 for more details.

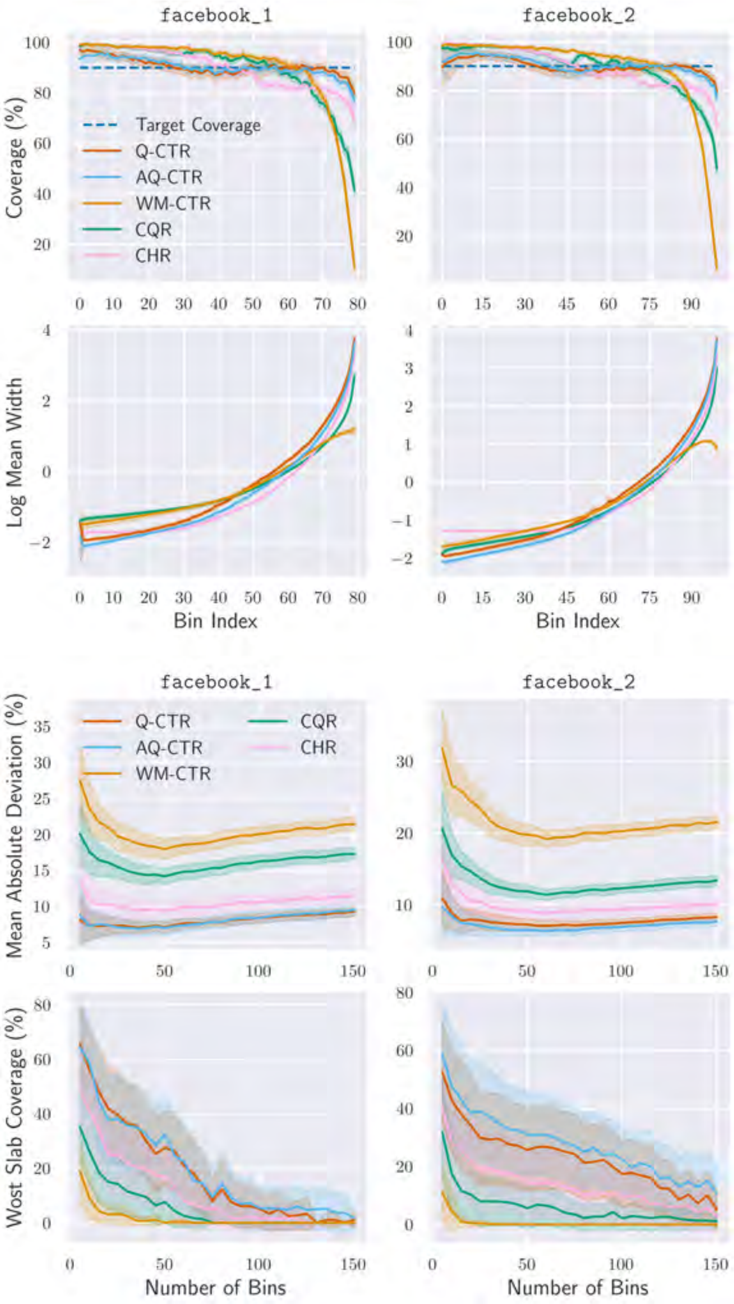

Fig. B.4 Various metrics measuring conditional coverage and predictive interval width for `facebook_1` and `facebook_2` over 30 seeded runs with means and half standard deviations reported. See Figure 4.2 for more details.